

Systems of Systems

Systems of Systems

Edited by
Dominique Luzeaux
Jean-René Ruault

ISTE

 WILEY

First published 2008 in France by Hermes Science/Lavoisier in two volumes entitled: *Systèmes de systèmes : concepts et illustrations pratiques* and *Ingénierie des systèmes de systèmes: méthodes et outils*
© LAVOISIER 2008

First published 2010 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms and licenses issued by the CLA. Enquiries concerning reproduction outside these terms should be sent to the publishers at the undermentioned address:

ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

© ISTE Ltd 2010

The rights of Dominique Luzeaux and Jean-René Ruault to be identified as the authors of this work have been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

Library of Congress Cataloging-in-Publication Data

Systèmes de systèmes. English
Systems of systems / edited by Dominique Luzeaux, Jean-René Ruault.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-84821-164-3
1. Systems engineering. I. Luzeaux, Dominique. II. Ruault, Jean-René. III. Title.
TA168.S887813 2010
620.001'171--dc22

2009044182

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISBN: 978-1-84821-164-3

Printed and bound in Great Britain by CPI Antony Rowe, Chippenham and Eastbourne



Table of Contents

Author Biographies	xv
Introduction	xix
PART 1. SYSTEMS OF SYSTEMS, CONCEPTS AND PRACTICAL ILLUSTRATIONS	1
Chapter 1. Systems of Systems: From Concept to Actual Development	3
Dominique LUZEAUX	
1.1. Network omnipresence creating a worldwide environment.	3
1.2. Increasing complexity of the environment	5
1.2.1. A particular field: defense.	6
1.2.2. Impact of context evolutions on the defense systems	7
1.3. Towards a definition of the concept of system of systems	11
1.3.1. From system to system of systems.	11
1.3.2. Examples for various value chains.	18
1.3.3. Epistemological return to the notions of emergence and openness	29
1.3.4. Small aside: system or system of systems?	32
1.4. Control of the system of systems	34
1.4.1. System engineering.	34
1.4.2. Is there a need for systems of systems engineering?	36
1.4.3. Architecture: a key element.	41
1.4.4. Control of the interfaces.	45
1.4.5. Traceability: a mandatory component.	47
1.5. Tools for the control of the system of systems	47
1.5.1. Simulation	49

1.5.2. Towards integrated infrastructures: the battle-labs	52
1.6. The need for standardization.	56
1.7. The human factor in systems of systems.	58
1.7.1. The user: operator, supervisor, decision maker	58
1.7.2. Operating system support	60
1.7.3. Designer	61
1.7.4. Customer, supplier	62
1.7.5. The human factor in systems of systems: man's expectations. . . .	62
1.7.6. Standardizing the human factor in systems of systems	66
1.8. Budgetary aspects of the systems of systems	68
1.9. The need for governance	70
1.9.1. New models of competition.	71
1.9.2. New organizations	72
1.9.3. New relations between project management and general contracting: the role of system integration project managers	74
1.10. Conclusion	75
1.11. Appendix: system of systems' definitions in literature.	77
1.12. Bibliography	84
Chapter 2. Emergence and Complexity of Systems of Systems	89
Patrice MICOUIN	
2.1. Introduction.	89
2.2. Matter and shape.	90
2.3. Systems	92
2.3.1. Systems and subsystems.	93
2.3.2. Resulting and emergent properties of a system	94
2.3.3. Natural and artificial systems.	95
2.3.4. Abstract and concrete artificial systems.	95
2.3.5. Technological systems.	97
2.4. Genesis of concrete systems	99
2.4.1. Genesis of natural systems	99
2.4.2. Genesis of technological systems	100
2.5. Complexity of systems of systems	107
2.6. Systems of systems engineering.	111
2.7. Conclusion	115
2.8. Bibliography	116
Chapter 3. Contractual Aspects of the Acquisition and Use of Systems of Systems	119
Danièle VÉRET	
3.1. Introduction.	119
3.2. An integrated set of components of various natures	121

3.2.1. Material components	121
3.2.2. Software elements	122
3.2.3. The human factor	123
3.3. Combining people with diversified skills and their contributions	125
3.3.1. Diversity of the agents	125
3.3.2. Project management	126
3.3.3. Competitive bidding	128
3.4. Commitments to coordinate	130
3.4.1. Effective date and duration of contractual commitments.	130
3.4.2. Delivery	131
3.4.3. Receipt	132
3.4.4. Financial matters	133
3.4.5. Guarantees	135
3.4.6. The combination of limitations of liability	137
3.4.7. The end of commitments	141
3.5. Ownership rights.	142
3.5.1. Corporeal property	142
3.5.2. The patent	143
3.5.3. Copyrights and the particular case of software.	143
3.5.4. The databases	146
3.5.5. Designs and models	146
3.5.6. Brands and logos	146
3.5.7. The domain name.	146
3.6. The most adapted legal strategies	147
3.7. Conclusion	148
 Chapter 4. The Human Factor within the Context of Systems of Systems	 149
Jean-René RUAULT	
4.1. Introduction.	149
4.2. Definition and epistemological aspects.	150
4.3. The issue	154
4.3.1. Example of system within systems engineering	154
4.3.2. The notion of system of systems	156
4.3.3. Organizations' constraints on systems of systems.	159
4.4. Current human factors in systems engineering	160
4.4.1. Designing an organization from the standpoint of systems engineering	160
4.4.2. Social networks and multi-agent systems.	162
4.4.3. Wrap-up on the current human factor in systems engineering. . . .	164
4.5. The organizations' complexity from the standpoint of social sciences: impacts on the systems of systems.. . . .	166

4.5.1. The organizations' design from the standpoint of social sciences	166
4.5.2. Informal and individual dimension within organizations.	173
4.5.3. Internal and external environment of organizations.	176
4.5.4. Professional, organizational and national cultures in organizations	180
4.5.5. Sensemaking in organizations and mutual intelligibility.	186
4.5.6. Impacts of the introduction of information technologies within organizations	190
4.5.7. Network-centric organizations	191
4.6. Social sciences implemented within the context of systems of systems	192
4.6.1. Impact of information technologies on network-centric operations	193
4.6.2. Example of the network-centric operations conceptual framework	195
4.6.3. Impact of network-centric operations: from technical to organizational interoperability	198
4.7. Recognizable good practices in the field of organizations	201
4.8. Conclusion	202
4.9. Acknowledgments.	203
4.10. Bibliography	203
Chapter 5. Space Communication and Observation System of Systems . .	207
Frédéric PRADEILLES and Dominique LUZEAUX	
5.1. The dual context of omnipresent information and the commoditization of space	207
5.2. The technical view: an interconnection of ground-based and space-borne systems	209
5.2.1. Telecommunication and navigation satellite systems.	210
5.2.2. Space-borne remote sensing and observation systems	212
5.3. Search for functionality and capacity.	213
5.4. A logic of exchange on an international scale.	214
5.4.1. The GEOSS program.	215
5.4.2. Necessary governance of the GEOSS program	217
5.4.3. Capacity exchanges in the military field	218
5.5. Conclusion	220
5.6. Bibliography	221
Chapter 6. Intelligent Transport Systems	223
Michel CHAVRET	
6.1. The field of intelligent transport.	223

6.1.1. ITS	223
6.1.2. The systems in use	223
6.1.3. An international approach	225
6.2. ACTIF	226
6.2.1. The ACTIF perimeter	227
6.2.2. The ACTIF model	229
6.3. Practical application	230
6.3.1. Context	230
6.3.2. Architecture approach	230
6.3.3. Modeling	232
6.4. Conclusion	234
6.5. Bibliography	234
Chapter 7. Systems of Systems in the Healthcare Field	235
Jean-René RUAULT	
7.1. Introduction	235
7.2. From capability challenges to the design of systems of systems.	236
7.3. Personal service, the main characteristic of systems within the healthcare field	239
7.4. Coordination of the medical and paramedical agents, in hospitals and in private practices.	242
7.5. The development of information technologies and their interoperability, heart of the healthcare networks issue.	245
7.5.1. Information technologies in the healthcare field.	245
7.5.2. Interoperability in the healthcare field.	247
7.6. Difficulties encountered	256
7.7. Conclusion	258
7.8. Acknowledgments	258
7.9. Bibliography	259
Chapter 8. Critical Infrastructure Protection	261
Jean-Luc ZOLELIO	
8.1. General context of critical infrastructure protection	261
8.1.1. Challenges	261
8.1.2. Structure of a vital infrastructure.	262
8.1.3. Hazards and threats.	265
8.2. Protection requirements	266
8.2.1. Looking at the infrastructure in its entirety	266
8.2.2. A structured, continuous approach.	268
8.2.3. Confidentiality	269
8.2.4. Security analysis file	270

8.2.5. Decision making	270
8.2.6. Admissibility	271
8.3. Security systems of the future	272
8.3.1. Proactivity, crisis management and resilience	272
8.3.2. Early reduction of risk	273
8.3.3. Electronic detection systems	274
8.3.4. Plug and play	278
8.3.5. Crisis management tools	279
8.4. The human factor	285
8.4.1. Monitoring	286
8.4.2. The man-machine interface	288
8.4.3. Training	289
8.5. Conclusion	290
Chapter 9. Globalization and Systemic Impacts	291
Dominique LUZEAUX, Jean-René RUAULT and Lui KAM	
9.1. Introduction.	291
9.2. System of systems “globalization”	292
9.2.1. Globalization: a concept with many meanings.	292
9.2.2. A long story	293
9.2.3. The facilitating factors of globalization	295
9.2.4. The necessity of a systemic standpoint	297
9.2.5. The various dimensions of the “globalization” system of systems’ value chain.	298
9.2.6. The utopia of a standardizing globalization	302
9.2.7. The use of new systemic interpretations to understand the mechanisms of globalization	303
9.3. Beyond the concepts of systems.	309
9.3.1. Human-intensive systems	309
9.3.2. Perverse effects and paradoxes	310
9.4. Globalization’s impact on systems of systems engineering.	312
9.4.1. New opportunities and new challenges	312
9.4.2. Cultural factors	314
9.4.3. Administrative factors	315
9.5. Conclusion	316
9.6. Appendix: a summary of the properties of nonlinear dynamic systems	317
9.7. Bibliography	318

PART 2. SYSTEMS OF SYSTEMS ENGINEERING, METHODS, STANDARDS AND TOOLS.	321
Chapter 10. Methods and Tools for Systems of Systems Engineering.	323
Dominique LUZEAUX	
10.1. Systems of systems engineering: from the control of complexity to the necessity of a model-driven approach	323
10.2. Architecture	326
10.2.1. Architecture: an ally of systems of systems.	326
10.2.2. Application examples: combat direction system within the naval aviation system of systems	328
10.3. From architecture to detailed design: reference architectures.	331
10.3.1. Reference architectures.	331
10.3.2. Two examples of architecture reference models	334
10.3.3. Openness: an essential criterion.	336
10.4. Requirement traceability and engineering tools.	338
10.5. Reverse engineering and impact studies	342
10.6. Distributed simulation tools for model engineering	344
10.7. Global control of operational security via testability.	346
10.8. Towards a virtuous circle of simulation-tests to control the tests	352
10.8.1. Integrated simulation-tests approach at the service of model-driven engineering.	352
10.8.2. VV&A and VV&C	355
10.9. Collaborative work tools	357
10.9.1. New technologies, new work methods, virtual teams	357
10.9.2. Collaborative work environments for systems of systems engineering	358
10.10. Conclusion.	360
10.11. Acknowledgements	361
10.12. Bibliography.	362
Chapter 11. Model-driven Design and Simulation.	363
Lui KAM	
11.1. General points	363
11.2. A few definitions.	365
11.2.1. Modeling	366
11.2.2. Metamodeling	368
11.2.3. Simulation	370
11.2.4. Interoperability	372
11.2.5. Verification and validation	374
11.3. Model-driven engineering	378
11.3.1. The MDA conceptual framework.	378

11.3.2. MDA methodological framework	381
11.3.3. Another instance of MDE: the DSL tools	383
11.4. Feedback.	385
11.4.1. Issue faced by the DGA	385
11.4.2. Feasibility study of the MDA approach	386
11.4.3. Feasibility study of the MDE approach	388
11.4.4. Feasibility study of the models' capitalization and reuse	391
11.5. Conclusion and perspectives	392
11.6. Bibliography	394
 Chapter 12. Standardization in the Field of Systems and Systems of Systems Engineering	 399
Jean-René RUAULT and Jean-Pierre MEINADIER	
12.1. Introduction	399
12.2. Example of the importance of standards in the interoperability of systems and systems of systems	400
12.3. Standards used in the field of systems and systems of systems.	403
12.3.1. Business process modeling notation	404
12.3.2. Business data exchange	406
12.3.3. Descriptions and characteristics of systems, products and services.	407
12.3.4. Engineering processes	422
12.3.5. Standards relative to the exchange of engineering data	430
12.4. Application and adaptation of system engineering standards in the context of systems of systems	433
12.5. Implementation of standards in the context of systems of systems	438
12.6. Conclusion	439
12.7. Acknowledgements	439
12.8. Appendix A. Standard relative to business process modeling	439
12.9. Appendix B. Standard relative to the Web services business process execution language	443
12.10. Appendix C. Ontology definition metamodel specification	444
12.11. Appendix D. UML profile for DoDAF/MODAF (USA Department of Defense and UK Ministry of Defense Architecture Framework)	446
12.12. Appendix E. Standard relative to software-intensive systems architecture.	451
12.13. Appendix F. Unified modeling language	454
12.14. Appendix G. Systems modeling language	457
12.15. Appendix H. Good practices of IT service management, ITIL	461
12.16. Appendix I. Standard relative to IT services management.	464
12.17. Appendix J. Software engineering – Product quality	466

12.18. Appendix J.1. Standard ISO 9126, part 1, quality model	466
12.19. Appendix J.2. Standard ISO 9126, part 3, internal metrics	468
12.20. Appendix K. Standard on software product quality requirements and evaluation	468
12.21. Appendix L. Standard on the common criteria for IT security evaluation	469
12.22. Appendix M. Standard relative to a system's life cycle process . . .	473
12.23. Appendix N. Standard relative to the processes for engineering a system.	482
12.24. Appendix O. Standard for the application and management of the systems engineering process	487
12.25. Appendix P. Standard relative to software life cycle processes . . .	494
12.26. Appendix Q. Standard relative to software measurement process . .	499
12.27. Appendix R. Standard relative to software product evaluation . . .	500
12.28. Appendix S. Standard on systems engineering, product and design data exchange	504
12.29. Appendix T. Standard on the exchange of product model data, products life cycle support.	507
12.30. Bibliography.	510
Conclusion	513
List of Authors	519
Index	521

Author Biographies

Michel CHAVRET

Engineer in automatisms and informatics, Michel Chavret has dedicated most of his career to the design of systems in the field of transport, with a few years spent developing complex systems for industry and the army. Since 2000, when the SETEC ITS society was created, he has directed Lyon's agency. He was a project director in the ACTIF project, keystone of modeling and design of systems of systems in France in the field of intelligent transport systems.

Lui KAM

Lui Kam graduated from the *Ecole nationale supérieure des ingénieurs des études et techniques d'armement* (ENSIETA) in 1996, from Paris XI University with a doctorate in electronics in 2000, and from the London Business School with a Master of Business Administration in 2007. During his 11-year-long career in the *Délégation Générale de l'Armement* (DGA), his fields of expertise covered the signal and image processing, the methods, tools and standards used in modeling and simulation, as well as in systems engineering. Since 2007, he has been working as a Business Strategy and Development Consultant, and is currently based in Shanghai.

Dominique LUZEAUX

Dominique Luzeaux is a graduate from the *Ecole polytechnique* (1987) and the *Ecole nationale supérieure des techniques avancées* (1989). After graduating with a PhD from Paris XI University in 1991, he was a research fellow at Berkeley University till 1992. Hired by the DGA, he has taken on various technical responsibilities in the fields of robotics, optoelectronics and observation systems. From 2002 to 2004, he was the director of Simulation-Based Acquisition at the DGA, where he notably oversaw the R&T (Research & Technology) programs in system engineering. From 2005 to 2007, he was director of the IT production center.

From 2008 to 2009, he was deputy director of the C4ISR programs. Since April 2009, he has held the position of director of Land Systems acquisition. Moreover, accredited to supervise research since 2001, he has overseen a dozen doctoral theses and published more than 60 articles in conferences and international reviews. He teaches robotics at the ENSTA, systems of systems engineering at the ENSIETA and the ISAE, and is also a speaker in mathematics and computer sciences theory at the *University of Montpellier II*. Laureate of the prize of *Ingénieur général Chanson* in 2006 for his works in the field of military terrestrial autonomous robotics, he co-wrote *A la conquête du nanomonde : nanotechnologies et microsystèmes* with Thierry Puig, published by the *Editions du Félin* in March of 2007.

Jean-Pierre MEINADIER

Jean-Pierre Meinadier is an engineer of l'*Ecole centrale*, an honorary teacher at the *Conservatoire national des arts et métiers*, and a scientific consultant at the AFIS. He first developed his career in the field of IT systems engineering and integration at the CEA from 1963 to 1974, where he created real-time systems activity. He then created and directed, from 1975 to 1986, the head office of GIXI, a systems engineering company of the CEA-CISI group. Consultant from 1987 and later professor, he founded the chair of systems integration at the CNAM in 1990. He has penned several works in the field of systems architecture and engineering. He has taught computer architecture, and later systems engineering, in several engineering schools.

Patrice MICOUIN

Senior consultant in the field of systems engineering, Patrice Micouin works with enterprises such as DCNS, Airbus, Eurocopter and the CNES. In 2006, he held a doctorate dissertation at the *Ecole nationale supérieure des arts et métiers* (ENSAM) on the definition and implementation of system engineering processes in the automotive sector. He teaches a class on system engineering at the ENSAM, gives speeches at l'*Ecole centrale Paris*, and supervises, in the LSIS laboratory (UMR CNRS 6168), research centered on requirements engineering, system design and knowledge engineering.

Frédéric PRADEILLES

Graduate of l'*Ecole Polytechnique*, and holder of a PhD in mathematics, Frédéric Pradeilles first taught mathematics at *SUPAERO* (now ISAE), before joining the DGA to take on various technical responsibilities in the fields of observation and intelligence, and the use of space in the gathering of intelligence. He has also been director of the research and technological programs in the field of complex system engineering, and has acted as a representative for the DGA at the System@tic Paris-Région Cluster. Today, he is Chief Technical Officer within *CS Systèmes d'Information*.

Jean-René RUAULT

After a DEA in experimental social psychology, Jean-René Ruault followed additional training in industrial informatics. He worked in various service firms for more than ten years, contributing to projects at various stages of the systems' life cycles. He joined the DGA in 2004, where he now works in systems engineering within the SdS pole. He co-chairs the working group on systems of systems within the AFIS. Moreover, he has published several articles in the field of systems of systems engineering and human-machine interactions. He was co-president of the Ergo'IA conference in 2006.

Danièle VÉRET

Danièle Véret a barrister at Paris Bar has a Master's degree in public and private law, and a DEA in comparative law, "*droit anglais et nord-américain des affaires*" (British and North-American business law). She rapidly turned to IT laws, and later to the laws regulating new technologies. A lawyer within the legal department of an SSI, and later at ALAIN BENSOUSSAN AVOCATS, she takes care of counseling and litigation cases, out of court, in court (commercial, civil and administrative) and in arbitration (arbitrator at the *Centre de médiation d'arbitrage de Paris-CMAP*). Lecturer in IT law at the *Université Paris XII-Créteil*, and in regulations of industrial maintenance contracts and legal risk management at the *Ecole des ingénieurs du Val de Loire* (Blois) and the *Ecole nationale des arts et métiers* (Paris), she has also contributed to legal workgroups within Syntec Informatique, IFESI, AFSM, and AFNOR. She has contributed to a dozen books on legal risk management, IT maintenance and regulations, IT contracts, as well as articles on new technologies and industrial maintenance, notably on public markets, in numerous reviews.

Jean-Luc ZOLESIO

Jean-Luc Zolesio is the chief of research and innovation within THALES's "*Solutions de Sécurité et de Services*" division. After a thesis in mathematics, he successively worked for IBM, ITT and THALES, all the while teaching, first at the University of Nice, then at the *Ecole Centrale* in Paris. He has been director of a department on exploratory research, technical director of ground and surface radar activity, and director of the THALES *Think Tank*, before taking on his current role. Moreover, he has been laureate of the "*Grand Prix de L'Electronique Général Férrié*" (1993). He is the author of more than 30 patents and has had many of his works published internationally.

Introduction

Today's society is permeated with the notion of systems: electoral system, ticket booking system, air traffic control system, etc. Is this a simple linguistic convention? Or a revival of systemics, perceived by some as the revival of a structuralism which, while formerly praised to the skies, had been brutally disparaged? Or, perhaps, the need to clarify a certain number of concepts and their dispersal within our society, a process accelerated by the rapid spread of technologies?

This book follows this logic, and aims to be a multidisciplinary reflection on "systems of systems", which are currently found in many fields: banks, army, transportation, etc. What should we see in this, beyond the simple repetitive use of the concept of "system"? What makes this new field worthy of theoretical and practical attention? Do we need new tools to manage those systems?

To try and offer an extensive review of the field, this book is separated into two parts:

- "Systems of Systems, Concepts and Practical Illustrations" (Part 1);
- "Systems of Systems Engineering, Methods, Standards and Tools" (Part 2).

Introduction to Part 1

After laying down the definition of a system (it should be noted that this definition includes the system's components and their interfaces, as well as the processes of their respective life cycles, from design to disposal and dismantling, and therefore includes the products and services necessary for these processes) and defining what a system of systems *is*. Chapter 1 ("Systems of Systems: From Concept to Actual Development", Dominique Luzeaux) will set out the ways of

Introduction written by Dominique LUZEAUX and Jean-René RUAULT.

monitoring a system of systems design and, more generally, its life cycle, with particular emphasis on the need for an integrated approach on the level of the engineering process and the use of simulation during the entire life cycle. It will also address the need to adjust the usual balance between general contracting and project management and their contractual relationships, in a context where the purchase of systems must be done in an incremental manner, in time, and in constant co-evolution. Examples will be taken from experiences in the field of armament concerning the management of complex defense systems and program management.

Chapter 2 (“Emergence and Complexity of Systems of Systems”, Patrice Micouin) will shed further light on that issue, first establishing a dichotomy between natural systems and artificial or technological systems, then including systems of systems within the family of technological systems. However, systems of systems distinguish themselves from individual technological systems by their specific formation mode, essentially linked to an initiative of voluntary association for the achievement of multiplied capability. The notions of interface, interoperability and engineering thus take on, if not a new meaning, an increased importance in this effort to control the increasing complication, or even complexity, of artificial systems.

The following two chapters will look at two complementary aspects which are essential for systems of systems. Chapter 3 (“Contractual Aspects of the Acquisition and Use of Systems of Systems”, Danièle Véret) deals with the legal aspects of the contracting stage, paying special attention to the transfer of ownership and intellectual property rights. It helps place the initial issue back within a context larger than the simple technical context, the one addressing economical aspects, and therefore requiring a legal framework. Chapter 4 (“The Human Factor within the Context of Systems of Systems”, Jean-René Ruault) will look at the decision making process in a system of systems from a more sociological standpoint, taking the organizational and cultural aspects into account.

The four following chapters will offer concrete illustrations of systems of systems. Chapter 5 (“The Space Communication and Observation System of Systems”, Frédéric Pradeilles and Dominique Luzeaux) addresses the spatial field; Chapter 6 (“Intelligent Transport Systems”, Michel Chavret) addresses the transportation field; Chapter 7 (“Systems of Systems in the Healthcare Field”, Jean-René Ruault) addresses the healthcare field; and Chapter 8 (“Critical Infrastructure Protection”, Jean-Luc Zolesio) addresses the field of crisis management with large human involvement (firefighters, ER, NGO, police, etc.) including the case of international mobilization (tsunami).

Chapter 9 (“Globalization and Systemic Impacts”, Dominique Luzeaux, Jean-René Ruault and Lui Kam) follows this reflection and addresses two topics: on the

one hand it shows how globalization can be modeled as a system of systems and how some phenomena benefit from such a model, in terms of interpretation. On the other hand, it broaches the possibility of entering new markets in emerging countries, in which we must control the risks linked to a misconception of the market, the potential users, the regulations and the culture, as well as new competition which requires us to keep the upper hand, to offer more complete and integrated products and services or to get those products and services on the market faster and with cheaper prices.

Introduction to part 2

Three chapters will provide the key to understanding all the technical aspects of systems of systems. Chapter 10 (“Methods and Tools for Systems of Systems Engineering”, Dominique Luzeaux) lays down the issue of collaborative working environments and specific engineering tools. It underlines the importance of models in every aspect of engineering work, in particular in the first stages of concept analysis and during the definition of architectures. Chapter 11 (“Model-driven Design and Simulation”, Lui Kam) follows on that work and studies software engineering techniques such as MDE (model-driven engineering, with its model transformation) and complex systems simulation. It shows how these techniques can help find tangible answers to the problems of interoperability, reuse and capitalization, three major aspects which need to be managed when working with a system of systems. Chapter 12 (“Standardization in the Field of Systems and Systems of Systems Engineering”, Jean-René Ruault and Jean-Pierre Meinadier) lists the key standards not only for systems engineering but also for the various data and models exchanged in the course of this engineering (15288, AP233, SysML).

Building on this triptych “theory-illustration-method”, this book, written by ten professionals with various specializations, offers multiple visions on a thriving subject.

PART 1

Systems of Systems, Concepts and Practical Illustrations

Chapter 1

Systems of Systems: From Concept to Actual Development

1.1. Network omnipresence creating a worldwide environment

The revolution brought on by the digital age, which had an impact as big as the industrial revolution, has deeply changed society. As a consequence of the spreading of information and communication technologies, businesses have grown under the influence of a new paradigm, created by the dematerialization of the economy and organizations, giving a new meaning to the notion of extended enterprise. Not only has the existing economical model evolved, but others have emerged, notably with the grouping of enterprises into networks – of different sizes and with different integration modes – to create value. This led to the development of business communities with an increasing outsourcing – and therefore the transfer of direct control via asset ownership – of parts of the traditional value chains in order to favor partnerships.

On a technical level, this cultural change goes hand in hand with the standardization of exchanges, and less centralized management techniques with contractual relationships between independent partners, rather than within proprietary organizations. Controlling these new virtual organizations, knowing how best to use the different resources provided via the networks (whether physical or not), is becoming a competitiveness factor in an ever changing world.

4 Systems of Systems

This new systemic context, moved by smaller and smaller time constants, requires a proportionally higher adaptability, hence the search for increased flexibility and agility in the agents. Let us briefly put into historical perspective the changes that happened during the last decades, in order to study how work methods, enterprises' organization principles and technical tools have led to this context evolution.

The growing industrialization of product manufacturing during the 19th century has led, during the turning point of the 20th century, to the scientific organization of work extolled by Taylor, culminating in Fordism and the industrial production lines up to the 1970s. Vertical integration, which means total control of every link inside a company, from raw material to the final manufactured product, is the main principle of this industrial design. It is represented by a monolithic organization, supported by cascade processes which control the entire chain.

The use of computers has led to increased performances from the former organizations, via the automation of the links whose added value essentially lay in the maximum repetition of a simple activity. The organizational processes haven't evolved, and sometimes have even been strengthened in their integrative vision, helped by tools such as ERP (electronic resource planning).

The introduction of individual work stations to replace big calculators (the famous *mainframes*) has brought a change of paradigm. It opened the way for decompartmentalization of the enterprise, the spreading of models based on transaction and coordination, thus putting into question the monolithic vision based on the neo-Taylorist accumulation of material assets, instead favoring the co-production and collective accumulation of digital assets. The *services* generation was born. The networks and the client server, then the distributed systems, rapidly evolved technically, enabling digital transactions and remote information management, which moved the value chain from the creation and ownership of data to the mediation of data. The technical evolutions in recent years have further confirmed this paradigmatic transition: "information society", "digital divide", concepts which have been systematically referred to for more than a decade, and underline the central spot taken by the ability to provide and access digital data.

Beyond the simple evolution from products to services, those technical progresses have a direct influence on the agents' organization: this evolution requires important investments, and their global optimization can only happen if a group of agents can share some of the expenses and look for productivity gains on the global value chain, via the creation of a partnership. Mediation and negotiation are thus becoming key-concepts in this new deal. For example, with its online Marketplace, Amazon provides millions of associates with the opportunity to use its payment and distribution infrastructure, thus allowing them to create their own

specialty stores while benefiting from Amazon's delivery services and its global network of clients, whom they could never reach otherwise. Likewise, those niches help Amazon's offer grow, since the consumer is more concerned with finding what cannot be found at the cheapest price, instead of knowing who is selling or shipping it to him.

Going hand in hand with the evolution of techniques and economical models, society has changed, with the emergence of new work methods (telecommuting, virtual communities, etc.) and of social links alternatively favoring combination and complementarities depending on the opportunities: the former to increase flexibility and therefore the value chain via the sharing of resources (each part being more efficient when grouped with the others), the latter to favor collective productivity and join a strongly integrated systemic set (the total achieving more than the sum of its parts).

Flexibility, exchanges, partnerships, opportunity alliances are becoming new models, constructed around the constant innovations in the new information and communication technologies. Thus, collaborative work tools, on the one hand, and digital transaction platforms, on the other, have been developed, thereby accelerating the spreading of these new models. These technological innovations are built on gigantic investments, developed in an incremental manner, and paid for by the services that use those innovations. We cannot insist too heavily on incrementality, which becomes natural as the years go by, but is an incessant round trip between fresh innovations and a standardization on different levels of abstraction, so as to enable the control of the widening perimeter while not putting the necessary innovative variability in jeopardy. For example, collaborative work applications will require software components dedicated to the sharing of data and information, sufficiently homogenous to enable the exponential increase of data load, which will in turn increase the need for ever-more sophisticated tools which will have to adapt to certain professional rationales, hence the development of business-modules, which act as new standards within the business communities, etc.

In short, the co-evolution of tools, work methods, organizations and relationships between agents is constant and seems to accelerate, contributing to the increasing complexity of the global environment. This complexity acts as the trigger of the subject we are about to discuss, via the systemic approach of its management.

1.2. Increasing complexity of the environment

The social, economical and political environment has become increasingly complex these last few years, even though this evolution is part of a millennial diachronic dynamics. However, the rhythm seems much faster nowadays. In parallel

with the evolution of technologies and the increasing integration capabilities, the designers' ambitions have evolved within the various fields of application. In the following sections, we will focus on the field of defense, but it is not the only field exhibiting the characteristics of complexity which we are about to study.

1.2.1. *A particular field: defense*

Let us first remark on the specificity of this field's economical environment: the acquisition cycles are very long compared to the technological maturity cycles of the elementary components. From technological study to implementation, a weapons program is developed over a dozen years, and the weapons system itself will be used for three or four decades, depending on updates. To highlight this dilemma, let us compare those delays to the ones of certain embedded technologies (embedded computing, electronic cards) whose life cycle is in the range of two or three years! Moreover, the field of defense is much less competitive than the civil market, and the investment and operation budgets do not follow economical principles. In France, for example, the ministerial budgets (including investments and operation costs) are allocated annually, within a military program regulation which will last for six years and is strongly influenced by presidential decisions, which might therefore evolve depending on the terms of office. Besides, the long-term planning of new systems requirements or system renovation happens on a scale of fifteen or twenty years. From it spawns the inherent difficulty in making the various life cycles tally, against terms of use which are by necessity evolutionary, or even inventive.

A changing international environment, with the need for more advanced weapon systems and limited resources, puts strong constraints on the acquisition of defense systems. The acquisition services' decision-makers must be able to consider future fighting situations when they design new weapons systems, while evaluating their performance and the production capacities so as to minimize the risks, delays and costs. Even if the new information and communication technologies spur technological advancements which make new methods of productions possible, more efficient and less costly, when faced with the financial stakes we have to "product the right product" on the first try.

At the individual system's level, it is therefore necessary to rapidly implement solutions that can be used in a few years instead of 15 to 20 years from now. Incremental implementation steps must be designed so as to mirror the evolution of the various technologies' maturity and take the feedback into account. Only this can help people identify the material solutions which need to cooperate within a highly connected set of collaborative resources. The goal is to find the best global solution, rather than the optimization of a specific system to answer the need. This leads to the consideration of systems as basic components which must be organized within

what can therefore be called a “system of systems”, depending on the desired effect. On the level of the system of systems, we must take into account the asynchronous nature of the acquisition of individual systems (the life cycles might or might not partially overlap), and facilitate the co-evolution of systems in terms of doctrine, organization and training.

The defense systems therefore offer an important disparity on the cycle length between integrating system and elementary components (setting decades against years, or even year quarters), and display a budgetary curve in which high peaks (production stages) alternate with long plateaus (design and development stages, then support). Such a chart is an asset in achieving big ambitions at a given time, but it is *a priori* penalizing when we must continuously incorporate the evolutions of certain components, in particular if the latter have a potential impact on the system’s architecture. As an example, let us point out the changing compromises between hardware components (dedicated electronic cards, etc.) and software (codes, embedded or not, more or less dedicated kernels or even quasi-generic operating systems, etc.) in telecommunication systems or nomad computer systems.

Moreover, the current geopolitical context stresses the necessity of acquiring systems which can interoperate with the systems of other partners: interventions happen *a priori* within multinational operations, within coalitions whose composition changes both in time and depending on the missions. In addition, the systems purchased must be operational within minimum delays (e.g. time is critical as much for humanitarian interventions as for peace keeping operations), and therefore need a reduced staff, easily deployed and with minimum needs for logistical support.

1.2.2. *Impact of context evolutions on the defense systems*

The context we are interested in is defined by the following main characteristics:

- new threats;
- evolution towards a capability rationale (both of design and of use);
- increasing complexity;
- a political context of cooperation.

Let us look at each characteristic in detail, and see what the demands are on defense systems.

Taking new threats (post-Cold War or OOTW – operations other than war, 4th generation warfare) into account calls for high flexibility and *adaptability* in the

exploration of defense system concepts. Indeed, the logics developed during the Cold War concerned symmetrical fights, in which bulks of metal faced off in wide battlefields or in the air; in terms of acquisition, this approach favored continuity of armament, with added technological sophistication in each new generation, but the number of weapons being more important than the individual differences between systems, the length of the acquisition cycles did not have a qualitative impact on a nation's capacity. On the other hand, the new threats highlight the possibility of asymmetrical conflicts: the potential adversary does not *a priori* possess the same off-the-shelf military technologies, but might put the common civilian technologies to inventive use, which would be a threat not on the level of the individual weapons system, but on the level of the opposing army's meta-system. The environment's *evolutionary nature* and the need for the system to adapt to those various evolutions contribute to its complexity.

The evolution towards a capability rationale of the defense tool orients the acquisition process towards systems of systems. This rationale is built on the idea of *operational capability*, which consists of reaching a desired state¹ via the effective combination of resources (staff, materials) and processes to create a set of tasks. Such an ability is expressed in terms of mission types and operation contexts, desired effects (which themselves can be described functionally or through levels of expected performances), doctrines, organization, training, equipment, formation, infrastructures, policies, etc. In the field of defense, an example of this is the "projection ability", which consists of being able to send a certain potential of armed forces and equipment in a certain time across a certain distance on an external field of operation. Expressing the needs in such a way rather than through the numbers of planes with this or that technical capacity (action range, fret volume and weight, etc.) belongs within the rationale of the evolution of products towards services: we can therefore imagine – actually, more than imagine, since the French state has signed such a contract – that to fill the previous capability, a society could rent, on request, aircraft usually destined to carry other types of merchandise, even if this necessitates being able to rapidly modify them, which means that the society offers availability as a service rather than selling aircrafts which the military would then

1. The notion of state is not easy to formally define, for its field of action is multidimensional: economical, politic, sociological, etc. For example, the air bombings on Iraq during the two Gulf Wars, and especially during the second Gulf War, didn't only aim for the tactical destruction of infrastructures, to prepare for the following fights: they were also looking to psychologically weaken the population, and even create a reaction in some layers of society which might have overthrown the established regime. We are dealing with the concept of EBO (*effect-based operations*) and not only with military objectives. Likewise, the current insurrection movements in Iraq (as well as the coalition forces) are seeking effects and not only strict military performances: some want to avoid an equilibrium point after which the country might reconstruct itself, and others want to reach and maintain that balance and convince the population about its benefits.

have to operate during their entire life cycle, and therefore maintain, repair and upgrade. In the civilian world, the same rationale means that United Parcel Service has transformed from a company delivering packages to the supplier of a global supply chain: thus, United Parcel Service has integrated certain parts of the repair chains of about 100 industrial partners, for example Toshiba computers or HP printers, which, for the user, heightens the quality of service in terms of delay, while allowing the involved businesses to concentrate on activities with a higher added value compared with their practical knowledge. According to some figures, 2% of the world's gross domestic product is going through United Parcel Service!

The ability to map the links between any type of resource and their use with quantifiable effects heightens strategic flexibility, since the capability is *a priori* unvarying against a specific context's requirements. The difficulty is obviously to determine capabilities, for, on the one hand, each of them must allow for tight couplings between the implemented components, and on the other hand the coupling must be as weak as possible between different capabilities. Moreover, if the capability shows the final purpose, the trajectory from the current situation to that purpose must be defined – through the definition of capability increments, and more generally through a true process of capability management, going through these three stages:

- definition of capabilities: conceptualization of new ideas on capabilities, with ideas on planning, acquisition, development, management of the life cycle of the staff, materials and processes potentially implemented within the capability. This study happens on a long-term basis, with a range of a decade or longer;
- capability maintenance: guarantee the level of availability of the capability, which happens on a medium scale, with a five year horizon;
- capability usage: plan and pilot the operations which use the capability. This obviously happens on a short term basis.

Therefore it is not only about switching one system for another (for example a fighter aircraft for another fighter aircraft), but working on the complementarity of certain weapons systems depending on what they are supposed to achieve: this is how the question of redundancy can be pondered between a tank and a helicopter, and how the necessity arises to renew them both in an independent fashion, both being antitank weapons. On a more general level, this exercise can be accomplished by comparing systems, then aggregates of systems: in fact, instead of following a product rationale, where ageing and replacement are planned, we favor a desired effect rationale, and the product base is adapted accordingly. This helps scale down, and also achieve a much more flexible defense tool, since replying to a new threat becomes an exercise in system architecture, with people trying to organize, as well as possible, the weapons systems or their off-the-shelf disposition, and only then try

to acquire what is necessary to cover the unachieved effect. Such an enterprise raises the issue of the *integration, harmonization and interoperability of a system within a higher-level system*.

When it comes to operational use, this capability rationale translates into the possibility of optimizing the implementation of resources (the available systems), within organizations which are not necessarily fixed, and are often determined on a case by case basis (within coalition operations, for example), to perform a set of actions and achieve the desired results. This notion is the same as the notion of an extended enterprise, in which the human agents of a sociotechnical group must use the available resources in order to collectively perform a certain number of actions, contributing to the creation of group value. In fact, the study of the transformation process, which has been carried out within NATO for a few years, shows that this is more than a simple analogy, and this aspect therefore deserves some attention.

The defense systems are, moreover, increasingly complex, for they integrate a higher number of heterogeneous components and life spans which vary greatly. Classical examples are in the range of 30 to 40 years for military aircraft, for example, or for armored vehicles, against a few semesters for some electronic or transmission equipment. Frequent renovations due to the obsolescence of the subsystems, which are more and more often built on civilian technologies, require control over system architectures whose component configurations are not known. Indeed, the problem of a component's obsolescence is not just a question of being able to replace said component in a one for one model, for such a replacement might put into question larger architectural choices: times when dedicated maps were the optimal solution to increase performance alternated with times when software got the upper hand; this back and forth movement of the current level of technological maturity represents the challenge of the hardware/software co-design issue. Likewise, implementation tricks used to heighten performance one day might turn out to be penalizing later on, or even put the system's operation into question, for example when a physical support of this implementation is altered. Complexity therefore is not an instantaneous property which must be taken into account at a specific time, but has to account for all potential future system evolutions.

Risk reduction and management through the various stages of a weapons program (upstream of feasibility through to deployment, or even to disposal, while being increasingly aware of environmental requirements) then becomes a crucial step in achieving *cost control* all through a program's life. In addition to this complexity, which manifests during the acquisition and maintenance of the system, challenges also arise during the actual use: often, human agents operate in close contact with "artificial" systems, as operators, supervisors or decision makers. This interaction must be taken into account within the global architecture: this heightens the complexity of systems when taken as a whole.

Finally, the political context, with the reduction of defense budgets and the building of a European Defense Policy, demand acquisition costs to be controlled, and developments to be shared between European partners. A nation can no longer acquire all the desired capabilities on its own. Furthermore, as has already been underlined, crises are, with increasing frequency, often solved within coalitions, something which requires and introduces a certain interdependency. It is then that *reuse* takes on its full meaning.

We do not want to conclude this first analysis without mentioning the civilian community: it should be highlighted that flexibility, the need for an evolutionary nature, capability rationale, the multitude of components, are just as critical for civilian systems as they are for defense and security systems. However, one point varies between them: military systems are frequently deployed within largely unanticipated contexts (*a priori*, no trustworthy market study exists to target deployment niches!), because of the variability of military operations in terms of type, scale and duration.

1.3. Towards a definition of the concept of system of systems

We will now define the various concepts which help resolve the previously exposed issues; we will then detail the main steps needed to go from these concepts to their actual implementation.

1.3.1. From system to system of systems

1.3.1.1. Systemic prolegomenon

Since systems will be the main subject of this book, defining their various terms must be attempted. The main standards (from the MIL-STD-499B to the more recent ISO 9000: 2000 then ISO/IEC 15288, last standard dating back to 2002, through the EIA/IS-632, ISO-12207, SE-CMM) define a system as:

“an integrated set of components – workforce, products, processes – connected to each other so as to meet one or more predefined objectives.”

The important points of this definition are the existence of elementary components (of varying nature: hardware, software, virtual, human, etc.) and the relationships between them. Biologists (Atlan, Bertalanffy, de Rosnay), epistemologists (Le Moigne), economists (Lesourne, Passet) even counselors in business management (Donnadieu, Yatchinovsky) or sociologists (Morin) all agree

that the stress should be put as much on the relationships which connect a system's components as on the components themselves. To quote Ferdinand de Saussure,

“a system is an organized whole, made of interdependent components which can only be defined in relation to one another depending on their place inside this whole.”

The system's properties would depend on the relationships between its parts more than on the nature of these parts. It should be noted that such a reversal of perspective has led in mathematics to category theory, which notably distinguished itself from set theory. It is not just conceptual, for even though similarities between both approaches can be demonstrated, doing so calls for the use of certain hypotheses which, despite being held as acceptable for most of common mathematics, are ontologically speaking, by no means insignificant. Only through this reversal of perspective, ontological primacy of the relationships over the explicit object, can a superior integration *a priori* be attained. This is what sociologist Edgar Morin underlines when he declares that we should “go from organization to interaction”.

It should be noted that these relations have no reason to remain stationary through time, which helps us to envisage temporal loops between components, but also evolutionary topologies (architectural forms for example) of these loops. Incidentally, it is this plasticity that will have to be tamed in future applications in order to profit from its very essence.

Even though, etymologically, a system is a set (σύστημα is derived from “put together”), the previous definitions do not stop at this structural vision but also insist on the teleonomy. Certainly inspired by the epistemological current, which advocates the importance of determinism and causality, taking the notion of purpose into account within the system's evolution is however crucial for the engineer. In addition to the dynamic aspect – beyond the necessary temporal vision, this means the structure does not remain stationary and a process point of view is also necessary – it puts forward the relativism inherent in any description, and the necessary identification, on the one hand, of the purpose which represents the system's *raison d'être*, and on the other hand, of the system's immediate environment, in relation to which the system evolves and with which it necessarily interacts.

The broadening of the definition to include systems of systems then seems trivial: we only need, following a well-known recursive scheme, to use systems as elementary components, which leads to a system of systems of systems, as said by Edgar Morin to qualify life. Reversing the analysis, Edgar Morin goes as far as quoting S. Lupasco: “there only really exists systems of systems, the simple system is only a didactic abstract.” But things are not so simple, and if the epistemologist

does not consider this distinction as important, the engineer does. This is why some people differentiate families of systems from systems of systems. A family of systems would be, according to the definition of the United States Department of Defense (US DoD) acquisition services, “a set of independent systems that can be arranged or interconnected in various ways to provide different capabilities.” We might call such a structure “complicated”. In that definition, the term “set of systems” is important, for it strips the organization *a priori* of individual systems of its possible relational characteristics.

A system of systems would go beyond the notion of a family of systems, “the loss of any part of the system will degrade the performance or capabilities of the whole”, still according to the US DoD. In addition to teleonomy, which we have previously mentioned and can now see making a discreet appearance insofar as the definition uses performance in relation to the intended purpose as the decision criterion, some see in that definition an allusion to the concept of *emergence*: since the system of systems authorizes emergent behaviors that cannot *a priori* be observed on the level of its separated individual components, any partial amputation leads *a posteriori* to an observable behavioral discontinuity. A system of systems might include a family of systems, which might be put to other uses, although this would happen to the detriment of that emergence.

Metaphorically speaking, we have gone from a Lego-like construction to something more akin to a living organism. The *complicated has become complex*, something which, in its dual etymologies of “embrace, include” or “weave together”, underlines this double aspect of interaction and integration to aim at the emergence of a new global property. As a reminder, complex systems, as they are used in the scientific community, present a large number of interacting entities, exhibiting structures of various scales, and covering natural systems from cell to ecosphere through anthills or other societies of social insects, as well as sophisticated artificial systems such as communication or energy infrastructures. These systems often result from evolution and adaptation processes and feature emergent properties: the underlying microscopic level brings out the organized forms on the macroscopic level, which in turn influences the microscopic level. Local and global interactions can thus be combined in the description of their dynamics.

Beyond their intellectual interest and their use in delimiting the field of observation, these various definitions are however poorly operational. The richness of the concept is only revealed through the use that is made of it. Moreover, the semantic ambiguity cannot be avoided, and it would seem the main factor differentiating the systems, the systems of systems – and all possible intermediate concepts – is the scale factor: in all cases, designers and users will have to master the transitions between the *local* and the *global*; the approach characteristic of

systems of systems is not *a priori* drastically different from the approach characteristic used on a system. However, a difference lies in the way some decisions, whether they concern the designers, the architects or the users, will have a much more important impact in the case of systems of systems.

As is stated in the case of systemics applied to business management, in order to constantly adapt to the modifications of their environment, the systems of systems must be inventive and plastic: it is a flexible organization in which nothing is set, so new ideas can be rapidly implemented; where the set of agents have sufficient knowledge of the orientations in order to contribute with their ideas; which develops internal and external networks, and therefore allows for cooperation and mutual support rationales, which facilitate reactions, interactions, and retroactions in the face of the environment's fluctuations.

1.3.1.2. *Definition of the system of systems*

The notion of system of systems first appeared as an identified axis of study in the early 1990s, in the fields of defense, air traffic control and information technologies. If nowadays this term is frequently featured in literature, it has no universally recognized definition (consult the appendix for a non-exhaustive list of definitions, gathered in open literature); it has however been adopted by everyone in the field of defense.

We can, however, identify a convergence towards some of the characteristics identified by Mark Maier in 1996 and rather largely shared among the international defense community. These characteristics allow for the distinction to be made between systems of systems and complex monolithic systems: it should be noted that the distinction lies not in the “complex”, but mostly in the “monolithic”. The characteristics are as follows:

- constituent systems have an operational independence;
- constituent systems have a managerial independence;
- the definition and configuration of the global system are evolutionary in nature;
- emergent behaviors of the global system exist;
- constituent systems are geographically distributed.

The operational independence of the constituent systems means they must be able to fulfill an independent mission on their own when the system is taken apart. This means that each of these systems is independent and has a use of its own (example of the naval military force and its components such as aircraft carriers or planes).

The managerial independence of the constituent systems means they are acquired separately, then integrated to create the system. They have a separate existence and organization, in their acquisition as well as their maintenance. This concerns the aspects of project and/or maintenance teams, as well as budgetary considerations.

Since the system's definition and configuration are evolutionary, it means the global system is not set in stone. Its development can be incremental (the constituent systems are not necessarily available at the same time) and/or evolutionary in time (functions can be added, erased or modified depending on the acquired experience or the evolution of needs). Moreover, each constituent system can go through its own life cycle, distinct from the system's.

The emergence of behaviors means that the global system features properties and functionalities not present in any of its constituent systems, since those emergent behaviors cannot be located or attributed to one of those constituent systems. Moreover, these behaviors cannot always be anticipated, which results in difficulties in validating the global system. An example of emergent service is IP (Internet protocol) routing, which is at the root of the Internet. No IP router knows the complete topology of the Internet's interconnections, or even the configuration of the local interconnections in its own neighborhood. Since the configuration of the links between routers is constantly changing, as is the bandwidth on a given link, the routing tables are constantly referencing a past situation. And yet, IP routing is an efficient process which can be trusted, and which transfers in a predictable manner source messages to the expected destination. Each IP router standing in the message's path decides which nearby router will constitute the next stepping point, even though it has no knowledge of the routers or the potential paths accessible in its immediate neighborhood. IP routing is therefore an emergent service, which also works with incomplete, imprecise and obsolete information, while providing an efficient and predictable functionality.

The geographical distribution of the constituent systems means that they are located in different places; this geographical extension is relative and largely depends on the available technologies and communication means. The systems can exchange information, but cannot exchange substantial quantities of energy or matter. This last restriction helps confine the studied perimeter.

Maier's definition is one of the most widespread definitions, even if a certain number of variations can be found in literature (see section 1.11. Appendix). However, it features a double defect. First, it only grants a *raison d'être* to the system of systems' constituent components, without mentioning what the system of systems brings in relation to its constituent elements, apart from the possibility of emergent behaviors. Secondly, it is restricted to the technical criteria of geographical distribution and possible new behaviors, even if it broaches the capability aspect;

however it only does so from the angle of sought effects measured in terms of technical performances. It is, moreover, very much oriented towards the systems' acquisition stage, and in particular neglects all notion of organization of technical and human resources during use, as is the case, for example, in extended enterprises, but also in the context of defense, as we have previously pointed out with the concept of transformation.

It should also be pointed out that in recent presentations, Maier went back on the two criteria of emergent behavior and geographical distribution, in particular since it is difficult to observe and exploit emergence in practical applications, and the geographical distribution is becoming common for many systems with the development of the information and communication technologies. These two criteria are therefore not really discriminating in differentiating a "system of systems" from a simple set of systems. This leaves us with the criteria concerning operational and managerial independence in addition to the management of a complex configuration. It is clear that this is not enough to correctly define the concept, even more so since independence is open to criticism in systems which, by nature, are increasingly intended to be coupled, whether they are software-intensive or not: by nature, the design of equipment in the various fields, civil or military, is now trying to anticipate the interfacing with other systems, or at the very least standard infrastructures, even if only because experience shows that many systems are destined to become components of systems of systems! The argument is, indeed, fallacious, and the reasoning circulatory, but how many (high technology) objects of everyday life are not equipped with plugs or multiple interface capabilities, in the hope that they will be used in another context rather than thrown out because they could not fulfill a need that was not known at the time they were put on the market?

This leads us to offer the following new definition, which is more general than Maier's, and which covers *a priori* all the variants found in literature, as well as the variants which concern extended enterprises, while remaining functional, as we will underline in the upcoming sections.

Definition: a *system of systems* is an assembly of systems which can potentially be acquired and/or used independently, and whose global value chain the designer, buyer and/or user is looking to maximize, at a given time and for a set of conceivable assemblies.

In this definition, we follow the general definition of the "value chain" made popular by Michael Porter, namely the set of interdependent activities whose pursuit creates identified and, if possible, measurable value (which means the customer is ready to pay to get the product or service). This therefore includes every stage, from the purchase of raw materials to the actual use, or even the after-sale service if necessary. The value chain's efficiency essentially relies on the coordination of the

various agents involved, and their ability to form a coherent, collaborative and interdependent network.

The concept of a value chain provides all the necessary variability to understand the difference with a system, and considers the abstraction in contrast to a simple technical vision. Moreover, the concept is rich insofar as it authorizes spectacular flips of reasoning: in theory, each customer and/or user community can have its own system of systems. Indeed, each of these communities has its own use contexts, its own priorities, constraints, roles. There are therefore manifold opportunities to create value by reinventing the same community we are trying to create value for! If this reversal holds few benefits in the field of defense, it is however useful in other fields where the notions of market shares and positioning in relation to potential customers are essential, and where supply can create demand.

The above definition evidently meets Maier's criteria: the managerial and operational independence is present since we mention the independent acquisition and use as basic hypothesis. The configurations' evolutionary nature is taken into account since the system of systems is an assembly that changes with time, since we talk of optimization at a given time and for a set of assemblies possible at that time, which opens the door to all possible evolutions, knowing that the system of systems also undergoes radical changes as the local solution of an optimization. The spatial distribution of systems should be noted on the level of the considered assembly. As for emergence, likewise defined as a global property not directly inherited from the constituent systems, it can be expressed *a posteriori* via a performance (*a priori*) of the value chain that must be optimized.

To sum up, the introduction of the value chain in the definition, and the requirement for its optimization in a potentially changing context, is the key to defining what a system of systems is in relation to constituent systems. With the following examples (taken in the civil and military fields), we will illustrate how different chains of value help characterize with precision what we intuitively perceive as systems of systems. In particular, we will study value chains which are not necessarily predominantly technical, something which will allow us to insist on potential added values of systems of systems other than their simple technical performance. Following the usual breakdown of business management into three variables – *people*, *process*, *product* – we will study value chains which are predominantly organizational, functional, and technical.

However, let us point out that the benefit of a value chain is precisely that we do not have to restrain ourselves to one of these particular aspects, but can combine them. The following presentation is deliberately oversimplified in order to be didactic, and the reader will see that the line between these aspects is very thin.

1.3.2. Examples for various value chains

1.3.2.1. Technology-intensive value chain

1.3.2.1.1. Network-centric operations

Adjusting to the fundamental changes happening in our society and the world of finance, changes brought on by rapid technological evolutions, the military field is going through an important *transformation*. For a few hundred years, the fighting tools and techniques have mirrored the evolution of military technologies: whether it is with ranged weapons in ancient times, then artillery, more recently the introduction of aircraft and armored vehicles, and finally high precision missiles. Today, however, we can consider that there is not only an evolution of weapons, but a true break can also be contemplated: the proliferation of and free access to some technologies, their apparent simplicity of use and at the same time, their sophistication if we are looking to achieve optimized use, place us *a priori* at the dawn of a transformation akin to the transformations marking the end of the feudal era and, much later, the beginning of the Napoleonic era. Indeed, at the end of the feudal era, the organization of society itself was called into question after the introduction of destructive weapons meant to be used by knights but which could be operated by mercenaries or any member of society. Likewise, the Napoleonic era was marked by the concept of *levée en masse* (mass conscription), which was a break away from the maintenance of a small professional army, and enabled the use of the entire male population within a certain age bracket, taking advantage of the growing industrialization of society which enabled this *levée en masse*.

For a few years, we have been witness to the thundering apparition of new operation concepts, called network-centric – better known as NCW (*network-centric warfare*) – which originate in the following paradigmatic shifts, easily observed in society:

- from the platform (or more generally the physical node) to the network, a shift which deeply modifies the technical value chain, via the interconnection of systems and therefore the sharing of information;
- from the individual agent, regardless of his environment, to the vision of an ecosystem in constant evolution, developing the importance of strategic choices and evolution strategies within evolving ecosystems: here the functional, or even organizational, value chain is the one most deeply impacted.

These changes obviously have their roots in underlying technologies (boom of the Internet, intranets, extranets, etc.), in terms of infrastructures, products but also and mostly the associated services. From these changes spawned a new view on information as an asset, with its added value no longer in the jealous protection of information, but rather in the supply of the right information at the right time. This

implies distribution capabilities, on top of the simple processing of information. The goal is to achieve a more flexible, dynamic system, two key elements in this new context: the former is fundamentally oriented towards a better survival within changing conditions, and the latter towards an increased adaptability; the purpose hasn't changed, we must draw the right conclusion first, in particular in contexts where the first drawer holds the best chances of surviving until the exchange!

These network-centric operations constitute a new set of associated military concepts and capabilities, which help fighters take full advantage of all the available information, and use the available assets quickly, flexibly and complementarily. The operations thus find coherence in an informational superiority which helps political authorities and military commandment set their own rhythm and grasp at every opportunity, regardless of the operation nature. The key ideas are:

- a force organized into a robust network improves information sharing;
- a large distribution of the information improves the quality of information and helps share a common tactical situation;
- sharing a common tactical situation enables the collaboration and synchronization of the opposing armies, and helps speed up the maneuver;
- a more efficient mission.

On a technical level, numerous technologies are implemented: deployment of physical networks with hardwired infrastructures and mobile relay stations; evolution of the systems in order to communicate with other systems within upstream and downstream connections, this being valid for embedded systems on aerial, terrestrial or naval platforms, as well as for the staff in the field, with all the integration constraints (weight, cumbersomeness, autonomy). All this entails development of processing and fusion algorithms to calculate and present to all agents the useful synthesis of the gathered information, and manage their updating. Notwithstanding some discretion requirements typical of military use, most of the ingredients of common civil applications can be found there. But beyond technological developments, we also have to make these things evolve:

- the doctrine, the organization and the command, for we do not organize and use an army in which all soldiers have a deep knowledge of their tactical environment, including possible enemies, the same way they would use an army in which soldiers have access to no information apart from what the chain of command tells them;
- the capability components, that is to say the various systems, insofar as the systems' interconnection permits new use combinations (see *infra* the example of zone control and surveillance).

All this is to be done within a vision of commitments taken among varying coalitions, which requires the creation of interoperability requirements, whether it be on the level of protocols and data exchange formats, enriched data or the high-level information which is deduced from the data.

We would be right to wonder whether all these ideas are not just reformulations of well-tried concepts, with a possible improvement of some performances, in terms of reaction time or the quantity of information shared. In fact, it is rather obvious that if the distribution speed of the right information eventually depends on this simple performance improvement, the situation is different when it comes to the need for heightened flexibility: the goal is to achieve the ability to completely reorganize, on all levels, in order to meet the wishes of the supreme commander. This is truly new, in particular concerning military organizations, for it goes, *a priori*, against the established top-down control, e.g. from the commander to the troops. The constraints that come with such a static, top-down vision are obvious: since each component has its own operation rhythm, its own spatiotemporal reference, it would be illusory, or even dangerous in certain situations, to try and force every component to follow the leader's rhythm. This hinders the ability to perform brutal accelerations, and instead defines a specific maneuver which might shape the enemy according to Napoleonic design, so as to lead him to the desired configuration (seeking the most important impact on the enemy, for example the biggest loss, and not necessarily the most important gain for the ally). This way of thinking has become widespread in the world of business management, and has also partly proven its worth in certain recent conflicts.

The admission price into these new concepts is high: on the technological level, it requires dense and high-performance information grids, in terms of sensors, processing capacity, decision and transmission, and finally the tight coupling of actuators with the previous elements. Beyond the appropriate technologies – which are largely available today, and constantly developed, the concept of *cooperative commitment* is emerging: it results from the combination of a high-performance sensor grid, and a high-performance grid of actuators, with short detection-decision-action loops, where the goal is to optimize the actuators' answer depending on the information detected by the sensors (beware: the actuator may be on a whole other system than the one sporting the sensor, on paper the concept is natural but picture yourself on a platform under enemy fire, where the best answer is to wait for a partner to move into action!). Cooperation results from the existence and the exploitation of the grids, in which the whole is supposedly more efficient than the elementary parts. It is then possible to think in terms of desired effects, which the cooperative commitments will implement. The added value is easily conceived; we have just crossed from a local and reactive point of view to a holistic and deliberative point of view, a well-known gain.

The price of admission is also high on the organizational level: the training, organization, and resource allocation processes are called into question. This is usual in enterprises, in which adapting the means of answer to the situation is common practice, even if it asks for a total reorganization of the enterprise. It may seem logical, but upon closer inspection, an organization built on its current missions can change a lot, in particular when the missions are not systematically defined with precision!

Let us come back to the economical world analogy: for a project, it is common practice to form a team on demand, by finding the appropriate skills for project management, technical trades, financial follow-up, etc., or sometimes by teaming up with other partners within co-contracting. The parallel is easily established in the military field: the forces have to be designed to achieve the desired effect, by fetching the necessary capabilities from the off-the-shelf stocks, stocks which might belong to partners within the coalition, etc. Easy to say, but harder to implement in organizations which are traditionally rather rigid..., in particular concerning commandment prerogatives and hierarchical levels! This creation of forces as a service is a real change of perspective, and leads to a radical transformation within the armed forces. For let us not forget that – like a company which doesn't have one single project but must instead manage several simultaneously – it is common practice to have missions to lead in parallel and within potentially very different coalitions: we need only look at each nation's intervention fields.

This organizational aspect, which is at the base of the transformation process much discussed among NATO's partners, is not only theoretical. As an illustration, in another field, its tangible effects were measured through the reorganization of public security in New York, where the crime rate has clearly diminished, following an in-depth reorganization of all the concerned services within competing processes. The same approach has prompted some initiatives taken during the summer of 2004 in order to develop a fire surveillance network in the Southwest of France, where agents from varied horizons (firefighters, local associations, volunteers, etc.) were integrated for the needs of the mission in a structure of common command with the establishment of a shared informational situation, which helps achieve a better coordination of the forces present in the field.

Beyond the organizational aspect, there is also the one of cost. This is not only a trend, but a sad reality: everything has a cost... including human life. And every constraint of cost is in direct relation with a constraint of time, and therefore of a performance to provide at a given time, leading to an intervention capacity more or less diminished, and therefore more or less risky. The global cost of a system and the cost of its use for a mission should therefore be considered in unison. The first takes into account the design and production, but also the entire in-service stage, which means maintenance, training, management of obsolescence and necessary

evolutions, as well as disposal. The second corresponds to the formerly evoked notion of service, where forces are built according to the desired effect. And, as demonstrated by some recent interventions, the cost may become recurrent, notably when interventions last in time.

Let us illustrate this through two operational capabilities.

1.3.2.1.2. Anti-ballistic missile defense

A capability which has been much discussed in recent years is the *anti-ballistic missile defense*: it concerns the protection of a theater of operations (for some countries, the protection of the territory is also considered) against threats of ballistic missiles. Such a protection comes under the notion of capability, and its implementation will require many various systems, dealing with the detection of the threat, then the tracking of this menace so as to calculate its trajectory, the decision process, and finally the treatment of the threat and implementation of possible weapons to destroy it. Let us go through each step with the added light of considered threats: ballistic missiles with a range of 400 miles following Keplerian flightpaths after their propulsion phase, flightpaths whose peak is equal to about a quarter of their range. The missile is therefore sub-orbital for a short duration after its launch (averaging out to a minute) and shortly before its impact (likewise, averaging out to a minute). Its detection demands the use of sophisticated spatial tools (infrared system of early satellite alert) or high-altitude tools (airborne infrared alert system). It should be noted that the global flight time is in the range of a few minutes, which accordingly puts strong pressure on the detection systems. Besides, the detection is only preliminary to the pursuit and tracking, *a priori* through surface-to-air radars, where the calculation of the trajectory is a key issue for the following decision-making. The latter is potentially dependent on the highest military and/or political echelon, which implies transit through information systems, and the need for prompt confirmations in the light of the temporal constraints. The last step is the final tractography, and depending on the decision made, it specifies the treatment of the threat through the implementation of anti-aircraft weapons, or adapted interception missiles.

We can see the complexity of the global technical architecture, as much in terms of the number of systems potentially put to use, their geographical disparity, the use contexts which can be within coalitions and therefore require the implementation of international cooperating systems, as in terms of the transfer of data, the whole being extremely constrained by time and information security. As the threats evolve (from fixed to mobile missile launchers, then from a single warhead to several, then missiles without a ballistic trajectory, etc.), we cannot acquire anew all the components of detection, treatment, communication, decision, tracking, neutralization, etc. Even more so since some of the constitutive systems are *a priori*

so-called major programs, with a heavy budget, and must therefore be put to different uses to justify the expense... (without talking about economical profitability, everyone knows a military program is not as easy to launch as it was a few decades ago!). The systems' individual logic, or at least the logic of the various capabilities which might call onto them, can very well be poorly compatible in pairs. Designing a system architecture is made all the more delicate by it. Added to this is the fact that the architecture's optimization – let us not forget that the technical criterion is not the only one to be considered! – can necessitate the implementation of systems into particular geographical zones which might cause problems on the political level, be it domestic (see the antinuclear protests in France a few decades ago) or foreign (see the current polemic on the installation of surveillance radars in certain European countries). This is truly a system of systems issue, in which the multidimensional character is fundamental, even if the technical component acts as the foundation.

1.3.2.1.3. Zone control and surveillance

Another interesting capability is *zone control and surveillance*, whose objective is to control, through predominantly air-ground means and minimum numbers, any action judged inopportune, harmful or even hostile, located inside a zone: by noticing and monitoring any activity that could be an opposition to the global objective, by provoking effects against the persons or systems deemed undesirable. This is a true organized system of systems, commanded and controlled, composed of means of surveillance and identification of activity within a determined zone, directly or indirectly linked to weapons with variable effects, working at a distance or deployed on the zone itself. These weapons are associated with the system depending on needs. Control can be enforced either locally in a sector occupied by allied forces, or at a distance, on an unoccupied sector. The means of action must allow the control of enemy activities in the “coercion” and “control of violence” operating modes, within symmetric or asymmetric commitments. The zone of operation, generally incompletely occupied, can reach lengths and widths in the range of dozens of miles.

The difficulty inherent in this tactical issue can be seen in recent conflicts where, independently from ideological positions, the technical difficulty is instantly noticeable. Which sensor(s) should be used, whether it/they be technological or human, and where to arrange it/them? How should the weak signals which it/they may collect be gathered and translated? Which information should be transmitted to the various points along the command chain, keeping in mind that the mesh network should not be clogged if we want to identify the strong premium of weak signals (individual movements or suspicious movements of vehicles, instrumented kernels of popular discontent, etc.)? Which targeted action should be taken when faced with certain situations, in order to avoid insurrectional contagion? How can the technical

value chain be optimized in order to take advantage of all these technical resources, whose interconnection and complementarity within a flexible architecture allow, *a priori*, an important improvement of performance, as long as we know how to define it?

1.3.2.2. *Predominantly functional value chain*

The intelligent transportation systems (one should say the intelligent transportation systems of systems, as we will see in the following paragraphs) cover a large set of potential applications, from the improved communication between public transportation services to automated highways with fully automated car fleets. They aim at optimizing the global management of vehicles, transported loads and transport lanes, while improving security, reducing vehicle use, and lowering transportation times and fuel costs. Many countries are taking an interest in them to try and solve the problems caused by traffic congestion, which is constantly amplifying worldwide as a result of increased motorization, urbanization, population growth and changes in population density. This congestion reduces the efficiency of transportation infrastructures and increases travel times, air pollution and fuel consumption. Moreover, simple technological improvements on the level of each contributing component (infrastructure, vehicles) are no longer sufficient.

Recent governmental actions, especially in the United States, are also motivated by the perceived need for homeland security: many of the systems proposed for intelligent transportation include road surveillance, a priority in homeland security, and may play an important part in the rapid mass evacuation of people from urban centers in the case of high casualty events such as natural disasters or other threats.

The systems used in intelligent transportation systems are extremely varied: car navigation systems, traffic signal control systems, variable message signs, automated radars, video surveillance, container management, systems that integrate live data and feedback from many other sources such as weather information, bridge deicing systems, etc. The technologies implemented in these various constituent systems include: short-range wireless communications (a few hundred yards, via Wi-Fi protocols such as the IEEE 802.11) or longer range (WiMAX, GSM, 3G), embedded electronics (data computer, GPS, mobile Internet), infrastructure sensors integrated within road infrastructures (video cameras, weather sensors, inductive loops counting vehicles by measuring their magnetic fields) or coupled to the drivers' cellular phones and providing floating car data (also called floating cellular data) in real time, electronic toll collection via RFID tags or automatic number plate recognition, etc.

Among the new functionalities, let us point out real-time traffic flow information, and the updating of options to get to a specific place, in terms of transportation means and itinerary to follow. Other functions concern the actual

organization of the transportation means and require real time and predictive estimations of the travel time between certain points, as well as statistics on the travelers' start and finish points and the paths they take. By furthering the integration, we can imagine functionalities of automated transportation with the system taking hold of the vehicle by inserting it in a monitored convoy (what is sometimes called "Automated Highway System").

It is clear that the implementation of this kind of function asks for an interconnection between many systems, whether they be software-intensive information systems (but let's not forget about all the hardware physical sensors which provide them with the necessary information) or physical systems such as the vehicles themselves and their localization, navigation and mobility devices. Moreover, the interconnection must happen with the utmost flexibility, through the use of all the various managed systems, for heavy infrastructures (sensors, communication links) dedicated to each vehicle, or an architecture dedicated to each vehicle, are out of the question and could only lead to an overload of the entire system. Cooperation and the layout of the set of components and available information are then essential in fulfilling the desired function.

This example of an intelligent transportation system is only one of many illustrations of the interest of an optimization of the predominantly functional value chain. Other examples could be the air transportation system of systems, from the order of e-tickets, to passengers or luggage transportation, air traffic control (with radio towers, radars, weather stations, control towers, control of civil and military, national and international flights), equipment and infrastructure maintenance, security services, in-flight meal services, luggage reception on arrival or during connections, to the booking of rental vehicles, hotels, package tours, etc.

The healthcare field, with the hospital services, the doctors, pharmacists, financial support services, the management of the medical records, provides us with another example in which the emerging function is the service of patient care via an interconnection of the various agents, while trying to be as transparent (in terms of process flow) as possible towards the user. As an illustration, we will now study a crisis management system, taken from the SafeCom program, based on a study of the NCOIC (*Network-Centric Operations Industry Consortium*). The scenario consists of an accident involving a patient who must be evacuated to a medical unit. Several means of medical emergency can be used, depending on the initial medical evaluation and the potentially evolving condition of the patient. The medical unit also depends on this data.

The implemented means are: an emergency call center, an emergency medical team aboard an ambulance, a doctor, sometimes a helicopter, a hospital. The emergency call center provides the following services: notification of the scene of

the incident to the emergency team, the fastest way to get there, notification of the doctor, communication with the helicopter in case of airborne evacuation, notification of the hospital about the patient's status. The emergency team aboard the ambulance provides the following services: reconnaissance of the scene of the incident, and transfer of that data to the call center along with the first diagnosis, transfer of the exams and medical data (electrocardiogram, cranial image, etc.) to the doctor, transfer of the patient's status to the hospital before their evacuation. The doctor provides the following services: he advises the emergency team as to which exams to perform, and sends the diagnosis to the hospital. The helicopter transmits the patient's vital stats to the hospital. The hospital warns the call center of this notification.

The integration of these systems creates new capabilities on the level of the system of systems: optimization of the medical and emergency resources; reduction of the global treatment time, which means a decrease of the mortality rate per accident. These new capabilities rest on the implementation of new functionalities: medical evaluation (telemedicine, teleradiology), transfer of adequate and optimized resources to the scene of the incident, dispatch of the patient to the adequate, available, hospital, depending on their needs and the availability of the emergency services, quick evacuation of the patient towards the chosen hospital, preparation of hospital resources depending on the patient's diagnosis.

The challenge for this system of systems is to define the architecture that will best fulfill these functions, and therefore the aforementioned capabilities.

1.3.2.3. *Organization-intensive value chain*

The introduction of electronic networks in the added value chain completely disrupts the structures, whether they're producing goods or services, and forces people to rethink organizational methods, thereby becoming a strategic factor of inflexion. Look at Toyota's just-in-time model, which helped Japan conquer the American market despite seemingly invincible giants such as General Motors, Ford and Chrysler. The world's distribution leader, Wal-Mart, is another example from the last decade: they organized the various positions and trades around an information network, and perfected the integration of common logistics. We will study this example of systems of systems in more detail; here, the value chain is predominantly organizational, even if the functional and technical aspects evidently matter.

Wal-Mart's chain of department stores was founded in the United States in 1962 and has since then become the world leader in the retail branch. First selling discount non-alimentary products, Wal-Mart started selling alimentary products in 1990, thereby attacking a business sector in which its competitors had long been established in the American market: Kroger since 1883, Safeway since 1915,

Albertsons since 1939. Wal-Mart then opened so-called supercenters, twice as large as its previous stores, with two different areas for alimentary and non-alimentary products. If Wal-Mart's growth rate was good between 1980 and 1990, going from 1.2 to 26 billion dollars, it then became exceptional: 94 billion in 1995, 233 billion in 2002, 288 billion in 2004. The number of supercenters went from nine in late 1990 to 888 in late 2000 and 1906 in late 2005, with an average of four new supercenters opening every week between 2001 and 2005. The rhythm even reached a rate of five supercenters a week in 2006. With 1.6 million direct employees in 2004 and close to 3 million indirect employees, Wal-Mart is the largest private employer in the United States. It has utterly outdistanced its competitors, both in the alimentary and non-alimentary sectors. As an example, in that last sector and on St Patrick's Day, Wal-Mart's turnover is higher than what its direct competitor, Target (number 2 in the United States) achieves in a year! It is five times bigger than Sears, and Kmart (discount retailer), despite both store chains recently merging in an attempt to stay in the race. Likewise, Wal-Mart has been outdoing Toys "R" Us in toy sales since 1998. Wal-Mart has become number one of the United States food industry with 16% in market shares, and is also number one on the worldwide scale, despite having entered the race only recently. As a matter of fact, it owns about 20% of all market shares in many American sectors. What is the secret of this growth?

Let us try and find the answer in some numbers. More than half the population of the United States (155 million people, or 59 million households) live less than five miles away from a Wal-Mart store; 90% (265 million people or 99 million households) live fifteen miles away; and 97% (285 million people or 107 million households), 25 miles away. The grid is therefore very tight, and not limited to areas with a high population density. In 2005, Wal-Mart published numbers indicating that 100 million Americans, and 138 million people worldwide, shopped in their stores every week.

While the network was exponentially growing, distributing costs were maintained around 3%, one or two points lower than the competition. All this thanks to the use of just-in-time production principles, which reduce storing costs to a minimum, and Wal-Mart's ability to anticipate the market's needs on the entire distribution network, in order to deliver the right product to the right place at the right time.

These two characteristics are essential, and naturally lead us to construe Wal-Mart as a system of systems, and draw a few lessons from it. First of all, Wal-Mart is composed of a network of stores, each individually managed and independent from the other stores. We find both independence criteria – managerial and operational independence – and the evolutionary criterion that are present in Maier's definition. This is lucky, because without them, it would not be possible to keep on opening new supercenters at the impressive rates mentioned above! The

geographical grid is also essential: this is what helps attract customers, organize the logistical branches and guarantee flexible prices, which will contribute to the optimization of the value chain. This is where we leave Maier's criteria to use our own definition; everything rests on the importance of Wal-Mart's organizational processes. In order to keep distribution costs at their lowest, and to offer lower prices than the competition, sales forecasts are performed for each product category, in each store. The goal is to minimize both stocks and unsold goods, and obviously to fulfill the customers' expectations by offering them what they want. Inventory management follows the just-in-time strategy, and each store carries different products, since said products vary according to the local clientele and the area's competition. Prices are adjusted, almost in real time, in regards to the local competition's prices. The packaging is also adjusted: for the same price, you can get more product, which mechanically means the price of each unit is lowered. This mechanism can sometimes be pushed to the extreme, since some products can be packaged in such quantities as to make it impossible, in a day to day consuming pattern, to eat it all, but the consumer is so subjugated by the money gain that he will purchase it anyway. At the same time, it also lowers the need for logistics and routing of the right product to the right place in a just-in-time logic, since a customer will buy fewer unitary products in one go as the unitary quantities increase!

The whole organization thus serves the efficiency of the whole business, with its systematic search for the optimization of the entire production, storage and distribution line, in order to achieve the lowest prices at every stage. Therefore, the value chain and its constant optimization, with the local adaptation of said optimization, is the determining parameter of Wal-Mart's success, a real system of systems: its organization is reticular, with multiple supercenters linked to supply centers, delivering the necessary products on demand. However, as has already been pointed out with the numbers of indirect jobs, Wal-Mart is also a network of suppliers, and suppliers of suppliers. Because of the width of the enterprise, Wal-Mart is frequently accused of shaping its suppliers. A single example can illustrate this scaling: salmon sales. Wal-Mart buys its farmed salmon from Chile, and actually buys 30% of the country's entire production! With such numbers, there is also an influence on many more aspects than the "simple" commercial value chain: the environmental impact is important, for the farms are so big that the natural waste from the salmon creates a serious pollution of sea soils (strangling the local flora and fauna). The impact on local work conditions is also important, both because of the high number of people they employ, but also because the current codes of conduct aim to keep wealthy countries from benefiting from the possible exploitation of the workforce of poorer countries, and instead bring to those workers fairer work regulations.

The entire ecosystem is therefore impacted by the organization processes and the global optimization of the creation of value. So much so that it is sometimes said

that Wal-Mart has changed the economy with its new orders of magnitude. More prosaically, studies show that the opening of a store in the United States modifies the local economic balance, leading some small shops to close, taking in their employees (so the job terminations are followed by new opportunities), and having a strong impact on the trades and the organization of the community. We're right in the middle of systems of systems, and, more than the emergence of initially unexpected behaviors, the system of systems' entire value chain, and its environment, are completely changing as they are more or less remodeled following a "Wal-Mart pattern".

To end on a more positive note, let us take another example, still involving Wal-Mart, in which the company takes advantage of its system of systems status to restructure entire value chains. In 2005 Wal-Mart decided, as part of its new strategy of sustainable development, to work with cotton producers in achieving organic farming practices. On a short-term basis this may cause the price of cotton to go up, since it takes farmers several years to obtain an organic farming certification. In order to balance those additional costs, Wal-Mart takes advantage of its influence to lower costs on the entire chain by eliminating some of the middle-men. The end product: clothes sold at the lowest prices, as is Wal-Mart's goal for all its products, and made of organic cotton. The company's influence on its ecosystem, previously mentioned, should enable mass marketing to evolve towards organic cotton. This is all the more significant as conventional cotton farming represents more than a quarter of the worldwide use of pesticides.

The first results are already appearing: in 2006 Wal-Mart bought a quantity of organic cotton equal to the entire world production from 2000 to 2005; the dynamics are launched, even if organic farming is still at a minimum, approximately 1% of the entire cotton industry.

To conclude the illustration of these concepts, let us look at another example of systems of systems where the value chain is essentially organizational: the United Nations, created in the aftermath of World War II. We might even say that a certain optimization of both the assembly and its operating modes was achieved, insofar as the United Nations learned from the failure of the League of Nations, created in the aftermath of World War I.

1.3.3. Epistemological return to the notions of emergence and openness

We have previously mentioned emergence among Maier's criteria in defining systems of systems, but how can we define this concept with more accuracy? According to Edgar Morin, "we can call emergences the qualities or properties of a system which present a character of novelty compared to the qualities or properties

of the components when isolated or combined differently in another type of system.” Emergence is therefore visible only on a global scale and cannot be separated from the system as a whole, but the paradox is that the so-called emergent properties can then be found on the level of the components – when these are grouped into a set, obviously – even though they weren’t there when those same components were independent. Emergence is therefore a quality, produced by the system’s organization, indivisible from the system in its entirety, and new compared to the former qualities of constitutive components. It cannot *a priori* be reduced logically and phenomenally, but imposes itself once the whole is set up. Therefore, *a priori*, it cannot be reduced to a superstructure which could be stuck on the global system, namely what we called the assembly in our definition of systems of systems. Moreover, emergence has an intermittent character: it cannot be observed as a component quality before the assembly, and it imposes itself phenomenally after it. All these considerations lead Edgar Morin to see in this constitution of systems of systems a double play of “formation of the whole” and “transformation of the parts”, via, on the one hand, the qualitative acquisitions and losses, and on the other hand the necessary transformation process which the constitutive elements go through by participating in the creation of the system of systems which subsumes them.

This idea of transformation is interesting, for it associates a logic of irreversibility with system of systems assembly, something which we might neglect in the great tradition of engineers and which should be taken into account to achieve a certain reversibility, either to discard a system of systems once the initial capacity requirements which had motivated its existence become obsolete, or to adapt to a new notable evolution of capability which results into new systems of systems assemblies. But the previous epistemological reflection warns us about the possible impacts on the components’ level when the latter are grouped within a system of systems.

So as to make the previous considerations more concrete, we are offering an example of emergence in a context we’ve already broached: the one of network-centric operations. When different networks are interconnected, phenomena such as distribution or percolation can be observed on the global scale. Distribution is a propagation mechanism which happens within a network, depending on the spatial dimension, and which can undergo a potentially infinite extension. Percolation is the phenomenon which can be observed, for example, in a coffee filter when a little water is poured in: the water drops go through some of the ground coffee without necessarily going through the entire filter at first, but manages it past a certain flow. To come back to the initial problem, distribution offers a means of spreading information through the entire network without necessarily implementing the global means needed to transfer this information: this is an interesting phenomenon, very energetically thrifty in terms of the organization of the information’s transmission, since the intrinsic property of the connected network makes this transmission

possible. Likewise, the percolation parameter shows the extent of the interconnected network's ability to transfer information through its spatial expansion. These two simple examples demonstrate the interest of concepts, but also the importance of these parameters which are qualified as critical parameters in dynamic system theory, for they bring about changes or sometimes disruptions of dynamics, and as tipping points in sociology, notably developed in the works of Morton Grodzins as well as the works of Nobel Prize winner Thomas Schelling. They also offer leads as to the command of these systems and the use of their emergent properties. We could envision being able to reciprocally design the interconnected systems of systems in order to exploit these intrinsic qualities to their fullest. The challenge is to eventually favor the emergence of unexpected but beneficial behaviors, while knowing how to avoid or quickly neutralize harmful behaviors.

Let us move on to the concept of openness which we have omitted until now: in thermodynamics, an open system, including material and/or energetic exchanges with the outside, is opposed to an isolated system, which does not allow such exchanges. The importance of openness emerges insofar as it defines, by necessity, an inside and an outside, and is therefore a boundary between the two, acting as a place of exchanges. The key element of this boundary is that it is also the place where the outside transforms itself, via the action of the (inner) system and the retroactive loop – which monitors outside disruptions – which it induces. To quote Edgar Morin, openness is “a notion at the same time organizational, ecological, ontological, existential”: fruit of the dissociation through observation between endosystem and exosystem, it plunges the system in an outside environment which then participates to the system's organization through the interactions it provokes and produces; it gives meaning to this outside environment, gives it a phenomenal existence through the establishment of “a transforming and reorganizing exchange with the environment.”

From the point of view of systems of systems, this concept of openness is doubly important: first, it complexifies the context of use, insofar as it updates the necessity to interact with an environment we do not command and which will constantly force the system of systems to adapt itself; secondly, it goes against the engineer's usual approach, which consists, as we will see in the following paragraphs, of trying to master (through the enclosing of its models) all the variability conditions in order to go back to deterministic situations (the issue is known beforehand if we can access the initial conditions), foreseeable (the issue can be calculated depending on the past and the launched actions), and controllable (it is possible to act so that the sought issue is obtained). A difficulty resides in this opposition between, on the one hand, the absence of an explicit boundary which stems from the fact that no component, individual, organization, whether it be inside or outside the system of systems, can see all the aspects because of the connection and the interaction loops between components, but also because of the bidirectional influences with the complex

outside environment; and on the other hand, the emergency to create a boundary around the intervention field, necessary to have any prospect of management, operation, funding, evaluation and qualification. As a result, the choice of boundary can be arbitrary and differ from one stakeholder to another. Conceptually, this is a true hard spot in system of systems engineering.

1.3.4. *Small aside: system or system of systems?*

The question is more delicate than it seems:

- a computer is a system made of subsystems (peripheral devices, central unit), themselves composed of various electronic components, each demanding independent knowledge and manufacturing chains and which, for some of them, can obviously be put to different use, and yet, we are tempted to say it is not a system of systems;
- a commercial plane like the Boeing 777 is composed of around four or five million pieces, and its development has provided work for over 10,000 people, over several continents: such a system is prominently complex, are we ready to classify it as a system of systems?
- everyone agrees that the air transportation system is a system of systems but can it be classified as a system? Opinions differ depending on people's more or less product-oriented acceptance of the concept of system.

The same question can be asked in biology: cells are capable of reproducing and surviving through transmission of their DNA information. This is the case with numerous unicellular organisms such as bacteria. With more complex organisms, such as an animal, which can naturally be seen as a system (do we not talk of the living system?) cells become differentiated during the growth, evolving into various "systems" (organs, members, etc.) which have the responsibility of keeping the whole organism alive. Each cell has its operational (it has a specific mission) and managerial independence (it finds its energetic resources around itself, lives, transmits its genetic inheritance). In medicine, man is studied as a system, via anatomy (study of structures) and physiology (study of functions), and can even appear to be an "integrated biological system" with various layered hierarchical levels (proteins, organelles, cells, tissues, organs, respiratory/cardiovascular/digestive systems, etc., body) exhibiting different properties and functions on each level.

Moreover, evolution has given rise to a new emergent behavior: life of the complete organism. A dog can therefore be seen as a system of systems. But it is also a system, as an individual inside a pack, the pack representing the system of systems. If some people find this example inadequate under the pretext that

individual cells cannot theoretically live without the organism they're part of (which is already to be debated, since you can make them multiply in test tubes, for example in order to create skin transplants), let us remember some bacteria (*Myxococcus xanthus*) which, under precise circumstances, gather into groups of bacteria with a definite shape and can be considered as true multicellular organisms, moving, eating and globally defending themselves against some aggressions. In that case, a new organism is truly created, for it will guarantee the survival of the individual bacteria, something they could not have achieved on their own.

On that subject, it is also interesting to note that, in biology, the more evolved the organism, the more each constitutive system will depend on the others, for evolution has a tendency to avoid functional redundancy, so as to achieve energetic and structural optimization. This is, at present, less obvious for artificial systems.

To try and clarify things, let us perform an exegesis of the notions of “system” and “system of systems”, building on given definitions, quoted for commodity, and which have the merit of being formulated more simply than the extensive definition given by Maier:

- a system is an integrated group of components – workforce, products, processes connected and linked, so as to satisfy one or several given objectives;
- a system of systems is an assembly of systems which can potentially be acquired and/or used independently, and whose global value chain the designer, buyer and/or user is looking to maximize, at a given time and for a set of conceivable assemblies.

The syntactic analysis of these definitions shows that a system of systems is a system: an assembly is, by definition, a reunion of objects so as to form a whole; this reunion happens through the juxtaposition of objects and the creation of links. Moreover, the maximization of the value chain's performance can be seen as a mandatory objective to achieve the aforementioned assembly. Reciprocally, here is a system; let us define the assembly of systems reduced to the singleton constituted by this system, and define the value chain as the objective satisfied by the system. The latter is therefore a system of systems, compliant with the definition.

To sum up, the formal analysis of those two definitions shows their equivalence, something which might seem confusing. However, it is important to point out that the interpretation of a system as a system of systems is completely *ad hoc*, since the demonstration is based on a spectacular contraction of the studied assemblies towards a single system. But if you look more closely, this is not surprising: in the examples that we have given, the fundamentally subjective character that decides whether a system is or is not a system of systems comes from the fact that the value chain is actually not so simple (typically not reduced to a “simple” objective) and

that we can imagine there being a plurality of possible solutions which might satisfy the adequate level of performance (therefore, typically not a single assembly). It is thus, finally, this variability and the complexity in precisely comprehending the added value characteristic of the system of systems, which yield this fundamentally subjective character, characterized in our definition by the necessary mention “in a given time and for all possible assembly”. Far from being an about face, this data which *a priori* concerns the field on which the optimization of the value chain’s performance is being performed, turns out to be the criterion which marks the distinction between the notion of system of systems and the notion of system.

1.4. Control of the system of systems

We have tried to delimit our chosen field, and have defined and illustrated a system of systems; let us now see how to comprehend it during its design.

1.4.1. System engineering

Going back to the engineering etymology, the Latin root *ingenium* highlights the necessary creativity underlying this activity: the engineer, far from the reductive portrait of a manager of a project where the options are *a priori* easily delimited, is before all else a “creator”, in that he must be proactive towards the evolutions or eventual revolutions that could have an impact on his work. Insufficient knowledge of this dimension has sometimes led to a confusion of genres between project management and system architecture engineering.

Let us therefore list the various tasks linked to system engineering, such as they have been formalized in various standardizing processes for a few decades (since the MIL-STD-499B, through the EIA/IS-632, ISO-12207, SE-CMM, or ISO 9000: 2000, to the more recent ISO/IEC 15288, last standard recorded in 2002), in fact, since the usefulness and necessity of such a standardization was recognized, following the big projects led by NASA and the American Department of Defense in the 1950s and 1960s. It is interesting to see that, historically, standardizing documents only originated in the last century, but belong to a much older tradition.

As an illustration, the formalized notion of project management goes back to the 13th and 14th century France, during St-Louis’ reign, where builders were organized into guilds by the King of France and entrusted with the monitoring of the royal works; then in the middle of the 14th century, competent administrations were created to conduct the great works of urban fortification, in order to organize the scale and variety of the works and agents (close officers of the prince, lord, qualified project managers, local craftsmen). The notions of management (the general

contractor designs a technical framework which the project manager will use all through the project, and daily pays the building teams) and of market (a tender contract is designed, and the winning bidder is paid upon signing of the final certificate) also date back to that time.

The engineering process follows a basic logic which can be adapted *ad libitum* depending on individual responsibilities and specific trades. This logic translates into a four-step loop. The first two steps form the so-called descending branch of the process, going from the general to the specific: *specification* and *design*. The last two steps, in reverse order, form the so-called ascending part of the process: *integration* and *validation*.

The important thing to see in this loop is the dynamic aspect of the two constitutive parts: by linking the various descending branches while moving towards the greatest refinement of the system's product tree (in other terms, further breaking the global vision down to the elementary components), then doing the same thing for the ascending branches in reverse order, we recreate the classic, or even emblematic, "V" shape, the variations of which (Y, W, etc.) can be easily interpreted through a modification of this relation mechanism: for the "Y", the branches linked to elementary components, which can be defined and designed independently before the final integration, are paralleled; for the "W", an intermediate level is defined in the tree diagram, from which the ascending branches are reached, and only then does the rest of the tree unroll to form the "W" specific to a first iteration of incremental development, represented by a spiral rather than a succession of "V"s.

Those various steps (specification, design, integration, validation) respectively correspond to:

- the definition of requirements and their standardization, so as to transform the need into data which can be handled on the studied level;
- the translation of these requirements into a solution on the studied level;
- the grouping of the level's "components", that is to say the products of the following refinement level;
- the validation of the previous step, the acknowledgment that the work has been well done and can be trusted, for subsequent treatments.

In fact, we should go beyond the apparent sequentiality of these four steps, hence the importance of the loop, which translates into permanent round trips (and compromises). Specification is therefore achieved through the conception of validation maps, in order to determine whether the requirement which translates part of the need is a good formalization, capable of guaranteeing this to an external participant. Conversely, validation must be carried out in comparison with the

specification's exhaustive coverage. The same mechanism is found between design and integration, that is to say between the definition of a technical solution and its assembly: this technical solution, as it becomes more precisely refined, must be assembled, hence possible constraints *a priori* on the solution itself, recorded in the integration mapping; reciprocally, we should check that the integration happens as planned.

Let us now go back to the longitudinal aspect of the process and what it entails in terms of responsibility. Each loop translates the previous loop's expression of need into requirements and solutions which can be verified on its level, some components of which will be transmitted to the following levels as a new expression of need, pending the components to integrate. We can see the emergence of a certain transfer of responsibility as we go further along the product tree, particularly in terms of risk; hence, the fundamental character of validation, which helps regain this responsibility and transfer it in turn.

These steps must be accomplished knowingly, with the utmost homogenous methods and tools, as much to facilitate rapid roundtrips in case a level features a defect, than to accelerate and parallel, when possible, certain tasks (an approach often called concurrent engineering, but which is slightly different on the conceptual level in the case of spiraling incremental development).

This aspect will reveal all its importance in later sections, when we will study simulation in engineering and the acquisition of systems of systems. The issue will be all the more noteworthy since the system's entire life cycles will have to be taken into account, cycles which will span over long periods, sometimes outlasting the contractual project management organizations. To the deployment in space of the product tree solutions, is added the deployment in time of all these components.

1.4.2. *Is there a need for systems of systems engineering?*

If we analyze the standardizing definitions of systems engineering, which present it as an interdisciplinary approach that enables the transformation of a need into a system solution, and helps adapt, make evolve and verify the system solution during its entire life cycle, to achieve client satisfaction (IEEE 1220-1994), nothing *a priori* keeps this concept from applying to a system of systems. However, a certain number of facts differentiate, *a priori*, the control of a system of systems from the control of a system.

1.4.2.1. *Which general contracting?*

First of all, because of the very desire of making a system of systems “emerge” as an immediate solution to a capability need without launching a new stage of

development, instead building on the intelligent pooling of existing systems, even if it means superficially modifying certain components (but favoring the use of the organizations and processes implemented during these systems' use), there doesn't *a priori* exist a person or a focal team in charge of system of systems engineering who could act on the whole set of implemented requirements, products, and services. The situation is therefore radically different from what we know in system engineering, in particular since the large space programs that were launched after World War II.

In terms of requirement management, the challenge is then to precisely define those requirements on the level of system of systems, and their coupling (in terms of redundancy, coherence, and eventual partial contradictions) with the requirements on the level of the constitutive systems. Indeed, among the many questions that may arise, can be found:

- the relationship between the various concepts of operational use of the system of systems and its components;
- the availability of the constitutive systems' requirements, documented and validated;
- the relationship between the system of systems' functional architectures and its components, and therefore the eventual distribution of certain global requirements on the level of several components, which is then translated into constraints on these levels, either functional or expressed as interfaces to define and manage;
- the requirements and constraints related to regulations, for the system of systems' deployment, and the coherence with the components' issue;
- the adequacy between the capability increments and therefore the pro forma evolution of the system of systems and the individual evolutions of its constitutive systems;
- the links between the stakeholders and the prime manufacturers (everyone knows the rivalry between user communities, as well as between the entities in charge of directing projects, who are naturally competing to gather the budgets they need).

All these questions bring into sharp focus the issue of explaining and verifying system of systems' requirements. The difficulty arises from the fact that, most often for now, no general contracting entity "owns" the systems of systems' requirements, insofar as it is not a project that can *a priori* be easily confined between boundaries, and because it builds on a preexisting set of systems, and therefore requirements. Moreover, some system of systems' requirements cannot be allocated to a constitutive system, hence the difficulty in having them taken into account by the project teams, as much on the level of their expression as their future validation.

Within organizations, the teams in charge of a system of systems generally do not have previous experience in terms of responsibility and authority over a constitutive system's project leader, there is no explicit description of the system of systems in terms of interfaces with a focus on optimization which would then reverberate onto system requirements. The same difficulty is found with feedback processing: the use of systems in an operational context most often brings about evolution requirements, as much on the level of procedures for use (called doctrines in a military context) as on the level of the system's technical architecture. The same thing is evidently valid for the system of systems; certainly with an increased weight on the procedures of use and the organization both in terms of use and system interfaces (we can easily imagine there is a wider margin of progress in terms of astute use of components than with the technological evolution of said components). But there again, the general absence of a global responsibility explicitly dedicated to an entity means that those considerations are treated without a general overview, and more often the operating system of systems' general alignment happens on the less efficient system.

1.4.2.2. Life cycle and development process of a system of systems

Upstream, the difficulty lies in controlling the link between the operational capability increments, the evolution of the technological capabilities, the individual development of the constitutive systems and the management of their evolution. To achieve this, interaction loops (proper loops, not only descending or ascending flows separated by large constants of time) must be set in place between, on the one hand, the capability analysis and on the other, for constitutive systems, the definition and analysis of the system requirements, the functional analysis and preliminary design of architectures. It is via these narrow and continuous interactions, where the process of system of systems engineering follows a true logic of spiral development similar to what was popularized in 1988 by Barry Boehm in software engineering (see Microsoft's development of the Windows operating system these past fifteen years), that bringing the logic of capability increments development to life becomes possible. Let us look at it in detail, for there can be an apparent confusion between incremental development and continuous spiral development.

A process of incremental development aims to deliver the system as successive functional blocks which will incrementally satisfy the need. To define an analogy in the field of navigation, the process goes through the definition of intermediate steps to go from point A to point B, each successive step bringing us closer to the final destination. On the other hand, a process of spiral development requires the flexibility of one of the variables, up to now considered as fixed: the functional perimeter. Rather than fixing it from the start, it is defined progressively, according to constraints and the technological and operational opportunities that arise. This desire for flexibility therefore limits the preliminary knowledge of what will be

delivered *in fine*. Such a level of freedom may seem shocking within certain current thinking contexts, where the transfer of risks to the project manager is systematic – and therefore leads to the rethinking of the relationships between project manager and general contractor, as we will study in an ulterior section – but makes perfect sense when you go back to the navigation analogy: every great sailor, even if he knows his final destination, sets his way as he goes along, according to the weather conditions and the place of his competitors. Among the plus of the approach, let us mention: its pragmatic nature, since innovations are experimented with and validated one by one, then progressively mastered through successive analysis cycles before launching into the final development; the possibility to analyze multiple options in parallel.

With systems of systems, both these processes must be combined: an incremental development is first run on the level of capabilities, and a spiral development is then performed for each increment. The logic is therefore to develop capability increments, which *a priori* means talking about the corresponding capability's life cycle is more straightforward than talking about the system of systems' life cycle. To follow a somewhat cliché image, capability could be seen as a river flowing from its source (the genesis of the capability concept) to its mouth (the disposal process), and the different systems of systems as segments of this river, sometimes dead branches: it happens in particular with software-intensive systems of systems, when they are not, like information systems, pulled out to leave room for the next increment.

The main advantages of the aforementioned development process of capability increments are of two kinds: first, the approach's flexibility, which helps best manage the implementation of components which were not known at first, whether they are new technologies, or even budgetary cuts insofar as the functionalities can be transferred to an ulterior increment; then, the early implication of all stakeholders, which heightens the adequacy between supplies and expectations; this also gives greater room for creativity.

On the other hand, on the level of requirement engineering, control of the transition from one increment to the other, as well as the set of decisions taken or reported on the level of an increment, requires traceability.

If we look deeper at the part of the process which is dedicated to configuration management – the system's configuration sets what belongs to the system and what is outside of it but may interact with it – we are once again faced with a problem concerning requirement engineering. Indeed, the various constitutive systems of the system of systems evolve through time, either because of technological maturity, or because of associated capability requirements. But this means that configuration management and requirement traceability are both required on the level of the

system of systems, because of the evolution of the system's capability concept, and on the level of each constitutive system! This double difficulty, linked to the configuration management of the system of systems, rising from the evolution of the need associated with the set but also with all the needs that motivate the individual components, has to be taken into account. Actually, is it only an added difficulty in configuration management, but in practice this could be an unbreachable obstacle if the process is not sufficiently tooled and documented.

1.4.2.3. *Process of systems of systems engineering versus process of system engineering*

According to engineering standards (ISO/IEC 15288 *Systems Engineering – Systems life cycle processes*), in which the process of system engineering is broken down into processes of agreement (acquisition, supply), organizational project-enabling (strategy, resource management, quality management), project management, and technical management of the system's life cycle, we have *a priori* all the tools necessary for systems of systems engineering, as long as all aspects are properly taken into account. As a matter of fact, work groups are currently proposing extensions of the standard ISO/IEC 15288 in order to adapt it to systems of systems. Of course, the terms used for the various stages within the standard are not necessarily the same, but we can still unroll the process: for example, the stage which concerns the system's definition of need must be clearly broken down into several stages of definition relative to the needs in capability, the use cases and scenarios, the business strategies, in order to define which systems to reuse, which standards to enforce. This is why the process might look like a "W" instead of a "V", so as to bring out this stage of design and validation of the initial capability need. We could also use the symbol "@" to underline the spiraling character, with the initial circle representing the stage dedicated to the initial capability need.

However, we cannot deny that the direct application of the process is often difficult insofar as there is no project team for the system of systems, unlike what has been done for systems for decades. In fact, careful attention should be paid to the aspects of organization and transfer of information, but also responsibility, between the teams working on the various systems' projects which constitute the system of systems. This transfer of responsibility must go both ways, so that budgetary, schedule, or performance impacts on the level of a component are not crippling on the level of the system of systems, and *vice versa*. Moreover, the strategy inherent to business processes also takes on an added importance: the concept of system of systems, corresponding to an acquisition strategy to achieve a specific effect, pertains to this strategic level. Even though the results of the strategic process are most often considered as the entry data of a system's life cycle, they are the first link of the systems of systems' requirement engineering process: they complete all the actions which concern the desired goals and demands, they provide

a global and permanent vision on the global system and the environments it will interact with, they act as a guide for the reduction of uncertainty and offer permanent adjustments of the environments.

Fundamentally, the technical engineering process is the same, as well as the various underlying processes, but it is the general organization of the different processes which poses difficulties. For there is a conflict between the descending approach, determined through a general capability policy, and the ascending approach centered on the software. Moreover, systems engineering implies the breakdown of the system into components, whereas systems of systems engineering must be focused on the composition. Even if the technical processes are *a priori* the same, the heightened complexity demands them to be completely unraveled, with no exception, paying special care to precision. Methods, tools and processes – we could add the experience of failure, for it is in fact the best way to realize that a stage that was validated too quickly under a false sense of control of the context, is in fact a crippling obstacle to the global performance’s guarantee and the system of systems’ durability – are therefore the keys to success in system of systems engineering.

1.4.3. Architecture: a key element

To quote sociologist Edgar Morin, the organization or architecture is what lends stability and structure to the interrelations between components:

“the system is the phenomenal and global character taken on by interrelations whose layout constitutes the system’s organization. [...] The organization is the internalized face of the system (interrelations, articulations, structure), the system is the externalized face of the organization (shape, globality, emergence)”.

In the authoritative accounts of systems engineering, the concept of architecture stems from the same idea. The standard MIL-STD-498, on which the various standards for system engineering in the last few decades were built, offered:

“architecture is the organizational structure of a system, identifying its components, their interfaces, and a concept of execution among them.”

The workgroup IEEE, while working on the standard P1471 (recommended practices for the architectural description) from 1995 to 1996, offered: “architecture is the highest-level concept of a system in its environment.” From this definition, three things should be remembered. First, architecture is a concept more than a structure, which would imply a connotation of physical structure. The generic aspect

is dominant. Secondly, the architecture is deliberately placed on the level of abstraction, hence the notion of highest level, with the downside that it suggests a notion of hierarchy, notion which should be avoided at all cost. Finally, the architecture is not a property of the single system; taking the system's environment into account is essential. The definition finally retained by this standard, and then called ANSI/IEEE Std 1471-2000, is:

“the architecture is the fundamental organization of a system, embodied in its components, their relationships with each other and with the environment, and the principles guiding its design and evolution.”

INCOSE's (International Council on Systems Engineering) SAWG workgroup (Systems Architecture Working Group) on the architectures of systems has adopted the following definition:

“the architecture of a system is the fundamental and unifying structure defined in terms of the system's components, interfaces, processes, constraints, and behaviors.”

The American Department of Defense's document on the architectures of control, command, communication and intelligence systems (U.S. DoD C4ISR Architecture Framework) says:

“architecture is the structure of a system's components, their interrelationships, the principles and guides which govern their design and evolution through time.”

Finally, the Technical Open Group for Architecture Framework (TOGAF) defines architecture as: on the one hand, the formal description of a system or the detailed map of a system's components so as to guide its development; and on the other hand, the components' structure, their interrelationships, and the principles and guides which govern their conception and evolution through time. It should be pointed out that in this same authoritative account architecture is also seen as a means to plan the acquisition of components and the investments necessary to reach the operational objectives. This is consistent with the vision defended by the *Open Group* which the TOGAF belongs to, an international consortium defining some standards of computer engineering and offering authentication services and tests of standard conformity. It defines business architecture as covering the four following aspects:

- strategy, governance, organization, key-process;
- organization structure of both hardware and software assets, and information management resources;

- plan for the deployment of individual application systems, their interactions and their relations with the organization's core business processes;
- technical infrastructure which will support the applications critical to the global mission.

System of systems architecture is therefore much closer to the notion of business architecture than to simple technical architecture in the way it could be envisioned for an individual system, which is a common mistake, for even software engineering standards such as the IEEE Std. 610.12-1990 define the architecture (of a software system) as: "the organizational structure of a system." The aspect related to the enterprise's key-processes takes on a specific importance, and to come back to the previous examples of a system of systems, we can see the coherence with notions such as the transformation process in the field of defense: the architecture of the system of systems becomes the first link, going from a capability rationale to the effective ways of implementing it, taking its various aspects into account, and not only the technical aspects.

If we organize the software, systems, business and even systems of systems architectures, we could oversimplify things by saying they answer the following issues:

- for the software, the key notions are the functions (the "how?") and the data (the "what?");
- for the system, the key notions are the interconnection network (the "where?") and the temporal dynamics (the "when?");
- for the system of systems, the key notions are the business vision, the organization and when all is said and done the human resources (the "who?"), as well as the business strategy and therefore the motivation (the "why?").

Beyond these general definitions of the notion of architecture, we should see the various standards that allow for a precise explanation of the concept of "structure", one of the terms present in all these definitions, in order to see the operational side of the concept and to be able to study its role within the management process of systems of systems. Notwithstanding the more or less different terminologies, we classically distinguish, in the engineering standards, both software and system (see system engineering standard IEEE 1220):

- *organic or logical architecture*: description as an architecture model which specifies the characteristics of the components and their interfaces in order to translate the functional analysis and satisfy the requirements;
- *functional architecture*: description as an arrangement of functions, their subfunctions and their interfaces, which defines the execution sequence, the data and

control flows that condition them and the performances that are needed to answer the requirements;

- *the physical architecture*: description as a set of physical organs and their interactions, which constitutes the solution which translates the functional architecture and satisfies the requirements.

In fact, the three main views of a system are represented here, the benefits of their use based precisely in their complementarity: the logical view is in direct relation with the requirements, the functional view is linked to the functional breakdown and those functions' sequencing, the physical view reasons in terms of technical solutions. It is immediately obvious that those views apply to systems of systems; the logical view focuses on the elicitation of the capability vision, the functional view also illustrates how the functional distribution can happen between systems, then within constitutive systems, and the physical view focuses on the technical vision of the components as much as the physical interfaces between them.

One of the major challenges is to manipulate these various views and exploit the crossing allowances, for example between requirements and functions, or between functions and the physical components which create them: in this way, the obsolescence can be controlled (going from the obsolete technology to the function, or even to the specific requirement), and impact analysis can be performed, leading to true system configuration management.

The architecture's analysis and design stages are milestones in systems of systems engineering. Notwithstanding the high combinatorics, the architectural work between a system and a system of systems would be the same if we could go back to the start. However, the major difficulty stems from the theoretical existence of inherited systems and their integration within the system of systems. Those inherited systems must be abstracted, and their architecture studied from various angles; if these architectures are incomplete or unavailable, a re-engineering process must be launched, so as to be able to consider the architectures in their whole, not necessarily as a single object but rather as a federation of objects. This approach's interest is not only intellectual, but stems from the numerous experiences in engineering (whether it be system or software engineering), where it has been demonstrated that the management of risks and corrections can only be acceptably controlled in terms of delays and costs if sufficient efforts are dedicated to the design and in particular architectural stages. Any skimming on this level later translates into drastic cost increases.

To guarantee a system's durability and ability to evolve (something which is *a priori* mandatory in an incremental capability logic and a tight budgetary context), it is therefore preferable to launch a process of architectural re-engineering of certain components rather than immediately attempt the integration. This is another

advantage of architecture, which provides a static and dynamic description of the life of the system of systems, and is a growth model on which to organize future evolutions through successive iterations.

In the end, the architecture's importance lays in its description of the system of systems without all the design details, in its identification of critical interfaces, and in its comprehension of the allocations between functions and components. It determines the product tree, that is to say the components that will be handled by independent teams in charge of the design. It allows for a more precise division and thus helps the design teams minimize their interactions via a play of interfaces which take on the role of foundation for the whole system and must have a high stability. From there, we can deduce the cooperative interactions between teams, the technical and organizational interfaces, the key-stages in risk management, the schedule of the necessary reviews and inspections on the level of the project management process. The architectures thus become a meeting place for the design teams of the system of systems and their constitutive systems. Through the control of the architecture, we achieve control over the entire assembly, as long as the flows and interfaces are also controlled, and obviously, as long as traceability is established between all these pieces of information, conditions which we are about to develop.

1.4.4. *Control of the interfaces*

We have seen how architecture design can help determine the product tree, and therefore the distribution of design tasks between teams. Because of this, it is fundamental for the study of the impact the addition or modification of one component or more, upstream of a requirement, might have on the integration. Indeed, it helps determine the number of unitary tests as well as integration tests which should be performed again in order to guarantee the non-regression. But this requires knowledge of the various exchange channels, hence the criticality of interface control.

Let us use an example of systems of systems to illustrate the various questions that might arise on the level of the interfaces. This particular system of systems contributes to the military capability called projection: it is composed of a naval military force, with an aircraft carrier and its aircrafts, a set of surface vessels protecting it from threats such as submarines, missiles, aircrafts, other surface vessels, mines, etc.; but the naval force itself interacts with other components, either airborne, submarine, or even terrestrial when it is acting as a back-up of ground forces in the context of operations being run on the coast or up to several miles inland.

This may give rise to a certain number of questions concerning the design of such a system of systems:

- how should the interfaces between the aircrafts and their carrier be managed: this technically translates into requirements about the weight of the aircraft, the size of the deck, the forces on deck, the catapults and arresting cables, the nature of the deck's skin, the plane's characteristics, such as its landing gear;
- how can the interface problems be identified upstream, and management structures implemented: for example in terms of interoperability, this poses the problem of tactical data links, the presentation of common situations (called a COP, *common operational picture*) while taking into account the diversity of sensors and carriers as well as the coherence with other systems of systems; this also poses the problem of moving from the older presentations (partial implementation of the interoperability standards NATO) to newer presentations (two by two interoperability tests, platforms interconnections);
- how can new systems, such as unmanned vehicles (aerial, surface, submarine), be integrated within the naval aviation group: launching and salvage problems, management of the flight deck, management of the transmissions, safety in the airspace (anti-collision);
- how to evaluate and identify: which tests to run and how to run them on the level of the system of systems (with the added constraint of an eventual impossibility in putting together the systems, since some systems might be in the prospective stage);
- how can the various standards be managed: this concerns the evolution of the aircraft carriers, for example, as well as the aircraft they carry, hence eventual impacts on the flight deck's definition variations, but also on the pilots' qualification; this also concerns, depending on the thrust mode's evolution, the dimensions of the support function (fuel and ammunition) and the maintenance system (should there be two different systems if the thrust modes differ?);
- how can a program belonging to different systems of systems be managed: the possible increase of the number of equipment, which has a beneficial effect on the development and production costs through a serial effect, must not be counterbalanced by increased interoperability costs; moreover, the diversity of missions *a priori* asks for more polyvalent systems, which is far from having a neutral effect on the global design;
- how can the future maintenance of operational capabilities be integrated in the initial choices: this means integrating the global ownership cost on the level of the system of systems, examining the evolution of the maintenance policies with regard to the operational capabilities of the whole naval force, managing the system of systems' performances in time with regard to the availability of its components.

1.4.5. *Traceability: a mandatory component*

Traceability is a necessary condition for controlling complexity. It must be exercised at every stage of the life cycle. During the acquisition of the system of systems, it is essential to record all decisions made about the use concepts and the eventual architectural choices. Indeed, this conditions the ulterior evolution capability insofar as new contexts of use may call previous choices into question, and profit from previous reasonings. Moreover, if traceability is maintained between concepts, requirements, architectural decisions, and later detailed technical specifications, then impact studies can be run, which help control the propagation of a change on one level to the others: for example, a new threat may lead to new capability concepts, hence new requirements, hence new architectures to answer those requirements. Likewise, a specific technical evolution, for example the unavailability of a component or a critical system because of the shutdown of a production chain (recent history is filled with such examples in the world of electronics and informatics, but it also happens with mechanical spare pieces), can lead to the evolution of the capability answer, and evaluating such impacts *before* dysfunctions arise is fundamental.

As we will see in Chapter 4, traceability is not only a matter of recording the facts; the latter have to be structured in order to be exploited within an impact study. Indeed, considering the combinatory complexity of systems of systems, this need for traceability is not neutral, and must be tooled to fully profit from it.

The same need for traceability is also present during the system of systems' implementation. Indeed, the networks' omnipresence heightens the imbrication of the agents, and their scattering. To control the flows, we must control their traceability: it helps to have a clear knowledge of what is happening, and check whether the required goods or service have gone through the correct steps, and it also helps anticipate possible upcoming flows, depending on the activity of such and such a user kernel of the system of systems, so as to prevent ulterior blocking or local non-satisfaction which might trigger a global failure through nonlinear spreading effects.

1.5. Tools for the control of the system of systems

Traditionally in engineering sciences, and in particular in systems theory, the control of a system is reached through several stages. The first analysis stage, also called *identification*, consists of:

- defining the system's boundaries;

- identifying the important components and the types of interaction they have with one another;
- determining the links which integrate them into an organized whole;
- classifying and prioritizing the components and the types of links, the positive and negative feedback loops, the delays.

Following this identification, the second stage, *modeling*, is launched: a representation (an organized and standardized set of opinions, beliefs and information referring to the system) is created, the *model*, from the analysis data. Modeling can be logical-mathematical, digital, analytical, etc. The studied system is then replaced by another system with a comparable structure and operation, but of a different nature, and easier to study. It is clear that any model is an approximation whose validity depends on certain requirements. As highlighted by J.L. Lemoigne: “modeling is deciding.”

The third stage, *simulation*, is the study of a system’s behavior from the model(s) used to represent it; modeling is then pushed to its final consequences. Simulation can be digital (run on a computer) or analog, featuring human- and/or equipment-in-the-loop. By definition, simulation does not only concern the model, but the interactive user-model set.

Evidently, the interest of these various stages only lies in the last stage, which represents the true control of the system, that is to say its *regulation*: this includes all the adjustment mechanisms constantly invented and implemented by the system so as to maintain its internal balance, and at the same time adapt to the evolution of its environment.

On this level, we should recall the law of requisite variety², introduced by the cybernetician Ross Ashby, who affirmed that the regulation of a system is only efficient if it leans on a control system as complex as the system that is controlled. It is important to keep this in mind, for it proves that, to achieve total control over a system of systems, regulating it will not be easy and requires the identification of the various degrees of freedom which might be played on in order to modify the global behavior: which interactions? Between which components? Are the loops stabilizing or destabilizing? What is their qualitative nature? Which types of action can we identify?

2. A system’s variety lies in the logarithm of the number of state configurations; it can be equated to a certain measure of information about the system. In reality, this law is close to the second principle of thermodynamics in physics, to the principle of internal model introduced by Wonham in control engineering, as well as to some of Shannon’s theorems in information theory.

A priori, it is therefore vain to hope to control a system of systems with an overly simple regulation, based for example on unrefined models. Hence the interest of owning models and simulations to help this regulation.

1.5.1. *Simulation*

First of all, let us define the terminology. The notion of a model is clear enough to everyone: it is an approximation, a representation or an idealization of the structure, the behavior or other characteristics of reality, whether it be a physical phenomenon, a system or a process (see IEEE 610.12-1990).

On the other hand, the notion of simulation suffers from multiple meanings: it covers the activities of model *realization* as well as the ones of model *implementation*, towards a given purpose. It appears that simulation helps reproduce the characteristics of the environment, the systems and some behaviors. Besides this descriptive angle, it helps control the conditions and situations, and therefore test solutions. Obviously, this happens with a flexibility, a security and a cost which real experimentation cannot offer (as a reminder, 30 years ago, a company like MBDA needed hundreds of actual launches to validate missiles, against half a dozen or less today). Simulation is therefore greatly helpful on the level of the equipments, and in parallel, on the level of the doctrines which regulate the use and implementation of forces, as well as their training.

Simulation is the key to the success of the engineering process; it is the pivotal work tool of an integrated multidisciplinary team at the service of the integration of complex systems, bringing together the various stakeholders: staffs, acquisition offices, industry. The various agents (participating with the definition, evaluation, manufacturing, support, etc.) must share the information and data resulting from these tests and these simulations, and must also identify the necessary information in terms of tests and simulation. As a matter of fact, that is how simulation based acquisition was initially defined by the United States Department of Defense: a true acquisition process for general contracting and project management, with a sturdy and collaborative use of simulation technologies, used in a coherent, integrated way all through the program's acquisition stages.

Simulation-based acquisition consists of using and reusing the available simulation tools and technologies on the system's entire functional breakdown, the program's stages, and the various programs (in particular for systems of systems). It goes hand in hand with a better management of modeling and simulation resources during the acquisition: from a simple punctual support – in time and space – to the program's engineering, we move on to a coherent and integrated process. Potentially, it helps reduce cycle times, the resources and risks that come with the

acquisition process of weapons systems, while increasing the systems' quality and reducing the global cost of ownership. However, the expected benefits can only be achieved if simulation is run through adapted processes, typically inspired of concurrent engineering. These processes include all activities, from the product's design to its production and maintenance, leaning on multidisciplinary teams so as to simultaneously optimize the product, its manufacturing and its support, with cost and performance control objectives.

Simulation is therefore put into perspective: the acquisition of systems as parts of systems of systems. Such an approach goes along with a revolution in military affairs, insofar as the acquisition procedures are now focused on the acquisition of individual weapons systems. Via simulation, certain facets of a system of systems may be analyzed on a much shorter time scale than if the real system of systems had had to be developed and experienced with. Thanks to simulation, certain architectural or technological choices may be pushed until much further on in a program's life cycle (the virtual aspect is then largely exploited). Indeed, before any major expense is triggered, simulation is critical for the exploration of concepts and their evaluation, architectural development, specification, detailed design of the system and its manufacturing process, risk analysis, support, cost analysis all through the life cycle, disposal. But the same approach can and must be applied to the system of systems itself, becoming part of an architectural data exchange strategy.

Let us detail the potential benefits which can be garnered from simulation all through the life cycle:

- contribution to requirement management via:
 - the evaluation of the global architecture concepts,
 - the analysis of the compromises made between operational capabilities, performance, cost,
 - the choice of an optimal system architecture;
- contribution to specifications management via:
 - proof of technical feasibility before actual realization,
 - the decision on the best organization for development,
 - the wording of verifiable specifications;
- contribution to realization management via:
 - the exploration of the various production options, so as to optimize the choice of a solution while respecting time and cost requirements;

– contribution to the management of evolutions and the integration within a higher-level system via:

- a coherence guaranteed all through the system's life cycle,
- a guaranteed reuse of all or part of the components and subsystems within other systems,
- the ability to help manage the obsolescence and take into account changes in technologies or requirements, via impact studies.

This entire approach comes to life through the iterative development of virtual prototypes, immersed within realistic synthetic environments, which help, on the one hand, develop a shared vision of the imagined system, and on the other hand provide the appropriate means to achieve a better comprehension of the complex interactions between the system's configuration elements. The designed prototypes will be more easily developed and evaluated as design, development and test engineers work together: hence, a lowered global acquisition cost.

This profit may be quantified, insofar as, through the use of simulation during the system's acquisition, the expenses which arise from the decision taken (we are alluding to the 80-20 of the Pareto chart that is used in project management: past 20% of effective expenditure, the decisions taken are *a priori* concerning 80% of the global budget). Indeed, simulation enables the simultaneous management of a large set of technical alternatives, for which more or less important portions of the life cycle might be virtually unrolled, and the impact of an upstream decision may thus be measured downstream, in terms of performance and/or cost. In that way, simulation essentially contributes to the management of the project's risk portfolio.

The other factor of economical profit lies in the reuse of the simulation of some components of the system of systems; said reuse must not be limited to simple software bricks, but must concern the requirements, the architectures, the design patterns, the interface models, the test plans, the data, the documentation, etc. During the last few years, numerous cost studies have helped us understand that factor.

We can actually demonstrate that simulation based acquisition is, when properly led, profitable right from the first incident. Moreover, reuse between projects, also properly led, helps achieve savings of the same magnitude than the total cost of the design of the system which includes the individual systems. This first technical-economical analysis demonstrate the investment's interest, and the first numbers garnered on current affairs confirm this theoretical analysis.

1.5.2. *Towards integrated infrastructures: the battle-labs*

In the light of the previous sections, the key-data concerns, on the one hand, requirements and specifications, and on the other hand, the integration tests and the validation methods and data. These pieces of information must be exhaustive and completely traceable, both regarding their origin and their configuration. It is via these two aspects, both of traceability and configuration management, that we can have a vision of the entire system all through its life. Sharing this vision is equal to getting access to the various architectural and technological choices, hence the control of system engineering. In a more mathematical reading, we might say we are faced with a canonical representation whose deployment corresponds to the system's possible updates with, at all times, the ability to reconstruct said system's current state.

Beyond these pieces of information, which provide a set of discrete elements which help reconstruct the system, we can profit from having a more continuous vision of the development of each level and architectural element. Hence the interest of having a toolled process, which can give immediate access to a global and behavioral vision of the system, obviously only if a vertical integration (to caricature: from metasystem to component and *vice versa*), and a horizontal integration (ideally: through the entire life cycle, from the genesis of the idea of a system to said system's disposal) can be achieved.

In fact, the question here is to define "the" information system corresponding to the complex system that is studied, which has the vocation of being a reference and a memory, and can be connected to the other tools that might be used, such as those linked to project management, financial management, or even to generic systems of logistical support...

Various types of methods and tools exist to answer these questions: collaborative work, role-playing or table games, technical, technical-operational and operational simulation, simulators (real-time simulation with a man within the loop), laboratory or field testing, global cost calculation methods and tools, engineering, knowledge management methods and tools, etc. These methods, these tools, and the skill profiles of their operators turn out to be complementary, each type adapted to the treatment of such or such question, or coming into action at a different stage of a question's processing.

To comprehend the level system of systems, we need a global approach so as to progressively and conjointly refine the definition of the operational need, the validation of concepts, the demonstration of capabilities and validation of the chosen solutions. All these questions must be looked at in a coherent, flexible and reactive way, for the answers to some questions are necessary to the processing of others,

which requires the engineers in charge of acquisition, and the operational users, to have a much more integrated operating mode.

Incarnating this general approach, an *engineering infrastructure* may be built on the coherent use of:

- a tool of requirement formalization, with traceability and configuration management functions, as well as the connection to simulation capabilities, so as to explore and justify specification decisions;
- a tool to assist integration tests and validation, acting as a direct link with the requirements as manipulated by the previous tool;
- a collaborative working environment, which might be geographically distributed, and will give access to the various parts which operate before and after each level, and to the data relative to that level, so as to facilitate round-trips and potentially accelerate cascades between levels; beyond this function, which secures exchanges and manages the information according to its level of confidentiality and each user's access rights, such an environment becomes the key dialogue structure during the system's life.

The concepts previously established for systems of systems and their engineering are naturally found on the level of their models. The chronological or geographical spreading of their components also applies to simulations and simulators. Since their acquisition cost and their complexity do not permit them to be reconstructed with each evolution of the system of systems, their coherence (vertical and horizontal) must be worked on, so as to achieve a system of systems simulation capability.

To perfect this harmonization, a *simulation infrastructure* must be created and managed in configuration. Through a technical architecture, notably based on international standards, it offers the proper level of interoperability between models, whether they already exist or are yet to be designed, and provides the necessary services for the design of global simulations, often geographically distributed: shared technical and methodological reference documents, model and tool libraries, model configuration management, validation process support, etc. It also provides access to the so-called model engineering methods and tools, whose objective is to provide conceptual and independent frameworks for specific implementations, to design models and, more generally, simulations, depending on the expressed needs, since the generation of code relative to a specific structure happens almost automatically.

Faithful to the previously defended iterative process, the simulation infrastructure (which in itself is a system of systems) alternatively takes on the role of catalyst and focus, both of the technical coherence of the simulation means which have been acquired through the various projects, and of the harmonization of

processes within the new context of complex system engineering. It helps, in situations which are complex and, by nature, evolutionary, implement harmonization and optimization loops between the various levels: politico-strategic, tactical-operational, and technical. It also helps achieve a strong imbrication of these various levels, which guarantees a better definition and their permanent adaptation to evolutions.

In the United States, as part of the defense transformation process, the CD&E method (concept development and experimentation), which is led by the JFCOM (United States Joint Forces Command), has the exact objective of shortening the loop “technological watch and feedback – concept development – experimentation – integration within forces”. The CD&E process is supported by methods and tools such as the battle-labs, which use new virtual enhanced reality technologies, the interconnection of technical-operational and operational simulations, and sometimes, in the case of real maneuvers, real hardware. Their main mission consists of providing the armed forces with technological and conceptual innovations, by locating, testing and evaluating the most recent progress in their field of practice: the users analyze new operational concepts and study the adequacy of systems of systems to emergent needs; the acquisition community establishes which systems may contribute to a potentially interesting system of systems, and defines the evolution trajectories for the existing systems; the developing teams test the efficiency of new systems within new environments.

In France and in Europe, the tools and agents, both government and industrial, are scattered: the current organization is not optimum, even if important efforts are made to connect the existing tools and skills, and make them able to interoperate. The concept of “technical-operational laboratory” (LTO in French, for “*laboratoire technico-opérationnel*”) was introduced to help federalize the agents and tools of a given capability field, so as to create a truly innovative tool, supple and reactive, of concept development and experimentation. If the technological demonstrators, no matter how imposing (such as unmanned systems), have for vocation to validate elementary technologies, a technical-operational laboratory must guarantee the extension of this demonstration to the systems of systems’ successive levels. Four main functions can be outlined:

- Acquisition engineering: provides the necessary methods and tools for multiproject conduct and complex system engineering (allocation of the various systems’ performances, specification of interfaces and their interactions with other systems of systems, etc.).
- Promotion and cooperation: by immersing the end-users into situations which are technically and operationally credible, they can evaluate the interest of certain system concepts or ideas; moreover, this undeniably represents a plus to initiate cooperating programs, for concepts can be illustrated which partially implement

foreign systems (during the latest Paris Air Show in Le Bourget, several industrialists have thus implemented systems of systems simulation which integrated the operational information systems of various countries).

- Analysis of the global costs via systems and interfaces cost models.
- Analysis, design, or even training tools: a classical function, which concerns all the activities happening all through the programs' life cycles (including deployment preparation, integrated logistic support, dismantling, etc.).

Via the common implementation in mixed teams, it is therefore the realization of a new type of relation between acquisition services, operational users and industrial project managers, each bringing a specific skill and responsibility, to develop and experiment, in an iterative, reactive and much more integrated way, complex systems which must be defined in both operational and technical terms.

A technical-operational laboratory will help follow one or several systems all through the acquisition programs. Thanks to this laboratory, recurrent information will be available, in terms of concepts, system ideas, functional architecture specifications, technical interfaces, and later technical architectures, prototype evaluations (in particular virtual, such as in the civilian aerospace business, or the automotive industry), training concepts, use doctrines, and finally evolution offers (for example for existing platforms, so as to improve their integration within the system of systems to which they contribute). Potentially, part of these cycles can be conducted in phase lead, which immediately leads to a heightened control of risks by anticipating virtually parts of the systems' life, then to a reduction of delays when it comes to deploying systems in response to sudden evolutions of context and threats.

All these tools, integrated within shared infrastructures, are becoming more and more widespread to control the complexity of systems and project teams, which are by necessity multidisciplinary and multisite. They embody new working practices, within so-called cooperative environments, uniting general contracting and project management. They are met in the civilian and military aerospace business, but also in the automotive field, or even in the design of multimedia product. Beyond what some may consider a trend, this actually reflects a coevolution of technology, engineering processes, and the means which tool them:

- The 1980s were marked by a type of system engineering which favored tree structures (breakdown following the functional architecture, and launching of the various individual tasks) and sequentiality (synchronization of individual tasks through successive program reviews). The essential medium was paper, associated with a heavy, static management of files. We can see in this the methodological heritage from the big programs that were launched by NASA during the previous decades.

– The 1990s saw the development of integrated engineering approaches, with the increasing integration of “business” skills through the various stages of the system’s life cycle. Such a change was driven by the technological acceleration, in particular on the level of electronic components, which led to the simultaneous presence of very varied cycle lengths within a single system. The essential tools were the workflow, which still retains a strong implicit sequential character, and technical data management, which replaced paper to work digitally, but without reshaping the underlying processes.

– With the boom of new information and communication technologies, this sequential character quickly faded in the late 1990s, leaving room for integrated processes, in which the key words became shared reference tables and intra and inter field reuse. The PDM (product design management), then PLM (product lifecycle management) processes fundamentally organize work and bring the dynamics and flexibility necessary to increase both efficiency and reactivity.

Faced with this transformation within industrial project management, which goes beyond the simple adaptation of working tools to provoke a true evolution of working practices, it is crucial for general contracting to also adopt practices which will favor this change: if it does not make its acquisition method evolve, it runs the risk of not benefiting from the scaling economies often practiced by its suppliers, and cannot control the risks and costs in a way that should be natural.

1.6. The need for standardization

The whole approach previously touched on, materialized by the engineering and simulation infrastructures – which are the current two major projects in the implementation of an increased capability in regards to the defense systems’ architecture – must go along with a standardizing approach, with the creation of the adequate methodological reference documents, and the definition, and later appropriation, of the necessary standards. This approach must cover the entire chain of information which have to be manipulated: systems and models description data, validation data of these various components, data relative to the engineering process of the actual system, data (or even metadata) of the infrastructures.

An ambitious vision to be sure, but facilitated by the very recent developments in software and system engineering, as well as the standardizing efforts and the wish of a vast international community, regrouping industrialists, academics and public and private establishments, for a coherent and integrated vision of the various running standards: UML 2.0 for data modeling, SysML for the modeling of systems and their relationships with their environments, ISO10303-AP233 for the exchange of product model data, MDA for model and metamodel transformation, ISO 12207 for the software engineering process and ISO 15288 for the system engineering process.

One of the prime objectives of standardization is the reduction of the global cost and the marketing delays without sacrificing either the performance or the quality. Controlling this activity is an integral part of a successful global engineering process. Indeed, favoring the use of hardware, software or subsystems, grants an immediate advantage when their use answers the client's demands in regards to performance, and represents an advantageous cost. However, we should make sure that the products' architectures are sufficiently open and flexible so as to enable the integration of new technologies into the design of these products, and have the tools intelligently manage the architectures, structures, systems, subsystems, interfaces, designs. In fact, this is the base of every reuse process when we want to garner all the possible profits from it. The higher the standardization, the easier it is to capitalize on the use of off-the-shelf "components" (which are more and more abstract).

Standardization usually starts out being *de facto*, meaning that a product or service gathers so many market shares that it imposes itself as the reference every customer will turn to, and then becomes *de jure*, meaning that the market agents regroup into consortiums and assert their knowledge within internationally recognized standardization organizations. We could criticize this state of things by saying that the initial situations of customer lock-in or even monopoly (which impose the standard *de facto*) should be avoided, for they make us fear a leveling off of all innovation which wouldn't *a priori* be profitable on the market; but in fact their interest lies in the fact that it is the market which supports both the initial stage and the evolution. Moreover, this guarantees a better dynamic of evolution of said standardization, for the feedback is immediately taken into account by the evolution groups. The important step is the moment where we switch to the *de jure* standard, hence towards a loss of the monopoly which gives the customers access to a broader community, while still providing a product or service of good quality, since it has survived market law.

A standard, as a document of reference, brings answers to technical and commercial questions, and is elaborated by all the market agents (producers, users, laboratories, public authorities, consumers, etc.). This is the best compromise at a given time between the state of a technique and the economic constraints. Moreover, it is a document of voluntary and contractual implementation. Moreover, a standard helps exchange (leading to the harmonization of rules and practices on top of the strict regulation constraint), develop (facilitating the transfer of new technologies), orient (the community character of a standard contributes to the user's information, hence a factor of trust), and finally innovate (a standard is not a finite object, it is in constant evolution: adopting a standard helps anticipate the market's needs by making our products or services evolve according to the current best practices). Let us insist on the exchange aspect: the interest of a standard, of a shared knowledge, is to help newcomers be instantly operational; it also helps facilitate communication

within multipartite teams such as are necessarily met within systems of systems, and erase possible cultural barriers when faced with certain tasks, in particular technical tasks.

These various points are all the more important in a field where innovation is constant and has important economical impacts, now and later, because of the increasing integration of systems. It is therefore essential to control the entire process, and standardization on various levels (the level must be correct, or it might act as a hamper) is a regulation factor. Having standards on the various levels is just as important as achieving global coherence between the various standards. There lies the main challenge, for a standard must, on the one hand, not hinder innovation, and on the other hand, be adopted by a large industrial base which alone guarantees a market broad enough for its development.

Only then, with the help of these tools, methods, standards and principles which guide it and assist it in its decision making, will the subtle art of systems of systems architecture become a science, if not accessible to others, at least leading to much smaller risk retention, delays and costs.

1.7. The human factor in systems of systems

Far from being ignored, the human factor has been implicitly studied since the beginning. It is necessary to first clarify man's role on each stage of a system of systems' life cycle: end-user and customer, during the preliminary analysis of needs and the preparation of the acquisition process's launch; architect during the detailed design; designer once more, and also agent, during the development and production; end-user during actual use.

1.7.1. *The user: operator, supervisor, decision maker*

As a user, whether it be as an operator or a supervisor, man imposes ergonomic constraints, in truth regular interface constraints, depending on the nature of the input and output he can manage. Such a systemic vision, which considers man as a system of its own within a more global architecture, may shock some people; in fact, it only represents a certain way of taking the physiological specificity of human beings into account! This factor first translates into survival, or even comfort, requirements for the operator (space, temperature, pressure, impact requirements, etc.) and the presentation of information via the sacrosanct man-machine interfaces (color codes, adapted symbols, search of adapted representations, etc.). For systems of systems like those previously considered, this implies necessary improvements of

communications, comprehension of the situation, and the capacity to exert control and command.

It is important to see that the operator can act either on an elementary system, or on the interface, or on the intersection of several systems. In that case, his role becomes essential, for the global performance depends on him. For example, in anti-ballistic missile defense, the operators are on the level of certain sensors for the detection function, others are on the level of the pursuit consoles and among the decision centers, and finally others are in command of the terminal weapon systems. The levels of operation differ greatly from one another, with more or less important functions of decision, and mostly a more or less direct impact on the capability efficiency. General architecture must therefore consider the human factor, on the functional level as well as on the level of physical interfaces.

You have to put things into perspective when looking at the advantages and drawbacks of the human factor: he might represent a weak link for certain tasks, especially repetitive tasks, due to his eventual fatigue, and might therefore become an overload spot which could put the global performance at risk. But on the other hand, he has the ability to rapidly process high-level information even if it is incomplete, imprecise or uncertain, as well as capabilities of inference, not only deductive (find a particular example by using a general rule), but inductive (find a general rule to describe particular examples), or even abductive (introduce a rule as an hypothesis in order to consider the observed example as a particular case falling within that rule).

However, the use stage is not limited to the system operation, far from it. In particular, the training must be taken into account: this is all the more important since systems of systems have to be flexible in their use, which means we have to be trained to use them in extremely varied contexts and configurations. It might be interesting to design the system of systems in part for training purposes: here is a corollary of the aforementioned law of requisite variety, insofar as control of the system will be impossible if said system has a level of complexity much higher than its operator's. For the operator to take into account training capabilities (while not neglecting any of the roles that might be taken on by the human as an agent within the system), supplementary requirements must be reflected upstream, during the system's design. Moreover, still considering a system of systems' desired flexibility, quick feedback must be available, as much on the operation as on the training towards physical, functional or architectural requirements. The economical component necessary for control of the global ownership cost must not be forgotten, as training costs contribute to it recurrently and thus significantly.

To put it in a nutshell, you can see the privileged (or at least essential, for it fundamentally impacts the global performance) spot occupied by man, as a key-user who has to be the best user possible.

We have mentioned the man-machine interfaces: there too the physiological limitations of man must be taken into account, in that the man-machine interface must not create specific tasks linked to its use and which would not present any capital gain in terms of global performance. Typically, the interface should not ask its user for resources which could be dedicated to other activities. This is not so much a question of the interface's ease of use – on the contrary, this ease could go against the objective of optimization of its user's performance; there has to be a quasi symbiosis between the interface and its user – than the adaptation of the interface to the sum of the tasks gone through by the person using said interface. This point is far more difficult to master in a context of systems of systems than in one of an isolated system where every task is perfectly defined and fixed, and where the optimization can therefore be performed once and for all.

The reason for this remark is that today, all armies are faced with an important downsizing of their workforce, because of operating costs: this creates new system requirements, namely the downsizing of crews (the typical objectives in the different naval forces for surface vessels are in the region of 50%) while obviously maintaining the global performance; this means hunting down operator's tasks with no capital gain, and a search for the operator's best added value. The desired impact in economical terms is the control or even the reduction of the mission's total cost.

1.7.2. *Operating system support*

Let us now study the process of operating system support. Here, man is an actor of this maintenance, and his limitations should be taken into account to minimize the length and complexity of this task. A simple example to shed light on the subject: for engine maintenance, access to all the parts is essential. If the engine must be taken out in order to perform small repair works, we can imagine the time lost and the added cost, arising first from the taking out, and then from the possible need for particular expertise that will need to be deployed and maintained (added workforce, training of said workforce, etc.) From the operational standpoint, that is to say in the course of a campaign, you also need to take into account that the maintenance will bring the system to a standstill. It is custom to talk about the level of intervention in the military: at the lowest level, maintenance is directly performed by the system's servant, which technically allows a minimal immobilization, or even the possibility of not interrupting the mission. On the next level, the service is interrupted in the course of the maneuver, and dedicated teams must work on the system. On the last level, the immobilization might last for the entire mission, dedicated teams must

operate, and the complete system might have to be sent back home. The impact of a bad level definition in terms of availability and costs is obvious: it is therefore essential, during the design and later the development of the system, to take into account the integrated logistical support in close relationship with the profiles of both users and human resources who might be present or available in the various contexts of the system's use.

Following the growing complexity of systems of systems, those points have an equally growing impact. On the one hand, the multiplication of elementary components naturally heightens the risk of malfunctioning; but there is also the fact that maintenance of an element will instantly have an impact on the global availability. It is therefore of great importance for the support to be taken into account on a global level. In recent conflicts, the critical importance of logistics was highlighted, for its help in achieving success with an operation in which the effect's complementarity is crucial – it might even be a possible cause for an operation's failure – to the point where it becomes essential to diminish the elementary systems' unavailability, and most of all to integrate the intervention level within the system of systems' desired nominal operation.

1.7.3. Designer

Man holds another position worthy of being put in the spotlight concerning systems of systems, which had *a priori* no reason to be for the simple system: the position of designer. Why? Because the system of systems does not have a fixed design. As we have said from the start, it belongs to a capability approach, the expression of needs and the requirements are not fixed during its entire life cycle, and a mid-life renovation cannot remedy this. It is then that the designer comes into play, for he must be able to follow the evolutions in capability, by keeping track of and minimizing the impacts on the systems, focusing on the global architecture, for essential reasons of deadlines and costs. However, if the designer's role is fundamental during the entire life of the system of systems, it becomes essential to take into account his specificity, in order to optimize his work and his efficiency! Since the designer is, *a priori*, man, he has physiological limitations. We must control all the information linked to system of systems engineering, and the impact of an evolution or an obsolescence on the system's architecture, be it functional or physical. The mass of information, and its relational complexity, can of course be processed with the assistance of a computer, but it is also important to minimize this work: design itself must be designed so as to facilitate its configuration management, allowing easy operation by man. For a few years, this issue has been the subject of modeling engineering and software engineering research, in which the purpose of retro-engineering techniques, on the one hand, and model transformation

techniques, on the other, is to enable the design's evolution, and the "on demand" insert of systems to achieve a new capability configuration when need be.

1.7.4. *Customer, supplier*

The last role allotted to man in his particular relationship with a system of systems is that of customer, or more precisely buyer, as well as the role of supplier. As strange as it may seem, we cannot neglect this aspect, since it has a direct influence on the control of said system's life cycle. As specified above, the system of systems is in constant evolution, on the level of its defining requirements, and of its components. This requires adapted purchase logic, which must also be sufficiently flexible to adapt to this issue; the man-induced constraint is relative to his role as buyer and supplier. Indeed, acquisition goes along with regulations, as well as competing enterprises and environments. These elements must evidently be taken into account, for they condition the purchase's feasibility. In particular, we find a link with the role of designer: within an incremental development framework, the buying process and the design process must be put in adequacy. Likewise, the integrated logistical support and maintenance policy must be mapped with the ability to execute the corresponding contractual actions, while taking into account the suppliers' fidelity all through that maintenance. Indeed, competition leads to the disappearance of some enterprises, the repurchase of others, evolutions of market strategies, all the more so since the defense markets are not always the most promising in the long run.

1.7.5. *The human factor in systems of systems: man's expectations*

Habits have changed as society evolved, notably with the distribution of new technologies. In particular, the interactivity between the public and systems of systems, such as the Internet or the global digital society, is now essentially based on the research and recovery of information or a service (pull mode), unlike the previous decade's habits, which were based on the principle of information supply (push mode).

In short, the human user becomes aware and demanding, wishes to have the right information at the right time, and *a priori* is not willing to comb through a block of information anymore. If he himself has to provide information, he expects to get useful feedback.

It is clear that these day-to-day habits have an impact on the users' profiles when they are faced with or integrated in a system of systems: the interactivity cannot be reduced to the simple role of operator, where man would only be at the commands

of certain levers, or would provide information within a complex decision process. Man will ask for feedback about what is demanded of him: his expectation towards the system of systems will therefore lean on a notion of fruitful exchange, a sort of win-win relationship. This is understandable, beyond the change of the simple societal context: faced with a system, man is indeed an operator, but he also has, to some extent, a dominant role. On the contrary, with a system of systems, this relation is lost: man is integrated, or even drowned, within the system of systems. It is precisely to avoid this feeling, which would go against the global efficiency that studying the human factor and his interaction with the system of systems is fundamental.

It is essential to consider this notion of interactivity, which goes beyond a simple interaction where exchanges would be limited to basic instructions without the use of any particular semantics. To illustrate, let us take the example of network-centric operations and the issue inherent to the establishing and distribution of a shared tactical situation. In such a context, all the subscribers must provide, in real time, their position and identification (which may be done automatically) as well as the eventual detected threats. So far, this is achieved through messaging systems alike the ones generally used in the civilian field. It is clear that, because of this analogy of means – the issue is not much different from an electronic business transaction with suppliers both distant and distributed within a complex chain: planning of a holiday trip through travel agencies, carriers, hotels, eventual pre-booking of services offered on the site, etc. – the user's expectations have a tendency to conform themselves to preestablished patterns. A certain delay is accepted for the answer, under certain conditions of due warning and prediction, but feedback is mandatory.

The same thing applies when we travel by plane: we are then thrown into the aerial transportation and traffic control system of systems. Adaptability, interactivity, feedback or the ability to provide information on request, have become daily requirements. It would be unthinkable, if we want man to subscribe to the system of systems in its use and operation, for it not to be the same in the field of defense.

But this is only an aspect of the necessary trust which must be established. The other aspect is linked to the system of systems' security: just like electronic commerce only really took off once the security of transactions could be guaranteed and trust regained, the information and operation security must be guaranteed.

Let us now look at the problems linked to reliability: it is the property which enables a system's users to place a justified trust in the service provided to them, or more precisely, all of a product's aptitudes which supply the specified functional

performances, at the right time, for the right duration, without any damage on itself or its environment. Reliability includes the notions of:

- availability (probability that the device will work correctly when asked upon);
- dependability (probability that the device will correctly operate for a certain time);
- maintainability (product aptitude at being put back into a given operating state, within the specified time limits, when the work is carried out following the prescribed procedures and conditions);
- and security (system aptitude at resisting against external attacks, whether natural, malevolent or involuntary).

These aspects are all the more important in the case of systems of systems, both because of the global complexity and the difficulty in formalizing the satisfaction of a desired capability. But even if it were possible to translate the latter in terms of quantifiable performances, there would still be the question of the adequacy of models representing man within the system of systems.

In order to determine man's optimum place within the system of systems' architecture, we must consider it *stricto sensu* like a component of this architecture, in the same right as all the other systems which constitute the system of systems in the entire engineering process: from the global analysis of the mission and the expression of need, through to the functional analysis and the allocation of functions – here man is a solution in the same right as others – and until the search for compromises resulting from the confrontation of the analysis and the functional and physical allocations.

At this time, it is licit to wonder whether man is a limiting or multiplying factor. In fact, in this analysis of the system of systems' global engineering, we are faced with the same question as with any other architecture: the analysis of the overload risk (transmission or treatment capability *versus* data flows) is the same than with a physical component, except that it translates into a concept *a priori* more affective. This character aside, what is the difference between an operator with a saturated load factor, which makes him inefficient, and a calculator saturated with too much data to process and which therefore suffers buffer overflow or must be reset? In both cases, the architectural component cannot properly operate anymore because of a physical limitation, which impacts all the components via the distribution of constraints. This leads to a constant search for added value, and the systematic tracking of potential capital loss for each component, human included. If, at first sight, the factors are, for the human being, a symbolic treatment and decision capability, and for the machine, the data processing and communication speed, these characteristics only constitute the top layer; with certain tasks, the roles might be

better switched, so as to reserve this natural distribution of roles for tasks carrying a bigger added value. Engineering must therefore be led on a global scale, *a priori* without elementary choices, within the most syncretistic vision possible.

A fundamental mistake that must be avoided is the use of man as a buffer to palliate certain obsolescence of the system, following the hypothesis that his cognitive abilities make such a thing possible: for example, facing him with obsolete interfaces, sometimes under the pretext of service continuity or already implemented training programs. For, in such conditions, the global efficiency finds itself diminished, insofar as some of the new components are not being exploited as well as they could be. Here lies the true challenge of systems of systems: the obsolescence does not only impact the elementary component, but, from the close integration of various generations of components, it impacts the other systems in the relationship, and through a distribution mechanism, the global behavior might be considered responsible.

A last aspect, once again linked to society habits, concerns the acceptability of man-machine interfaces: video games or various software aimed at a broad audience offer capabilities of information presentation which are both fun and appealing. This state of things should therefore be taken into account during the design and development stages. We might even think of making the interfaces of defense systems more “fun” – beyond possible ethical scruples – just like what is attempted in common software, even if only to lower the stress levels of operators.

The previous remarks might suggest that trend effects prevail; We might also object that daily life and the evolution of habits are under complete servitude to the technology imposed by suppliers who care little about human beings as intelligent users, or a source of added value during its interaction with the product, and instead see them only as clients to seduce, or to exploit. Subsequently, we might think that the design of systems of systems, which are used daily, is not guided by an architectural optimization of the human factor, but simply by the creation of a captive clientele. This is probably partly true, but this force of habit cannot and must not be neglected, for it might impact the acceptability of the entire system of systems. Introspectively, let us remember our reactions when a function disappears from a familiar product, even if we barely use it. Such an omission creates a feeling of unease, dissatisfaction, or even regret, etc., which momentarily impacts on our performance as a user.

We can wish that man is as rational as he claims to be!

1.7.6. *Standardizing the human factor in systems of systems*

Today, frameworks are available to help organize and plan the design of a system. Those are standardizing documents such as EIA 632, designed in the USA by the ANSI, and ISO 15288, which are becoming, little by little, the reference on which businesses base themselves to design complex systems.

Activities specific to the human factor are explicitly specified in the EIA 632 processes. These elements represent the first steps: however, there is still a long way to go. Integrating the human factors requires common methods and tools which will help organize, plan, manage and implement the human factors' activities, all through the system's life cycle. Those methods and tools do not, however, solve all the difficulties: other aspects, cultural, social and organizational, have important impacts on the integration of the human factors in the design.

First of all, the human factors' engineering process, and its activities, must be identified. To that effect, the process of human-centered design is specified within ISO 13407. It only treats the design process of computer systems and does not take into account all the dimensions and disciplines of human factors. For example, aspects about radioprotection, or the health risks induced by muscular or skeleton constraints, are not taken into account, just like other aspects about work overload, or goal conflicts in terms of double constraint on the operators' stress and their ability to act whichever the situation.

Later on, questions might be raised about the acceptance of human-machine interactions, in which case we can turn to ISO 16982, which builds on the aforementioned ISO 13407 as well as ISO 12207, which concerns software engineering.

But a process-oriented vision is needed. This is the focus of ISO 18529, which presents some of the human-centered process's requirements and activities, such as identifying and planning the users' role, planning human-centered design, identifying and documenting the physical environment, using the existing knowledge to develop design solutions. However, building on ISO 13407, it does not take into account the other dimensions of human factors. It must therefore be pushed on, through the identification of all activities, in terms of resources, skills, costs, methods, tools, indicators, flows, legal aspects, in short all the necessary factors to take into account and integrate the human factor in the design. Such an initiative should be integrated within a proper maturity model, such as CMMI (capability model maturity integration).

Beyond these first standards, as far as the integration of the human factor in the design process is concerned, the current works in terms of human factor engineering

(identification of the processes and activities, the skills and the activities' products) must therefore be continued and amplified, so the human factor may be understood as an organized engineering discipline, planned and managed like all other disciplines.

Let us now leave the world of standards for the world of efficiency measurement, which is its operational extension. The goal is both to better evaluate human performance within the architecture, whether as an operator, supervisor or decision-maker, and to optimize the human factor in the system of systems (his place in the architecture, the measure of the impacts technological changes have on the activities, the evolution of the interfaces, or, in more general terms, the human factor following performance measures). To that purpose, we should possess predictable, standardized models, as well as metrics.

These two attributes raise difficulties, insofar as the first insists on the behavioral aspect, which, beyond a possible apprehension of the underlying cognitive processes, requires people to have an idea of the way the human operator might fulfill the task entrusted to him. However, if this is a possibility for stereotyped operating modes, such as the operation of dedicated systems for which operational feedback helps know the load factors, the reaction delays, etc., it is much more delicate when applied to decision making, such as the classification of targets and the evaluation of situations, asked to the tactical coordination officer during a mission of air or sea surveillance. The standardized aspect is just as fundamental, for, without it, serious comparative architectural evaluations cannot be led; moreover, standardized models ought to take into account the various operational contexts, so as to correct the bias induced by stress, fatigue, or the ethical dimension attached to certain decisions. We can see the distance left to cross till models can fulfill all these objectives. If this question is already important in the case of systems, it becomes critical in the case of systems of systems: see the issue of a down-scaling of crews when complex missions must be carried out such as air defense. The question of the automation of certain tasks notably appears on the level of detection and classification, or even during the decision process which concerns the implementation of specific procedures. In order to determine the precise levels of automation and the logical architecture which would take into account the decision and data distribution processes, these standardized models might bring part of the answer. The quantification of performance gains is obviously completely dependent from these models' validation, which presents another technical challenge.

In order to address all these aspects, standards, metrics, models, the tools must help tackle the task's complexity. Indeed, gone are the days when an architect could claim to have an integral vision and detailed knowledge of his system. Would it be only with weapons systems, such as a military aircraft, requirements come in tens of

thousands; we can imagine the human challenge in mapping a system of systems' requirements, and ensuring the temporal coherence in terms of configuration management. If you add to this, varied and ever changing environments and conditions of use, as well as multiple interfaces with other complex systems, the task becomes inhuman! To compensate for these difficulties, inherent to systems of systems engineering, it is essential to have an integrated vision of engineering data, as well as any technical data (linked to the system, an equipment, user contexts and scripts, threats, etc.) with the associated filing, capitalization, management, update functions, etc. Without getting into details, such a thing requires the use of standards, methods and tools, specific to engineering and which "only" need to be faithfully and exhaustively applied on the level of a system of systems, in close relation with distributed simulation capabilities. Beyond the technical necessity of such collaborative engineering workshops, which pertain to the recent extended enterprise concepts, their economical advantage can also be proven.

Concerning the issue we are interested in, namely the study of human specificity, it should be mentioned that man is not forgotten in those various tools: in part behaviorally or phenomenologically simulated, when the validated data and models are available, it is often physically integrated, in particular when its contribution as an architecture component is *a priori* perceived as critical. Facilitating technologies are found in augmented reality (which is different from virtual reality in that man is not thrown into a purely digital virtual world, but a hybrid world in which hardware components are enriched or completed by virtual decorum), as well as digital virtual prototypes, which become more complex and more efficient every day, thanks to everlasting technological progress in computers and visualization means.

1.8. Budgetary aspects of the systems of systems

As capability studies effectively translate into the development and use of systems of systems, via continuous spiral processes, the question of budget comes into play. If the existing cost models help gauge the efforts necessary to develop or modify constitutive systems, they cannot apply to the development, integration, test and maintenance of the system of systems "top layer" which actually helps avoid a simple juxtaposition of systems without any actual gain in terms of added value.

In fact, cost models which are available on the level of individual systems cannot just be transposed, for they do not place enough importance on strategic reengineering aspects (that is to say, the organizational and functional modifications necessary to achieve a common operation of systems, in order to create a capability) and the coordination of potentially independent proprietary project entities (a system features a global project manager, and the coordination cost is actually the cost of integration: we will come back to this point in the following sections).

Studies are ongoing to design such cost models, COSOSIMO (Constructive System-of-Systems Integration Cost Model) in particular, which would then complete the traditional toolbox of cost models such as COCOMO II (Constructive Cost Model), largely used in the field of software-intensive systems, and its extensions COCOTS (Constructive Commercial-Off-The-Shelf Cost Model), COSYSMO (System Engineering Cost Model), which broach the cost aspects linked, respectively, to the integration of off-the-shelf components, and system engineering. Such a systems of systems cost model must identify: the contributing key-factors, such as width and complexity, the architecture's maturity level on the level of the system of systems, the schedule demands, the integration risks, the maturity and stability of the constitutive systems, the validation level of each constitutive system, the integration team's capabilities and the integration processes' maturity on the level of the system of systems. The following subprocesses might require specific expenses: planning, requirement and architecture management, but also the selection and supervision of all subcontractors, as well as the integration and testing stages, and the transition between one capability increment and the next.

Among the costs of systems of systems, those of the organizations which buy and use them should be taken into account, even more importantly when the system of systems is a value chain centered on organization. The notion of transaction cost is found in a certain number of economical organization models, generally defined as the transfer between breakable units of user rights over goods or services. A transaction's capital gain lies in the way the ownership rights over physical or even virtual assets provide some agents with a lever to command the action of other agents in need of those assets. The approach, in terms of transaction costs, provides a coherent explanation to the existence of settings which organize the production and exchange, and the arbitration, between these modes. The choice between such or such a setting will depend on the transaction's characteristics, and the contractual risks it generates. The organization, as a structured entity which coordinates the actions of the parties within a set of decision rights and systems, whose articulation determines the efficiency, or lack thereof, of the choices, is an instance of coordination based on two complementary modes: communication and command. The first mode is built on the circulation of information, and the associated costs, whether they be buried costs linked to the infrastructures, or costs linked to the specificity of the human asset. The second mode helps reveal the important information, filtering the signals so as to facilitate the decision. To make this dichotomy more flexible, to shorten the time needed for decision making, to facilitate the transfer of information, the organization is also a place of negotiation, based on contractual relations between members of the organization, something which helps formalize, or even pilot, the transactions between members. Within the implementation of capability approaches in the world of defense, where service level commitments replace the obligation of means, this notion translates into "operational contracts", which help regulate the relations between acquisition

services, operational users, and users within an alliance (NATO, for example). The difficulty lies, however, in the possibility of finding an acceptable exploitation of these organization-related transactions, and in integrating it within the global cost of the system of systems.

1.9. The need for governance

As the systems become imbricated into larger systems of systems, the constitutive systems' development teams and stakeholders will realize the benefit of having adapted governance processes (of creation and implementation of general policies), and participate to these processes. Indeed, in a context of systems interconnected into networks, it is clear that no supreme authority can claim to own, control or monitor everything (even if such an authority may seem to exist on organizational charts: for example, in the field of defense, regardless of the varying denominations in each country, while the chief of the defense staff or the secretary of defense occupy this position on the chart, they do not play this part in the day-to-day life of the system of systems), unless it is in a cooperative, distributed manner. From there stems the risk that, on the level of the individual systems (and the entities in charge of those systems), policies with negative effects on the systems of systems might be implemented. Governance itself is more akin to an exchange rather than the imposition of one group over another.

This is strengthened by the way individual systems have different rhythms and periods of evolution. Hence the difficulty in coordinating all these changes in adequacy with the phasing of the global system of systems' increments, and the risk of divergence, in particular in terms of compatibility and interoperability, is obvious: the issue is all the more acute for the interfaces between systems, which often crystallize the conflicts in priority and control. Hence an increased need for clear policies regarding the evolution of the components in relation with the global interoperability, as well as evolution and configuration management methods to share with the other components.

The acquisition process which we have talked about previously, in which incremental and spiraling developments alternate, presents advantages and drawbacks, especially when governmental agencies are involved. Among the advantages, we have already mentioned its flexibility, and the possibility to take into account the users' feedback among acquisition teams more quickly. Among the drawbacks the following can be mentioned:

- The difficulty of maintaining a governmental control of the process: besides problems of schedule management and contractual vectors, we might wonder about the government's ability to control the spiral stage. Whether it concerns the technical skills necessary to evaluate the risks, or the ability to arbitrate in the event

of a disagreement with operational users, the governmental agencies are naturally led to take on a level of responsibility they are not always prepared for (notably in terms of human resources). Various approaches might be retained, from the quasi total transfer of responsibility towards industry (such is the case with system integration project managers) to the other extreme which consists of a very high involvement of said agencies and them taking on the complete role of integrator.

- A complex configuration management: the multiplication of increments greatly heightens the complexity of configuration management, in particular if a great many copies of the system are installed. It is important to understand that here, the problem arises from incremental development, and not from spiral development. We are therefore faced with a well-known problem, which would seem to be unavoidable, considering the permanent, fast evolution of technologies, in particular when they concern computer sciences or electronics.

- Too high a variability within successive increments: surely, a heightened flexibility is always a good thing, but a minimum stability should be maintained. Whether in the form of alternation between the users' various schools of thoughts and fashionable technologies promoted by engineers, the risk of zigzags around the proper path must not be overlooked. Too broad a lack of continuity in the various delivered increments might lead to delicate problems in the users' training, and might even impact the efficiency of the system's final implementation.

1.9.1. *New models of competition*

Previously, competitiveness rested on our ability to dominate one another, searching to possess as many resources as possible at the expense of the competition. Today, this competitiveness is more about collaboration, through the exchange of goods and services but also knowledge, while seeking our own niche so as to have our added value stand out. From a dominant/dominated model, we move on to models of codevelopment, coproduction, comarketing and codistribution.

Going from a model of integration to one of cooperation requires us to know how to decompartmentalize and remain open to opportunities, hence to know how to permanently call our economical model into question. Indeed, competition still exists, but its shape is different; most importantly, the agents are no longer systematically competitors, but often partners, with the possibility of standing out (a neologism was created to describe this situation: "coopetition"). These cooperations, or these partnerships, are not only motivated by opportunities, for the exchanges are too restricted and, in the long run, will not provide added value. In fact, the agents come together to create value with the purpose of finding the ideal compromise between contributing to this cooperation which creates useful value, and differentiating ourselves through added-value usage. Obviously, all these

observations are both valid in the business world and in a political-military context, with the various notions of coalition, partnerships and alliances, whose advantages and drawbacks have been observed in the last decades.

From the standpoint of the end-user, this mutual interdependency of needs, services and agents in charge of their development becomes more and more transparent: what is purchased and used is no longer a product or a set of products, but an end-to-end, on-demand service. The potential problems concerning the integration of the acquired components do not interest the end-user anymore, but are transferred to the global supplier project manager. If this provides the user with a heightened flexibility and resilience concerning the use he might make of the global service, it however transfers these requirements to the supplier and leads to the redefinition of the latter's role and responsibility.

Typically, the air transport system of systems, from the booking of e-tickets to the actual transport, including housing and the possible supply of various rental services, must now include the evolutionary safety measures (following the various terrorist attacks) on top of air traffic management: this requirement's difficulty, but also the system of systems' global resilience, were demonstrated upon the terrorist attempts in London, in the summer of 2005, which led to immediate safety measures, leading in turn to huge waiting lines and significant traffic delays for a few days, but which were rapidly reflected, as much in terms of delays as in terms of costs, over the various links of the system of systems.

This is not an obvious process, and it is clear that simple technological innovation and the decrease of the cost of technical infrastructures are not sufficient conditions for the improvement of the performance/price ratio. The latter goes through the renovation of organizations to permanently adapt them, a key-element for the system of systems' success, as well as through a true reflection on the associated governance, namely the means necessary to implement a certain order in relationships within which potential conflicts threaten and might cancel or compromise opportunities of achieving mutual gains.

1.9.2. *New organizations*

Organizational intelligence is a key factor in product differentiation, and it would be a mistake to ignore it to focus on technical and/or functional performances. Just as an enterprise's assets are not evaluated on their patrimonial capitalization alone, but on their ability to create wealth, likewise the system of systems, via the search for the value chain's optimization, demands that the various available levers be open to influence. The rigidity of the organizational and institutional structures is a hamper, and often a handicap, against the unpredictability of events. The

observation is easy to make, but how can the various rhythms be synchronized again? With pluriannual investments, reorganizations whose implementation is made all the harder by their depth and the context's unforeseeable evolutions, along with possible strategic upheavals. There lies the true challenge of the system of systems, combining network capabilities – which can heighten agility and resilience as long as it does not fall in the traps of overly rigid and systematically balanced contractual relations – with the value chain logic, which helps it achieve abstraction and therefore free itself somewhat of such or such product's particular environmental conditions and instead create a global value based on the transformation of material or virtual assets.

Something becomes obvious: the stakeholders must be grouped as early as possible, and as often as possible, before preliminary designs, and the end-users must be included. In a system of systems, just as in a transforming enterprise, three essential stages must be performed with the users: report on what is existing, definition of the long-term target, and transition map from the current position to the target position. Only then can a target be designed which will correspond to users' needs, and most of all a realistic transition plan which will give meaning to the various capability increments which are being defined. An architecture can then be designed, with its requirements and constraints, and certain options can be validated in real-time.

An organization which so gathers the various stakeholders is bound to achieve success when faced with the various interpretations those stakeholders may hold on the supplied products or services. It also helps, if not totally discard, at least benefit from those stakeholders' various cultures. This notion of an integrated team is all the more important since it is not usually possible, as we have already mentioned, to define an "owner" – that is, a person or an entity which would own or wield authority or control – of the system of systems (stemming from the system's complexity, but also because it is composed of several systems which have their own general contracting and project management organization, with possible constraints on industrial and intellectual property). Moreover, the global planning cannot be designed *in abstracto* without taking into account operating modalities, for the same reasons of managerial and budgetary independence of some of the constitutive systems. The organization must therefore gather, but not dilute, everyone's responsibilities, while getting them to work in a common direction and towards a goal greater than each individual team. Evidently, this goes through the acknowledgment and explicit prioritizing of such goals, and therefore through an enterprise governance which goes beyond individual projects (and local priorities), regardless of their importance.

1.9.3. *New relations between project management and general contracting: the role of system integration project managers*

In fact, the main argument seems to lie in the responsibilities of the various agents which will operate during the life cycle. Indeed, in the case of a system, a large part of the acquisition can be delegated, in terms of risks, to a project manager. In theory, the same thing can be done with systems of systems, through the appointment of a lead system integrator (LSI), that is to say an organization chosen to supervise the definition, development and implementation of the system of systems, in charge of the concurrent engineering of requirements, architecture and planning; the identification and evaluation of the technologies to be integrated; the selection of suppliers (*a priori*, the development of the system components does not pertain to him); the management and coordination of supplying activities; the validation and evaluation of the system of systems' architectural feasibility.

Some experiences have been led, mainly in the United States, but also in Europe, with mixed successes, in particular in terms of the control of costs and delays. In the case of system of systems, this actually raises a dual problem:

- considering the economical equation, it does not seem generally possible to simultaneously purchase all the components of the system of systems, something which requires people to take into account currently operating systems (which might have to be upgraded), systems in the design stage, and finally other systems which are either still planned, or in even earlier stages. This calls on many project managers, with potentially different contracting conditions (in time, the geopolitical and socio-economical contexts lead to the evolution of the acquisition procedures), which *a priori* complicates the achievement of a system project management which could, on the one hand, integrate a set to turn it into a true system of systems and not a simple juxtaposition of systems, and on the other hand, could take on the responsibility of this integration;

- more often than not, acquiring the system of systems requires – especially at the beginning of its life cycle – an incremental analysis of needs and the formulation of requirements; entrusting this work to a (unique) project manager would be risky, considering the stakes.

The first point is critical: risk sharing is a blocking point. In order to achieve a true system integration project management, which will fully assume the development of systems of systems; we must find contractual modes which will use innovating mechanisms of financial incentive, so the project manager may be encouraged in having a global vision of things, while satisfying the milestones related to individual tasks. Global performance commitments may be envisioned (akin to the service quality commitment used with information systems): it is not easily achieved within a capability rationale in which an effect is sought, more than

a performance, the conceptual difference lying in the way the effect is linked to the application target, whereas the performance is linked to the vector which provides said application.

A possible solution would be *a priori* to follow a model of team partnership integrated with the global project manager, and to share the risks, or even the conflicts of interest, between the general contractor and the project manager via strategic agreements and financial incentives anchored in time.

This notion of partnership is essential: let us not forget that most of the technological base comes from project management, hence the importance of knowing the latter and working hand in hand with it to exploit, on the level of the systems, the room for maneuver potentially provided by the technological improvements on the components' level. In many cases, the desired technologies have not been fully demonstrated on the scale of the system of systems and their level of maturity is not sufficiently high for the scaling to happen without risk. The choice is difficult: should we integrate fully mastered knowledge and risk not being able to optimize the value chain on the expected level, or take the necessary risks, knowing how to contractually share the level of risks between general contracting and project management?

To sum up, the global management of a system of systems, answering a strategic capability on the scale of a nation (whether it pertains to defense and security, or to a public service critical for the national economy), implies such a strategic positioning towards, on the one hand, the industrialists concerned by the various systems in terms of transfer of responsibility and the transfer/sharing of risks, and on the other hand the reflection of need (in the present case, a non-negligible part of the defense tool or the public policy), that it can only be taken on by the public persona. However, this does not mean that the latter should take on the role of integrator at every stage, from the definition of the capability need to the role of operator, through the offer and evaluation of global architectures, detailed engineering and development, and the integration of pre-existing systems. Both value and risks must be analyzed for each stage, to evaluate on a case by case basis which position to adopt in terms of steering and responsibility.

1.10. Conclusion

We have demonstrated that the notion of a system of systems fully belongs within the evolution of a society driven by technological progress, and that an increase in complexity is ineluctable. It would be useless to try and avoid the issue under the pretext that it is still in the early stages of development, at least compared to systemics and system engineering, for the transformation is ineluctable. We have

therefore offered a definition which has the merit of tackling, grouped within the concept of value chain, the three pillars (the “3P”s, People, Process, Product), that is to say organization, process or function, and technology.

It is possible to object that various concepts, such as the creation of value, networks, flexible organization between agents, far from being revolutionary, have always been used. Indeed, Adam Smith’s economic models on the creation of wealth are still topical today; the commercial empires, from Venice to the Commonwealth, through the Hanseatic League, are historical proofs of these concepts.

However, it is now obvious that the evolution dynamics have changed, and that the coordination, negotiation, trade and lastly growth capabilities have become much more important, leading to deep breaks in the design and organization of systems. Due to the compression of certain time constants, some older methods are no longer valid, as their implementation is no longer compatible with some of the current cycles. Hence the necessity of making them evolve and most of all equip them, something we have tried to highlight.

Among the fundamental changes, we have insisted on the human factor – the user need not be the most proficient, so as to avoid any bias – for the human being becomes an essential link: he is the one, *in fine*, who is in direct contact with the time constant requirements of the system of systems. He therefore plays an essential role in the adaptation and transformation of what already exists to fit the current requirements, and in properly anticipating future evolutions.

It should also be pointed out that this systems of systems issue is globalizing, insofar as it is not an epiphenomenon which might be contained and reduced, as much in terms of the application field as the evolution of working methods. Without pushing the comparison too far, the situation is a bit like the one of the automotive industry, in which the manufacturing system used by Toyota did not only pull the company to the top, in front of American giants which had been ruling over the field for years, but also transformed the entire industry – insofar as the competition had to adopt the same just-in-time production techniques to try and become competitive once more – but also other sectors, such as aircraft construction, the manufacturing of IT equipment, or even mass-market retailing. Likewise, systems of systems go hand in hand with a deep transformation of enterprises and the relations between agents, which, through a domino effect, leads to the progressive transformation of the entire society, within a global economy, with new ways of understanding complex problems.

1.11. Appendix: system of systems' definitions in literature

The precise bibliographical references of the following quotes are not included in this chapter's bibliography, but can be found within this very bibliography.

[*Anderson, Campbell and Chapman, 2003*]. Systems of systems are characterized by complex combinations and interdependencies of technologies, operations, tactics, and procedures.

[*Army Software Blocking Policy, Version 11.4E, U.S. Department of Defense, Department of the Army, 2001*]. A system of systems is a collection of systems that share/exchange information which interact synergistically.

[*Army Acquisition Policy, Army Regulation 70-1, U.S. Department of Defense, Department of the Army, 2003*]. A system of systems is a set of arrangements of interdependent systems that are related or connected to provide a given capability. The loss of any part of the system will degrade the performance or capabilities of the whole. An example of a system of systems could be interdependent information systems. While individual systems within the system of systems may be developed to satisfy the peculiar needs of a given user group, the information they share is so important that the loss of a single system may deprive other systems of the data needed to achieve even minimal capabilities.

[*Association Française pour l'Ingénierie Système, 2006*]. System resulting from the collaborative operation of constitutive systems which can operate in an autonomous fashion to fulfill their own operational mission.

[*Bar-Yam, 2004*]. Systems of systems have the following characteristics: evolutionary development, emergent behavior, self-organization, adaptation, complex systems, individual specialization and synergy.

[*CapDem program, Canadian Department of National Defence, 2006*]. A system-of-systems is an assemblage of components that individually may be regarded as systems and that possess two additional properties: Operational independence of the components: If the system-of-systems is disassembled into its component systems, the component systems are able to operate independently; that is, the component systems fulfill customer or operator purposes on their own. Managerial independence of the components: component systems are separately acquired and integrated, and maintain a continuing operating existence independent of the system-of-systems.

[*Carlock and Fenton, 2001*]. System of systems engineering is focused on coupling traditional systems engineering activities with enterprise activities of strategic planning and investment analysis.

[*Carney, Fisher and Place, 2005*]. Systems of systems have an evolutionary context that carries an interoperability relationship between systems that is preservative and adaptive.

[*Cook, 2001*]. A system of systems is a set of interdependent systems evolving at different rates, each at a different phase of their individual system life cycles.

[*Crossley, 2004*]. A system of systems is a mix of multiple systems, each of which are capable of independent operation but must interact with each other in order to fulfill the global mission or missions. The mix may include existing and yet-to-be-designed independent systems.

[*de Laurentis, 2005*]. System of systems problems are a collection of trans-domain networks of heterogenous systems that are likely to exhibit operational and managerial independence, geographical distribution, and emergent and evolutionary behaviors that would not be apparent if the systems and their interactions are modeled separately.

[*de Laurentis and Callaway, 2004*]. The combination of a set of different systems forms a larger system-of-systems that performs a function not performable by a single system alone [...] In fact, system-of-systems generally have the following distinguishing traits: physically distributed systems, prime dependency of overall functionality on linkages between distributed systems, and system heterogeneity, especially the inclusion of sentient systems, for example, thinking and evolving individuals or organizations. Looking deeper, some basic requirements for effective system-of-systems lexicon can be observed: (1) diverse parties understand the description, (2) all relevant portions of the problem are included, (3) it facilitates the recognition of the boundaries of interactions, “seeing the forest for the trees” (the holistic perspective). Thus, the lexicon must include the ability to understand both hierarchy and organization.

[*Defense Acquisition Guidebook, U.S. Department of Defense, 2004*]. The set of systems comprising the systems of systems are independently useful systems, yet when integrated together, they deliver significantly improved capability. A single system or less than full combination of all systems cannot provide the capability achieved by the system of systems.

[*Eisner, Marciniak and McMillan, 1991*]. A system of systems is a set of independently acquired systems, each under a nominal system engineering process;

these systems are interdependent and form in their combined operation a multifunctional solution to an overall coherent mission. The optimization of each system does not guarantee the optimization of the overall system of systems.

[Eisner, 1993]. Systems of systems are large geographically distributed assemblages developed using centrally directed development efforts in which the component systems and their integration are deliberately, and centrally, planned for a particular purpose.

[Gur and Levi, 2004]. Systems of systems are actually a collection of systems that are created by different development teams and form as single application... Systems of systems synergize the performance of all systems within the parent system and should be ready for failures within daughter systems to prevent regression in the total organization productivity (the domino principle)... Permissions define which entities (commands and controls) of the systems the users can operate and which they cannot (add, copy, update, etc.). Compartmentalization organizes clusters of data and information which a defined group of users can access while other users cannot. A system of systems enables its applications to share and interchange information between one and another, of course each application can retain its own data differently from another, but the interchanging should be transparent and considerable to appropriate security policy... To enable the application more scalability, availability and adaptability the application components need to build such way that enable them to work in clusters. Such clusters should contain a dynamic number of components that can be routed to handle coming requests.

[Holland, 1995]. It is feasible to understand any system of systems as an artificial complex adaptive system. It is manufactured to achieve a predefined mission and will involve a large number of interacting entities with persistent movement and reconfiguration, changing based on changes in context, ordered through self-organization, with local governing rules for entities and increasing complexity as those rules become more sophisticated.

[*Joint Capabilities Integration and Development System (JCDS)*, U.S. Department of Defense, 2005]. A system of systems is a set of arrangement of interdependent systems that are related or connected to provide a given capability. The loss of any part of the system will significantly degrade the performance of capabilities of the whole. The development of a system of systems solution will involve trade space between the systems as well as within an individual system performance. An example of a system of systems would be a combat aircraft. While the aircraft may be developed as a single system, it could incorporate subsystems developed for other aircraft. For example, the radar from an existing aircraft may be incorporated into the radar being developed rather than creating a new radar. The

system of systems in this case would be the airframe, engines, radar, avionics, etc. that make up the entire combat aircraft capability.

[*Keating et al., 2003*]. A system of systems is a metasystem, comprised of multiple embedded and interrelated autonomous complex subsystems that can be diverse in technology, context, operation, geography, and conceptual frame [...] These complex subsystems must function as an integrated metasystem to produce desirable results in performance to achieve a higher-level mission subject to constraints.

[*Knisley, 2005*]. A system of systems is a complex purposeful whole that: is composed of complex, independent, self-organizing, component parts whose high levels of interoperability enable them to be recomposed into different configurations and even different systems of systems; is characterized by poorly-defined issues that significantly affect its behavior and make it difficult to understand; has ambiguous boundaries with critical contextual influences involving a mix of technical/non-technical factors; and exhibits emergent nonlinear properties. The complexity of a system of systems is a function of the number and diversity of its components and their linkages. System of systems linkages range from loosely to closely connected, but all systems of systems exhibit non-deterministic evolution and behavior and are cybernetically self-organizing.

[*Kotov, 1997*]. Systems of systems are large-scale concurrent and distributed systems that are comprised of complex systems themselves.

[*Krygiel, 1999*]. A system of systems is a set of different systems so connected or related as to produce results unachievable by the individual systems alone [...] A particular system of systems may be configured and used for a period of days or weeks to support a mission-transient operation. Other combinations of systems may be integrated and sustained for longer periods of time.

[*Lane and Valerdi, 2005*]. In the business domain a system of systems is the enterprise-wide integration and sharing of core business information across functional and geographical area [...] In the military domain, a system of systems is a dynamic communications infrastructure to support operations in a constantly changing, sometimes adversarial environment [...] For some, a system of systems may be a multi-system architecture that is planned up-front by a Lead System Integrator [...] For others, a system of systems is an architecture that evolves over time, often driven by organization needs, new technologies appearing on the horizon, and available budget and schedule [...] A system of systems is the integration of existing systems into network-centric, knowledge-based systems.

[*Luskasik, 1998*]. System of systems engineering involves the integration of systems into systems of systems that ultimately contribute to evolution of the social infrastructure.

[*Maier, 1996*]. Five principal characteristics are useful in distinguishing very large and complex but monolithic systems from true systems-of-systems: (1) Operational independence of the elements: If the system of systems is disassembled into its component systems the component systems must be able to usefully operate independently. The system of systems is composed of systems which are independent and useful in their own right. (2) Managerial independence of the elements: The component systems not only can operate independently, they do operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system of systems. (3) Evolutionary development. The system of systems does not appear fully formed. Its development and existence is evolutionary with functions and purposes added, removed, and modified with experience. (4) Emergent behavior. The system of systems performs functions and carries out purposes that do not reside in any component system. These behaviors are emergent properties of the entire system of systems and cannot be localized to any component system. The principal purposes of the system of systems are fulfilled by these behaviors. (5) Geographic distribution. The geographic extent of the component systems is large. Large is a nebulous and relative concept as communication capabilities increase, but at a minimum it means that the components can readily exchange only information and not substantial quantities of mass or energy.

[*Maier, 1998*]. A system-of-systems is a set of collaborative integrated systems that possess two additional properties: operational independence of the elements and managerial independence of the elements.

[*Manthorpe, 1996*]. In relation to joint war fighting, system of systems is concerned with interoperability and synergism of command, control, computers, communications, and information (C4I) and intelligence, surveillance, and reconnaissance (ISR) systems.

[*Northrop et al., 2006*]. A system of systems is a system comprising independent, self-contained systems that, taken as a whole, satisfy a specified need.

[*Office of the Under Secretary of Defense for Acquisition, 2004*]. A system of systems is a set or arrangement of interdependent systems that are related or connect to provide a given capability. They are also a set or arrangement of independent (not interdependent) systems that can be arranged or interconnected in various ways to provide different capabilities.

[*Pei, 2000*]. System of systems integration is a method to pursue development, integration, interoperability, and optimization of systems to enhance performance in future battlefield scenarios.

[*Purdue, 2005*]. A system of systems does not have all of these traits, but it will clearly exhibit a majority of them: operational independence of elements, managerial independence of elements, evolutionary development, emergent behavior, geographical distribution, inter-disciplinary, heterogeneity of systems, and systems of networks.

[*Rabelo, Bardina and Brown, 2003*]. One interesting characteristic of a complex system is that it is by default a system of systems.

[*Sage and Cuppan, 2001*]. Systems of systems exist when there is a presence of a majority of the following five characteristics: operational and managerial independence, geographic distribution, emergent behavior, and evolutionary development.

[*Sage, 2003*]. One of the major contemporary characteristics of large systems is that they are often formed from a variety of component systems: newly engineered from the “ground-up” custom systems, potentially tailored existing commercial-off-the-shelf (COTS) systems, and existing or legacy systems. There are a number of inherent characteristics of these systems, and such related terms as system of systems (SoS), federations of systems (FOS), or federated system of systems (F-SoS) or coalition of systems (COS), are often used to characterize them.

[*Sage and Biemer, 2007*]. A system of systems is a large-scale, complex system, involving a combination of technologies, humans, and organizations, and consisting of components which are systems themselves, achieving a unique end-state by providing synergistic capability from its component systems, and exhibiting a majority of the following characteristics: operational and managerial independence, geographic distribution, emergent behavior, evolutionary development, self-organization, and adaptation.

[*Saunders et al., 2005*]. A system of systems is defined as: the process of planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into a system-of-systems capability that is greater than the sum of the capabilities of the constituent parts. This process emphasizes the process of discovering, developing, and implementing standards that promote interoperability among systems developed via different sponsorship, management, and primary acquisition processes.

[Shenhar, 1994]. An array of systems (system of systems) is a large widespread collection or network of systems functioning together to achieve a common purpose.

[SOSECE, *System of Systems Engineering Center of Excellence*, <http://www.sosece.org>, 2003]. System of systems management is the process whereby a singular or distributed entity exercises authority for planning, organizing, staffing, controlling, and leading the combined efforts of participating/assigned civilian and military personnel and organizations, for the management of an integrated capability. System of systems management focuses on ensuring that: individual systems, acquired through individual programs, operated as autonomous components, satisfy capability needs within one or more systems or systems while providing appropriate functional capabilities to each of those systems of systems; the system of systems is planned, acquired, and operated to explicitly accommodate a wide range of ambiguous and changing conditions; and the composition of a particular system of systems can be reconfigured to form new system of systems implementations as conditions demand.

[SOSECE, *System of Systems Engineering Center of Excellence*, <http://www.sosece.org>, 2005]. System of systems engineering focuses on ensuring that: individual systems can operate as autonomous components within one or more systems of systems while providing appropriate functional capabilities to each of these systems of systems; can explicitly accommodate a wide range of ambiguous and changing conditions; and the composition of a particular system of systems can be reconfigured to form new system of systems implementations as conditions demand. System of systems engineering incorporates a dynamic mix of technical and non-technical factors, operational and business contexts, and enterprise enablers and constraints to satisfy system of systems capability needs.

[Stevens, 2005]. Mega-systems are defined as large scale, potentially complex systems that cross boundaries to provide capability beyond that achievable by their component elements.

[Stoudt, 2005]. Researchers have coined the term “system-of-systems” to describe the emergent behavior of new mega-systems created by the tight integration of previously distinct and independent systems. The emergent behavior of a system-of-systems results in a new capability that did not exist when the component systems were separate and distinct.

[United States Air Force Scientific Advisory Board, 2004]. A system will be called a system of systems when: the component systems achieve well-substantiated purposes in their own right even if detached from the overall system; the component systems are managed in large part for their own purposes rather than the purposes of the whole; it exhibits behaviors (including emergent ones) not achievable by the

component systems acting independently; and functions, behaviors and component systems may be added or removed during its use.

[*United States Air Force Scientific Advisory Board, 2006*]. A system of systems is a configuration of systems in which component systems can be added/removed during use; each provides useful services in its own right; and each is managed for those services. Yet, together they exhibit a synergistic, transcendent capability.

1.12. Bibliography

- [ALB 99] ALBERTS D., GARTSKA J., STEIN F., *Network Centric Warfare*, Command and Control Research Program (CCRP) Publications, Washington (DC), 1999.
- [ASH 58] ASHBY W.R., “Requisite variety and its implication for the control of complex systems”, *Cybernetica*, vol. 1, p. 83-99, 1958.
- [AUT 07] AUTRAN F., CATTAN D., GARNIER J.L., LUZEAUX D., PEYRICHON M., RUAAULT J.R., “Coupling component systems towards systems of systems”, *International Conference on Software and System Engineering and Applications*, Paris, 2007.
- [BER 50] VON BERTALANFFY L., *Théorie générale des systèmes*, translation Paris (1993), Dunod, Paris, 1950.
- [BOA 06] BOARDMAN J., SAUSER B., “System of systems – the meaning of”, *Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering*, Los Angeles, USA, 2006.
- [BOU 56] BOULDING K.E., “General systems theory – the skeleton of science”, *Management Science*, vol. 2, n° 3, 1956.
- [BRI 00] BRILL J.H., CHEVALLIER J., MERCHADOU J.L., *Manuel des meilleures pratiques pour le développement de systèmes spatiaux*, Cépaduès-éditions, Toulouse, 2000.
- [CAR 05] CARNEY D., ANDERSON W., PLACE P., *Topics in Interoperability: Concepts of Ownership and their Significance in System-of-systems*, Carnegie Mellon University Software Engineering Institute, CMU/SEI-2005-TN-046, 2005.
- [CAR 06] CARLOCK P., LANE J.A., “System of systems enterprise systems engineering, the enterprise architecture management framework, and system of systems cost estimation”, *21st International Forum on COCOMO and Systems/Software Cost Modeling*, Hendon, UK, 2006.
- [CON 70] CONANT R.C., ASHBY W.R., “Every good regulator of a system must be a model of that system”, *International Journal of Systems Science*, vol. 1, n° 2, p. 89-97, 1970.
- [COU 04] COUTURE M., “CapDEM TD – systems of systems: state of the art report”, *Contract Report W7701-3-2621*, DRDC Valcartier CR 2004-427, 2004.

- [DAG 04] DAGNINO G.B., “Complex systems as key drivers for the emergence of a resource and capability-based interorganizational network”, *E:CO Special Double Issue*, vol. 6, n° 1-2, 2004.
- [DON 02] DONNADIEU G., KARSKY M., *La systémique: penser et agir dans la complexité*, Editions Liaison, Rueil-Malmaison, 2002.
- [EDG 03] THE EDGE, “Enterprise modernization”, *The Edge – MITRE’s Advanced Technology Newsletter*, vol. 7, n° 2, 2003.
- [ETT 00] ETTIGHOFFER D., VAN BENEDEN P., *Mét@-organisations: les modèles d’entreprises créateurs de valeur*, Village Mondial, Paris, 2000.
- [FIS 06a] FISHER D.A., An emergent perspective on interoperation in systems of systems, *Technical Report CMU/SEI-2006-TR-003*, 2006.
- [FIS 06b] FISHMAN C., *The Wal-Mart Effect*, Penguin Books, New York, 2006.
- [FRI 06] FRIEDMAN T.L., *The World is Flat: a Brief History of the Twenty-first Century*, First updated and expanded edition, Farrar-Strauss-Giroux, New York, 2006.
- [HAR 07] HART S.L., *Capitalism at the Crossroads: Aligning Business, Earth and Humanity*, Wharton School Publishing, Upper Saddle River 2007.
- [HOO 04] HOOKS I., “Managing requirements for systems of systems”, *Crosstalk – The Journal of Defense Software Engineering*, August 2004.
- [INS 05] INSIGHT, “Capability engineering for systems of systems: a coalition perspective”, *Insight – A publication of the International Council on Systems Engineering*, vol. 8, n° 1, October 2005.
- [KAM 02] KAM L., LECINQ X., LUZEAUX D., CANTOT P., “ITCS: the technical M&S infrastructure for supporting the simulation-based acquisition process”, *NATO Modeling and Simulation Group Conference*, Paris, 2002.
- [KLI 01] KLIR G.J., “Facets of system science”, *IFSR International Series on Systems Science and Engineering*, second edition, vol. 15, 2001.
- [LAN 07] LANE J.A., VALERDI R., “Synthesizing SoS concepts for use in cost modeling”, *Systems Engineering*, vol. 10, n° 4, 2007.
- [LEM 77] LE MOIGNE J.L., *La théorie du système général*, PUF, Paris, 1977.
- [LES 74] LE SOURNE J., *Les systèmes du destin*, Dalloz Economie, Paris, 1974.
- [LUZ 98] LUZEAUX D., “Towards the engineering of complex systems”, *Journées Nîmes 1998 sur les systèmes complexes, systèmes intelligents et interfaces*, Nîmes, May 1998.
- [LUZ 02] LUZEAUX D., LODÉON P., “Simulation-based acquisition of the future air-land combat system”, *NATO Modeling and Simulation Group Conference*, Paris, 2002.
- [LUZ 03] LUZEAUX D., FINCK R., “Défense aérienne élargie: l’apport essentiel de la simulation”, *Revue de l’Armement*, June 2003.

- [LUZ 03a] LUZEAUX D., "Cost efficiency of simulation for complex system acquisition", *SPIE Aerosense'03, Conference on Simulation*, Orlando, USA, 2003.
- [LUZ 03b] LUZEAUX D., "La complémentarité simulation-essais pour l'acquisition de systèmes complexes", *Revue de l'Electricité et de l'Electronique*, vol. 6, June 2003.
- [LUZ 03c] LUZEAUX D., "Acquisition des systèmes complexes: le cercle vertueux simulation-essais", *Revue de l'Armement*, March 2003.
- [LUZ 04a] LUZEAUX D., "La bulle opérationnelle aéroterrestre et la démarche de simulation pour l'acquisition", *Revue de l'Electricité et de l'Electronique*, vol. 6-7, June-July 2004.
- [LUZ 04b] LUZEAUX D., "La place de l'homme dans les systèmes de systèmes", *Ergonomie et informatique avancée ERGO-IA'04*, Biarritz, 2004.
- [LUZ 04c] LUZEAUX D., SCHANNE P., "Le laboratoire technico-opérationnel: un nouvel outil pour l'étude du futur système de combat aéroterrestre", *Revue de l'Armement*, March 2004.
- [LUZ 06] LUZEAUX D., "La simulation comme outil d'aide à la décision pour l'ingénierie des systèmes complexes", *Revue des Nouvelles Technologies de l'Information*, RNTI-E-8, numéro sur les systèmes d'information pour l'aide à la décision en ingénierie système, Cépaduès Editions, June 2006.
- [LUZ 04d] LUZEAUX D., WARINGHEM E., "Le trio gagnant pour le futur combat aéroterrestre: simulation, battlelab, démonstrateur", *Revue de l'Armement*, March 2004.
- [MAI 96] MAIER M.W., "Architecting principles for systems-of-systems", *6th Annual International Symposium of the International Council of System Engineering*, Boston, 1996, available at: <http://www.infoed.com/Open/PAPERS/systems.htm>.
- [MAI 98] MAIER M.W., "Architecting principles for systems-of-systems", *Systems Engineering*, vol. 1, n° 4, p. 267-284, 1998.
- [MAR 05] MARIOTTI F., *Qui gouverne l'entreprise en réseau ?*, Presses de la Fondation Nationale des Sciences Politiques, Paris, 2005.
- [MEI 98] MEINADIER J.P., *Ingénierie et intégration des systèmes*, Hermès, Paris, 1998.
- [MEI 02] MEINADIER J.P., *Le métier d'intégration des systèmes*, Hermès, Paris, 2002.
- [MEN 04] MENARD C., *L'économie des organisations*, La Découverte, Paris, 2004.
- [MEY 06] MEYERS B.C., SMITH J.D., CAPELL P., PLACE P.R.H., *Requirements Management in a System-of-Systems Context*, Carnegie Mellon University Software Engineering Institute, rapport technique, CMU/SEI-2006-TN-015, 2006.
- [MOR 06] MORRIS E., PLACE P., SMITH D., *System-of-Systems Governance: New Patterns of Thought*, Carnegie Mellon University Software Engineering Institute, rapport technique, CMU/SEI-2006-TN-036, 2006.
- [MOR 77] MORIN E., *La méthode: la nature de la nature*, Le Seuil, Paris, 1977.
- [PAS 83] PASSET E., *L'économique et le vivant*, Payot, Paris, 1983.

- [PRI 06] PRINTZ J., MESDON B., *Ecosystème des projets informatiques: agilité et discipline*, Hermès, Paris, 2006.
- [RAP 06] RAPIN T., *La maîtrise d'ouvrage de Jean de France, duc de Berry (1340-1416)*, *Tabularia "Etudes"*, n° 6, 2006.
- [RAS 06] RASMUSSEN M.V., *The Risk Society at War: Terror, Technology and Strategy in the Twenty-first Century*, Cambridge University Press, Cambridge, 2006.
- [ROS 75] DE ROSNAY J., *Le microscope*, Le Seuil, Paris, 1975.
- [ROU 05] ROUSE W.B., "A theory of enterprise transformation", *Systems Engineering*, vol. 8, n° 4, 2005.
- [SAG 07] SAGE A.P., BIEMER S.M., "Processes for system family architecting, design, and integration", *IEEE Systems Journal*, vol. 1, n° 1, 2007.
- [SOS 06] *System of Systems Systems Engineering Guide: Considerations for Systems Engineering in a System of Systems Environment*, Systems and Software Engineering, Office of the Under Secretary of Defense for Acquisition and Technology, version .9, December 2006 (available at: <http://www.acq.osd.mil/sse/docs/SoS-SE-Guide.pdf>).
- [STU 05] STUTZKE R.D., *Estimating Software-Intensive Systems: projects, products, and processes*, SEI Series in Software Engineering, Addison-Wesley, Upper Saddle River, 2005.
- [THO 07] THORNTON R., *Asymmetric Warfare: Threat and Response in the Twenty-first Century*, Polity, Cambridge, 2007.
- [YAT 99] YATCHINOVSKY A., *L'approche systémique pour gérer l'incertitude et la complexité*, Collection Formation Permanente, ESF Editeur, Issy-les-Moulineaux, 1999.

Chapter 2

Emergence and Complexity of Systems of Systems

2.1. Introduction

The expression “system of systems” is, without a doubt, currently rather fashionable. Thus, 62,400 references could be found on the Internet in January 2006 compared to 14,000 a year before (January 2005), and by July 2007, there were no less than 402,000 references. A trend does not however hold the power to transform an expression into a concept, no matter how suggestive it may be. The concept should first have a meaning, and references. It should also be relevant, which means that, within a theory, it should help bring unity where only diversity could be perceived, or on the contrary, difference where there was only unity.

In an article published in 2001, A. Sage and C. Cuppan [SAG 01] defined a system of systems as a system displaying at least three of the following five characteristics: (1) operational independence of elements, (2) managerial independence of elements, (3) geographical distribution, (4) emergent behavior and (5) evolutionary development. The shape of this definition (three criteria among the five quoted above), wholly in keeping with the definition offered by M. Maier in “Architecting principles for systems of systems” [MAI 98], based on observation alone, is hesitant about the relevance of the selected criteria and does not provide any explanatory schema, something which makes it characteristic of stammering theorizing.

Indeed, what kind of community could have two systems of systems, the former only answering to criteria 1, 2 and 3 and the latter to criteria 3, 4 and 5? Incidentally, Sage and Cuppan, in the same article, stressed the absence of a consensual definition.

In this chapter, we offer to demonstrate how a “system of systems”, as we call it, constitutes a concept which we will contrast with the concept of “unitary technological system” within technological systems theory. We will place the relevance of the concept of “system of systems” within the scope of the formation, the ontogenesis of this category of technological systems under an alliance’s initiative (coalition or association). We will show which type of concrete referents this concept denotes. Finally, we will clarify this concept by developing a set of consequences concerning, on the one hand, these systems’ engineering, and on the other hand, their complexity.

This chapter is largely, but also freely, based on the works of the epistemologist Mario Augusto Bunge, and notably “*The Furniture of the World*” (ontology I) [BUN 77], “*A World of Systems*” (ontology II) [BUN 79] and “*Chasing Reality*” [BUN 06].

2.2. Matter and shape

The first distinction concerns concrete objects and abstract objects. In the direct tradition of Aristotelians, we call any piece of informed matter a *concrete object*, that is to say possessing properties (for example, the property of being located in space and time). On the contrary, an abstract object is a pure form, a fiction, a complex of formal properties. Thus, the number “2” is an abstract object, featuring the formal property of succeeding to the abstract object “1” within a numeral system.

Concrete objects are distinct from abstract objects not only through their materiality, but also through the type of properties supported by each category: material for the former, formal for the latter.

Thus, concrete objects carry properties which hold no meaning for abstract objects, such as “being localized in space”, “being animated with a certain speed” or “being alive”. These material properties $p(t)$, which characterize concrete objects, carry through time values which can vary or not. We then call a concrete object’s *state* the set of values of the various material properties which it features at a given time. Two concrete objects, precisely in the same state at the same time, would be identical. Conversely, two concrete objects are distinct through at least one of their properties.

The state of a concrete object can evolve through time, in that the values of some of its material properties can evolve through time; for example, a material object in movement is subject to variations of position and speed. We will call a change in a concrete object's state an *event*, and a series of events or a series of state changes, within an object a *process*. Thus, the combustion of a mixture of air and gas constitutes a process within the object shaped by the mixture; likewise, the series of actions an operator will perform on a machine constitutes a process within the object composed of the operator and the machine.

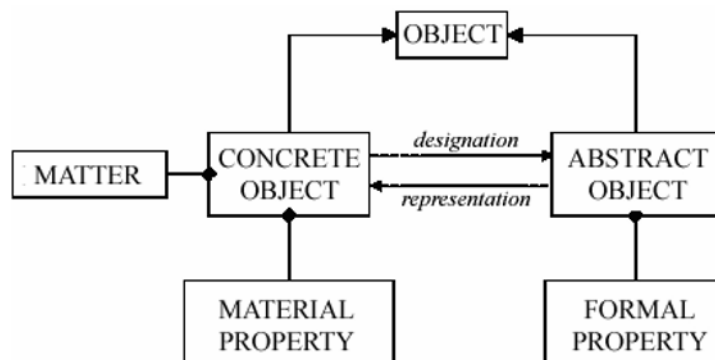


Figure 2.1. Concrete and abstract objects

We support the idea that these changes feature fixed qualities (obey laws), which may either be already known to us, such as with a certain number of cases related to physical sciences, or broadly unknown to us, such as with human behavior. We also state that some of these properties are *intrinsic*, insofar as they belong to the concrete object and to it alone, independently from the other objects in the world, while others have a *relational* character, which means they also depend on other objects. A concrete object's mass or electrical conduction belong among the intrinsic properties, while its weight, solubility or flammability belong among the relational properties. The statements we provide about the material laws governing changes within concrete objects tie some of their intrinsic and relational properties. Thus, an object's reaction (effect) to another object's action is linked to some of the former's intrinsic properties, such as in Coulomb's law. Another reading of this distinction between intrinsic and relational properties gives way to a concrete object's *actual* and *possible properties*. Possible properties only are only produced in situations within which these dispositions may express themselves. In that way, some synthesis are only obtained under certain conditions and in the presence of catalysts.

The *abstract objects* do not possess any of the properties we have just mentioned, but possess others, which link them to other abstract objects, such as “being an odd number” or “being a primary number”, which we will call formal properties and are characterized by their unchanging nature. Thus, an abstract object does not have a state and does not experience change. Abstract objects therefore do not exist in the same way as concrete objects.

However, abstract and concrete objects are not completely separate from one another, and there may be connections between them. A concrete object, such as a symbol drawn on a piece of paper, a byte in a memory card, an uttered word (acoustic waves), may reference an abstract object, a concept or a proposition. Conversely, a concept may reference a class of concrete objects and represent them within propositions. Thus, an Ada program, such as may be recorded in a file, is a concrete object which references an abstract object, the model of the behavior which will be sported by the object(s) which will carry this program in their memory card (as long as these objects’ resources are coherent with the model and the situation is favorable, which means the signals arriving to the objects’ terminals must be the ones that were expected). This connivance between abstract and concrete objects is one of the bases of modeling, one of the major tools in engineering.

2.3. Systems

When using the term *system*, we refer to an object represented by a triplet $\Sigma(t) = (C(t), E(t), S(t))$ where the composition C of Σ is a set of objects, the environment E of Σ is a set of objects, disjoint from C , and the structure S of Σ is a set of relationships between the components of C on the one hand – this is the internal structure S_{int} – and between the components of C and E for the external structure S_{ext} on the other hand ($S = S_{int} \cup S_{ext}$).

The composition $C(t)$ of a system $\Sigma(t)$ cannot be reduced under the line of two components, while its environment $E(t)$ is only empty by way of an exception (the universe is a system whose environment is empty, by construction). Moreover, if we look at the system’s internal structure $S_{int}(t)$, then a component closely related to any element c of $C(t)$ – for example the set of the components of $C(t)$ which are directly or indirectly in relation with c – cannot be isolated indefinitely from (without relation to) the rest of the composition $C(t)$.

Thus, for example, a satellite system, featuring a ground component – the ground control stations – and space components, may experience periods of times during which satellites are isolated from the ground component (beyond the scope of visibility); however, they cannot be permanently isolated (in which case the isolated system doesn’t belong in the system anymore and therefore is no longer controlled).

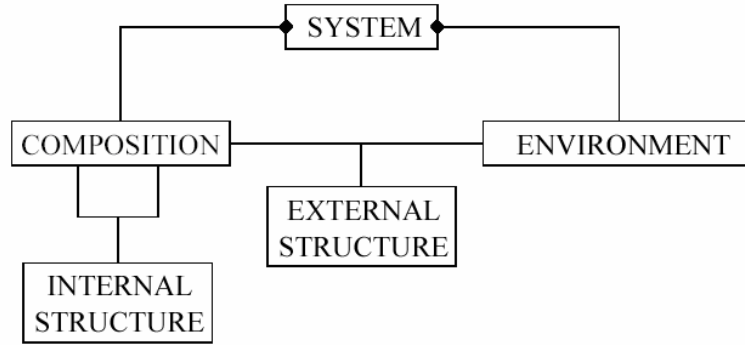


Figure 2.2. *General representation of a system*

The notation $\Sigma(t) = (C(t), E(t), S(t))$ focuses on the parameter t (time) to highlight the fact that the composition C , the environment E and the structure S of a system $\Sigma(t)$ are generally not fixed through time.

This notably means that the elements (whether organized or not) situated at a given time in the system's environment may be integrated within the system's composition, or, on the contrary, that the elements (whether organized or not) situated at a given time in the system's composition may be placed in the system's more or less remote environment.

Such is the case, for example, with the crew and passengers of an airplane who, during the flight, belong to the composition of the airplane system (by contributing, notably to the airplane's dynamic properties, such as its balance), and once off the plane, will enter the system's environment, or permanently separate from it.

2.3.1. Systems and subsystems

Let us consider two systems $\Sigma_1(t) = (C_1(t), E_1(t), S_1(t))$ and $\Sigma_2(t) = (C_2(t), E_2(t), S_2(t))$. We still say that $\Sigma_2(t)$ is a subsystem of $\Sigma_1(t)$ for a period Δt if during this period all the objects that compose $\Sigma_2(t)$ belong in the composition of $\Sigma_1(t)$ ($C_2(t) \subset C_1(t)$), the ones forming the environment of $\Sigma_1(t)$ belong to the environment of $\Sigma_2(t)$ ($E_2(t) \supset E_1(t)$), and finally if the relations linking the components of $C_2(t)$ between them and the relations linking the components of $C_2(t)$ to the components of $E_2(t)$ are also relationships $S_1(t)$ ($S_2(t) \subset S_1(t)$).

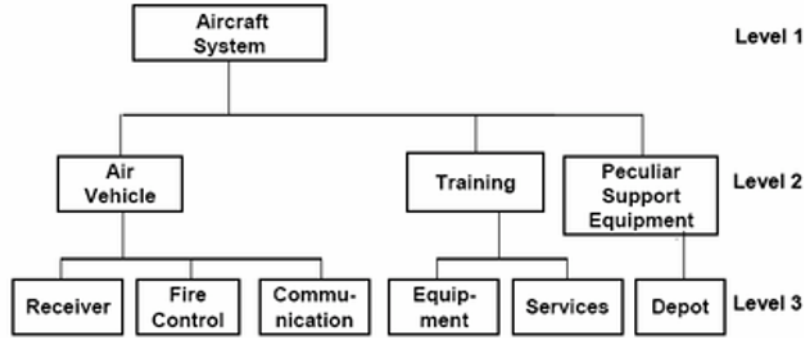


Figure 2.3. *System Breakdown Structure (from Wikipedia)*

This definition of the subsystem notion allows us to introduce a strict definition of the *System Breakdown Structure*, also called “system structure”. A breakdown of $\Sigma(t)$ into first rate subsystems is a family $\{\Sigma_i(t)\}_{1 \leq i \leq n}$ if the objects $\Sigma_i(t)$ are subsystems of $\Sigma(t)$ and if the family $\{C_i(t)\}_{1 \leq i \leq n}$ is a partition of $C(t)$, meaning that for all i, j , we have $C_i(t) \cap C_j(t) = \emptyset$ and if $\bigcup_{1 \leq i \leq n} C_i(t) = C(t)$. A breakdown of the different components of the family $\{\Sigma_i(t)\}_{1 \leq i \leq n}$ into first rate subsystems constitutes a breakdown of $\Sigma(t)$ into second rate subsystems.

In this way, step by step, a breakdown structure can be defined. These breakdowns are not unique; a system can be broken down in several ways: for example, breakdown into organs, premises or line-replaceable units (LRU).

2.3.2. Resulting and emergent properties of a system

If we design by P_Σ all of Σ ’s properties, we will say that a property P of Σ is *resulting* if there exists a subsystem of Σ which features P . On the other hand, if P is a property of Σ which none of Σ ’s subsystems already feature, we will say that P is an *emergent* property of Σ .

Thus, a system’s mass, such as an airplane’s, results from the mass of its parts, while the ability to transport, from one point of the globe to another through air, a useful charge within delay, security and cost prerequisites, is an emergent property of the airplane system which none of its parts possesses on its own, even if each part contributes to this property.

We can therefore split the set P_{Σ} of Σ 's properties into two parts: on the one hand, Σ 's resulting properties, and on the other, Σ 's emergent properties. Our postulate is that, for a system Σ , the set of Σ 's emergent properties is never empty.

In other terms, *the presence of emergent properties is a distinctive feature of systems*. Every system features at least one emergent property, no matter its “complexity”. From where we stand, emergent properties do not automatically entail complexity. Moreover, characterizing systems of systems as objects featuring, among other things, emergent behaviors, as done by M. Maier, does not seem to be of much relevance either, insofar as this criterion holds no discriminating power.

2.3.3. *Natural and artificial systems*

The second line of separation we will introduce is the one which distinguishes *artificial systems* from *natural systems*. This is a genetic distinction which underlines the human origin of the former, while the latter belong to an evolutionary process which goes from the nuclear synthesis of atomic physical systems in galaxies and stars, to the formation of social systems composed of biological systems equipped with a central nervous system, with no possibility of fixing the limits of this evolutionary process.

We are hypothesizing that natural systems do not answer any purpose, namely an intention, and that they do not fulfill any function, namely a desired effect (according to standard EN 1325-1 [EN 96]). However, this hypothesis goes hand in hand with the claim that those systems' evolution obeys, from their formation to their disappearance, material laws (physical-chemical, biological, psychic and social, depending on the system's level), which are influenced by the environmental conditions. This ontological *parti pris* doesn't contradict the fact that many of these evolutionary processes may escape, in part or wholly, our understanding.

On the other hand, artificial systems are produced by human intentions and therefore feature a purpose, following the aforementioned meaning. Unlike natural systems, artificial systems, during part or all of their lives, fulfill functions, namely effects sought after by human groups, for the benefit of the people operating them. We will later define with more precision the purpose of a specific type of artificial systems, namely the technological systems.

2.3.4. *Abstract and concrete artificial systems*

The third line of separation we will introduce is the one which distinguishes between *abstract artificial systems* and *concrete artificial systems*. This is an

ontological distinction, which considers the former as pure forms, fictions, and the latter as material entities.

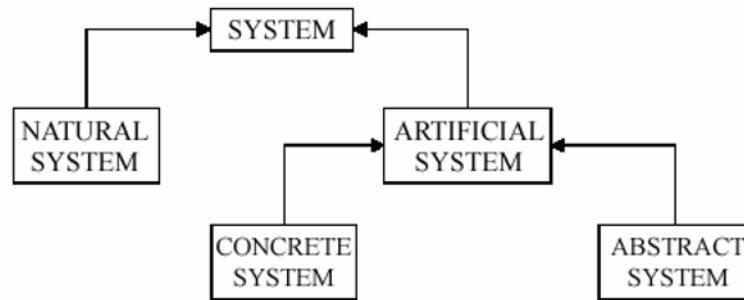


Figure 2.4. *System typology*

Abstract artificial systems belong within our beliefs and bodies of knowledge. They include both the myths and the scientific theories which human groups have designed to answer their own questions. They also include the languages shared by these groups.

In particular, scientific theories are $\Sigma = (C, E, S)$ systems whose composition C regroups a set of concepts and whose environment may group abstract and concrete objects. These concepts have formal relationships, both with themselves and with the environment's objects, such as the relation of representation (the concept A represents concrete objects C).

In this way, probability theory is a mathematical theory whose composition C includes concepts such as universe, event, random variable, probability. In its environment E can be found, among others, first-order logic, set theory, and measure theory. The structure S is then made of its axioms (Kolmogorov axioms) and all the consequences which might be deduced from them (theorems).

Likewise, kinetic theory is a physical theory whose composition C includes concepts such as gas, molecules, elastic collisions and electromagnetic interaction, temperature, pressure, viscosity, diffusion, thermal conduction. In its environment E can be found other physical theories such as classic thermodynamics, elastic collision theory (mechanics) and electromagnetism theory, and also mathematical theories such as probability theory. The structure S is then made of its hypothesis (for example, the volume of molecules is negligible, the molecules follow a Brownian motion, etc.) and of all the consequences which might be deduced from

them (theorems linking the gas temperature to the molecules' kinetic energy, the gas pressure to the collisions of molecules with the wall, etc.).

The fundamental difference between a mathematical theory, such as probability theory, and a factual theory (about concrete facts), such as kinetic theory, is that the former does not feature any concrete object in its environment, while the latter does. Thus, a factual theory can be formally true but factually false, while mathematical theory is only concerned by formal truth. Moreover, factual truth, also called verisimilitude (named so by K. Popper) to which factual theories are referring, is only a partial truth, approximate (Newton's theory is only true up to a point), unlike formal truth. However, factual theories, such as kinetic theory, are, at their basis, models which give access to concrete systems, whether they are natural or artificial.

2.3.5. *Technological systems*

Like all concrete systems, a *technological system* can be represented by a triplet $\Sigma(t) = (C(t), E(t), S(t))$ in which the composition C groups humans and artifacts, the environment E of Σ is constituted of humans, natural and artificial concrete objects, and where the structure S of Σ is a set of relationships between elements from C and from E . Other delimiting characteristics will be introduced shortly.

2.3.5.1. *Composition and environment of a technological system*

The composition $C(t)$ of a technological system $\Sigma(t)$ groups humans $H(t)$ and artifacts or products $P(t)$. This composition can vary through time. Let us consider, for example, a technological system such as an aircraft. During its design, it does not exist but in the imagination of its designers; only its design system exists, notably composed of its designers $H(t)$ and a set of support products $P(t)$ which concern, on the one hand, its definition (specifications, maps, models, study simulations, maquettes and prototypes) and on the other hand the design and simulation tools. During the production stage, new agents and new products appear: the technological system's end products. During flight, an aircraft's composition will include, on the one hand, the staff, the passengers, the freight, and on the other hand, the aircraft's product components: cell, cabin, cockpit, wings, engines, fuel, fluids.

The individuation principle we have adopted (two concrete objects in the absolute same state are identical) brings us to consider that, on top of the technological system design models, a family of distinct technological systems shares the same definition. Each of them follows, from the first metal panel or the first formed component, a specific life cycle.

Moreover, in this definition of technological systems, humans are "in the loop". They constitute one of the two components of technological systems, alongside or

rather above the products. In the design of a technological system, *the artifacts are tools used by humans to modify the state of the environment*. We are upholding a vision in which the technological systems, instruments at the service of human desires (whether rational or irrational) further the immediate intervention capabilities of human groups. A technological system should be designed more precisely as a *social-technological system*.

The environment $E(t)$ of a technological system $\Sigma(t)$ groups humans $H(t)$ and products $P(t)$, but also natural objects $N(t)$. This environment may also vary through time.

In this way, during flight, an aircraft's environment will include, among other things, the air traffic controllers which operate the system of the same name, other aircrafts, the air mass and earth surface.

2.3.5.2. Structure of a technological system

The *structure* $S(t)$ of a technological system $\Sigma(t)$ is constituted by the set of material relationships between each component of Σ , as well as the set of material relationships which the composition's components have with the environment's components. This structure varies through time and includes a multiplicity of interactions between the composition's operators, the operators and the environment's objects, the operators and the composition's products, the composition's products and the environment's operators and objects. These interactions are organized into a process, which not only modifies the state of the system's composition, but also the state of its environment.

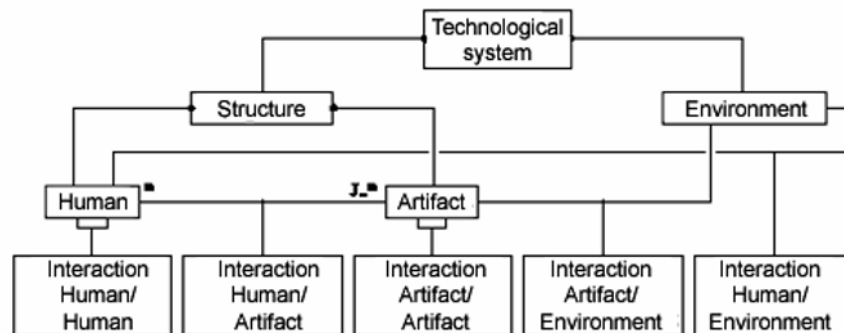


Figure 2.5. Technological system

A technological system, unlike a work of art or a body of scientific knowledge, does not contain its own purpose, but finds its *raison d'être* outside of itself. Indeed, *the purpose of a technological system is always to modify part of the state of the*

world, that is to say to modify its environment. Such a characteristic is not specific to technological systems, and has been a feature of technical systems since the birth of mankind. Technological systems are, however, distinct from technical systems through their formation model. The design of technical systems can be completely empirical, such as the creation of the very first tools by proto humans. As underlined by Norbert Roozenburg [ROO 95], among others, the operational model (projected or realized), embodied in a technological system, stems from scientific theory. A technological system is a realization of technosciences.

2.4. Genesis of concrete systems

In this paragraph, we will look at the way concrete systems, and in particular technological systems, develop, evolve, and finally become obsolete.

2.4.1. Genesis of natural systems

Keeping in mind that exchanges between the composition and the environment of a system do exist, in a continuous and/or discreet fashion, we affirm that *natural systems emerge from the synthesis of pre-existing concrete elements*. This synthesis of pre-existing elements is materialized through their interaction, which links the parts into a whole. This formation of a natural system from pre-existing elements coincides with the emergence of specific properties of the newly formed system, and the possible disappearance of properties which were present in the pre-existing elements. Such is the case with the chemical activity of reactants in a chemical reaction.

Examples of this synthesis mechanism are plentiful on every level. It is possible to quote, among others, the nuclear synthesis mechanism within stars, which creates helium, carbon and iron nucleus from hydrogen nucleus, depending on the conditions within the stars. These heavy nucleuses, created within stars, during their entire life cycle, will then be disseminated in the interstellar medium upon transforming into a supernova. It is also possible to quote the molecular synthesis processes which translate into the rupture of some chemical links between atoms, and the construction of new chemical links, such as polymers formed from carbohydrates. We might still quote biosynthesis processes, such as the synthesis of proteins within a cell.

This formation of natural systems can happen in one or several steps; among those steps may be featured the breakdown of a previously synthesized super-system. Such is the case with the different forms of cellular division, where a cell

synthesizes its “double” before parting with it (mitosis), or splits itself into two cells possessing different genomes (meiosis).

This formation of natural systems happens as soon as the formation process is truly possible, and each and every development condition of this process is present ($\{\text{truly possible element}\} + \{\text{fully gathered actualization conditions}\} \Rightarrow \{\text{event realization}\}$), either in a definite manner, or with a certain probability.

As for natural systems, two levels of synthesis can be distinguished: the self-assembly of pre-existing systems into a supersystem, and the self-organization of a system, whose constitutive system and subsystems are created within a unique process.

2.4.2. *Genesis of technological systems*

Technological systems are also constituted by the synthesis of pre-existing concrete elements. This synthesis process is a concrete process of fabrication, assembly and integration of constitutive elements and system implementation. This concrete synthesis is, therefore, similar to that of natural systems.

However, unlike natural systems, *technological systems are intentionally synthesized by human agents* (some animals possess this aptitude, to a lesser degree). Their purpose is to reach the objective set by their designers. Therefore, they are expressly developed to fulfill, during part of their life, functions, namely desired effects, for the benefit of their users. This is both their purpose (*raison d'être*), and the thing that can give them value (both operational and commercial).

This concrete synthesis of a technological system, performed by some of the system's agents, is preceded by an abstract synthesis, performed by some of the conception system's agents. This abstract synthesis process concerns (1) the goal to reach, (2) the means to reach that goal, meaning the system to develop, (3) the system's production process. This abstract synthesis results in an abstract model of the system to develop, model from which it can generally be duplicated if necessary.

Let us recall that *the purpose of a technological system is to modify the state of the outside world*, or more precisely, to modify, in the intended way, the trajectory of its environment within its state space. This intervention of the system must bring its environment into a different state than the one it would have reached in the system's absence, the new state being deemed preferable to the former. These sought states match the intentions of the technological system's designers. The states that *are* reached by the environment can however differ from the results which were sought.

2.4.2.1. Unitary technological systems

We will shortly introduce a new distinction between technological systems which we will call *unitary systems* (Mark Maier uses the expression *monolithic system* to reference the same systems, but the term “monolithic” seemed unfortunate to us after decades of modular design), and technological systems which we will call *systems of systems*. To introduce this distinction, we will look at the life cycle of these two categories of technological systems.

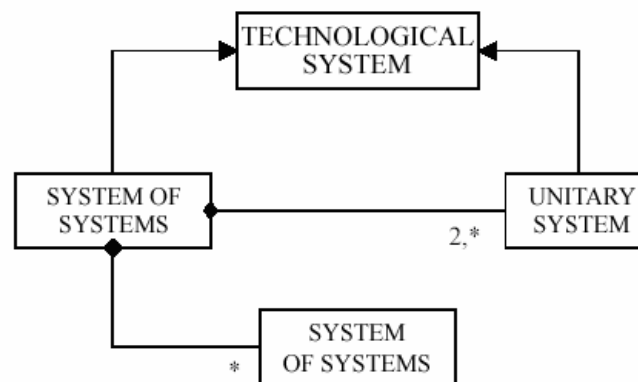


Figure 2.6. *Typology of technological systems*

A unitary technical system is a technological system whose purpose has been determined either by an acquirer, or by a supplier. For the acquirer, the goal is generally to improve one of his operational processes. For a supplier, it is about identifying an operational need which customers still demand, or prospects which might be fulfilled by a new offer. The definition of a unitary technological system originates either in an enterprise or a specific administration, or in a group of enterprises or specific administrations for the programs defined in cooperation. There is unity of definition. A unitary system fills a “personal” need; it is designed for physical or moral persons and the different copies of the system are subject to an appropriation by these persons.

In the same way, during its design and development, a unitary technical system is a technological system generally designed within a framework which stays on the scale of enterprises (societies or groups of societies), which can call on subcontractors or join a partnership network. In other terms, in the case of unitary technical systems, there is unity of design and integration, achieved by an architect and a set integrator while the achievements may be distributed.

The design of a unitary system aims to fill a previously determined operational need. This design aims to be global, moved by a common “design momentum”. On the one hand, a unitary system is designed as first-class subsystems, and on the other hand, those first-class subsystems are exclusively designed as contributors to the design of the system’s functions, even though some of them pre-exist and are being re-used.

Apart from being part of a unitary system, the first-class subsystems of a unitary system have no utility, no operational value. In this way, a subsystem such as an engine (car, plane) has no operational value unless it is integrated in the type of vehicle it has been designed for.

To sum up, unitary technological systems have, as such, an operational value, whereas the operational value of their first-class subsystems lies in their integration within a unitary system, and their operational value is therefore only relative. Once designed and developed, a unitary technological system is meant to be acquired either by the acquirer who has expressed that need, or by the customers targeted (or not) by the supplier. A unitary technological system’s vocation is to be the property of a person (physical or moral).

To conclude on unitary technological systems, we will mention that once designed and acquired, a unitary technological system’s vocation is to be exploited either by the acquirer or by a third party. This exploitation may occur within a very constraining framework, having to follow, for example, drastic maintenance cycles and strict statutory regulations. However, the operator enjoys autonomy of exploitation which allows him to act in the best of his interest.

2.4.2.2. *Systems of systems*

Like unitary systems, systems of systems belong in the definition of technological systems. However, systems of systems have several distinguishing characteristics, which we will try to explain.

Unlike a technical unitary system, whose purpose has been defined either by an acquirer or by a supplier, and which is acquired by an “individual”, systems of systems spawn from a certain number of agents’ common decision to cooperate by *assembling* the unitary systems owned by each member. The resulting system of systems has, therefore, no unique owner, nor even, in certain cases, any identifiable owner (to whom does the Internet belong?). We call *voluntary alliance* the understanding between agents who each own unitary systems and want to make them all interoperate, therefore presiding over the formation of a system of systems. Each member of the voluntary alliance hopes to derive a profit from their cooperation with the other stakeholders. To this purpose, they are ready to invest in

the endeavor, since they hope it will garner benefits higher than the ones they could garner on their own.

Therefore, the voluntary alliance brings about the formation of the system of systems. Unlike a unitary system, whose purpose is determined by an acquirer or a supplier, *a system of systems answers a field of specific purposes, the results of which are not fixed in stone*. Each member of the alliance behind the formation of a system of systems, or joining such an alliance, holds exclusive shares of the unitary systems which are part of the system of systems. Each agent will adhere to and support the voluntary alliance, as well as provide unitary systems for the creation of the system of systems, depending on the benefits he is hoping to garner, and *will* garner, from the common venture.

Before continuing with our study of systems of systems, let us quickly digress on voluntary alliances.

The concept of voluntary alliance developed from sociological theories which can be attributed to authors such as Caplow or Gamson, and has its origins in the game theory designed by Von Neumann and Mortgenstern.

Voluntary alliances group the *coalitions* and *associations* together. A coalition is a social system based on a temporary and voluntary instrumental social link between agents with different, but converging, goals, while associations are more durable alliances. Thus, an alliance of military forces forming for a period of time which is *a priori* delimited, and for limited objectives, takes on the shape of a coalition, while alliances such as NATO or the European Union are closer to associations. In time, a coalition can evolve into an association (durable voluntary link) or, on the contrary, dissolve. Associations can also dissolve, like the Warsaw Pact. However, there is no strict boundary between coalitions and associations, so that some authors indifferently use the term “coalitions” to talk about one or the other while keeping in sight the level of integration that can be reached by associations. In any case, the voluntary alliances (coalitions or associations) have, for each ally, an instrumental aim which converges with the aims of the other allies, without them being identical.

Sociologists interested in those socio-systems, such as V. Lemieux [LEM 98], distinguish three types of relationships which structure the voluntary alliances: transactions, affinities and controls. The transactions concern the benefits sought by the different allies and the contributions they will bring to that end. The affinities refer to the links of synergy, neutrality or hostility which can form between certain allies for cultural, historic, or even interpersonal reasons. The controls refer to the balance of strength that establishes itself first between the alliance and its competition, and secondly between the allies themselves. Voluntary alliances are rarely alliances between equals, and go hand in hand with a play on power.

When airline companies decide to coordinate the means of transportation they have at their disposal within an alliance (*SkyTeam*, *Oneworld*), these mergers obviously answer an instrumental intent (ensure its expansion and profitability, improve its planes' loading, etc.). Affinity/hostility relationships may contribute to shape the alliance, and shape the competition (alliances form around leaders in competition with one another), while leaderships are created within those very alliances (a pivotal State, a pivotal company, etc.).

However, the absence of a strong common purpose, which would favor personal interests, threatens constant bargaining over each person's contribution, and also over the distribution of the collectively generated benefits, during the alliance's entire lifecycle. In other terms, *their converging interests do not keep the allies from arguing about the distribution of profits and expenses*. The problem, a classic in socioeconomics, called the "stowaway", demonstrates the contradictions which can plague an alliance. A stowaway is an ally who benefits from the profits without paying the "right" price for his participation.

Let us now go back to the concept of system of systems, in order to identify the consequences of our genetic definition, consequences which correspond to the observable characteristics taken as criteria in M. Maier's definition.

We will say that a system of systems is distinct from a unitary system in that the first-class subsystems of a system of systems have, as such and as isolated units, a proven operational value. *The first-class subsystems of a system of systems are unitary systems*. These unitary systems constitute an important part of the contribution of the different allies to the alliance.

Outside of their inscription in a system of systems, the unitary subsystems keep all or part of the operational value they had before the formation of the system of systems, or independently from it.

Let us consider, for example, the public transport service system of major cities. It is a system composed of various means of transportation, such as trains, subways, busses, tramways, coaches, naval transportation, all interconnected, and to which we could add, why not, aerial means (helicopters). In such a system, each of the aforementioned means possesses its own operational value (transportation capacity) even if it is isolated from the rest of the system. The same thing cannot be said about a car engine isolated from the rest of the vehicle.

The unitary systems which compose a system of systems therefore enjoy an operational autonomy as well as a maintenance autonomy to which the first-class subsystems of a unitary system do not have access. These two properties, which are the two first criteria to be listed by M. Maier to distinguish systems of systems, are

therefore mere consequences of this type of technological systems' formation model. Even though the allies decide to make their unitary systems cooperate, they do not however relinquish the exploitation and maintenance of these systems, nor do they relinquish their ownership.

This operational and managerial autonomy of the subsystems of a system of systems opens the question of the subsystems' cooperation in such a system, when in fact the question has been answered (more or less successfully) in the field of unitary systems. This is the reason why *systems of systems are usually called cooperative systems*.

Moreover, at any given time, a system of systems is composed of unitary systems, some of which are active while others will be, for example, confined due to maintenance. Further still, the configuration of a system of systems is equally easily upgradeable. The unitary systems may join, or on the contrary separate from, the system of systems with a malleability which the first rate subsystems of unitary systems are devoid of. We find there the fifth criterion listed by Maier, also present in our genetic definition.

Unlike unitary systems, systems of systems are not designed within a common design momentum and a single organization. Chronologically speaking, the unitary systems which compose a system of systems are generally older than the system of systems which integrates them. *Systems of systems are designed as a process of assembly (sometimes progressive) of pre-existing unitary systems*. Such is the case with computer networks, and especially with the Internet, which could only truly take off after its large distribution through nominal partnerships such as PC (*Personal Computers*). In the same way, the public transport system of the Parisian suburbs emerges from the federation of a set of various transportation means subsystems, each autonomous, like the RATP's subway and bus system, the SNCF's suburban trains and RERs, the innumerable coach societies which interconnect with the aforementioned systems.

Likewise, the North American electrical energy production and transportation system is a remarkable example of a system of systems. As reported by the US-Canada report [USC 04] on the power blackout of the 14th August 2003, it is one of the biggest achievements of engineering in the last hundred years. Its value exceeds 1 trillion US dollars. It can produce 950,000 MW distributed over several thousand generators, transport them over more than 200,000 miles of transportation lines at over 230 KV, and supply more than 100,000 clients and 283,000 people through 3,500 retailers. The electric energy is produced by various generators, some belonging to retailers, others to independent producers, and others still to clients. The electric energy is then transported over long distances on many different lines. Transport lines are interconnected via electric stations and substations, which form

the transport network. The network's surveillance is performed by a supervisory authority, NREC (North American Electric Reliability Corporation), a non-government unit (implemented after a previous blackout) operating as a voluntary organization of the operators themselves, based on reciprocity, peer pressure and the interest of all who implement safety requirements.

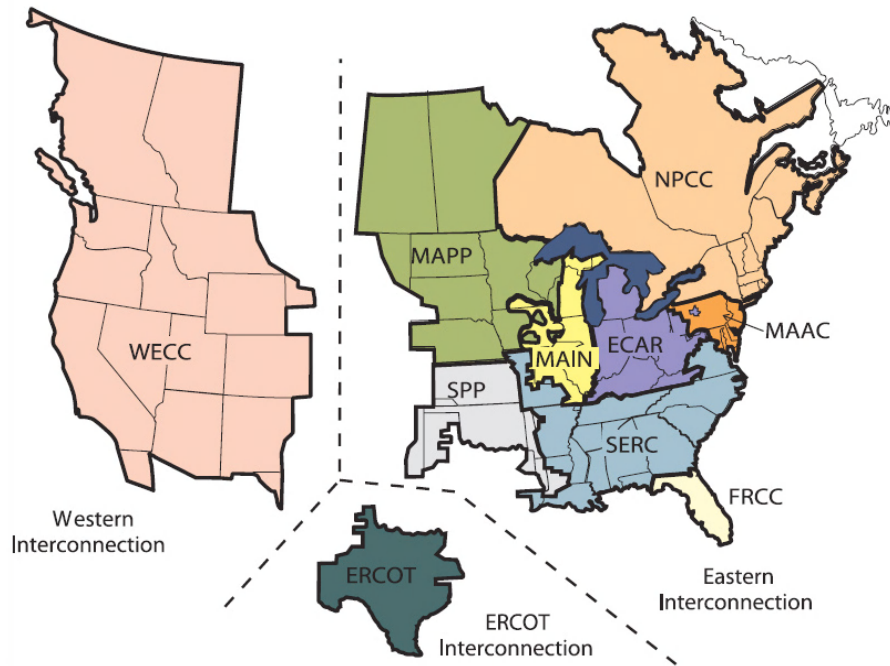


Figure 2.7. *The North American electricity production and transportation system*

To conclude our definition on a more formal note, a system of systems is a technological system $\Sigma(t) = (C(t), E(t), S(t))$ with characteristic features concerning its composition and structure, which we are about to detail.

The composition $C(t)$ of a system of systems includes a human component $H(t)$ and a product component $A(t)$ like in any technological system. However, the component $H(t)$ can be divided into a certain number of parts $(H_i(t))_{1 \leq i \leq n}$ each representing a separate organization $\overline{H_i(t)}$. The family $(\overline{H_i(t)})_{1 \leq i \leq n}$ constitutes an alliance (coalition or association) of the different organizations which have decided to make the unitary systems they own cooperate with each other. The product component $A(t)$ can also be divided into a certain number of parts $(A_i(t))_{1 \leq i \leq n}$. The couple $(H_i(t), A_i(t))$ then represents, for each i , the composition of a unitary

technological system, property of the organization $\overline{H_i(t)}$ and exploited and maintained by it (or for it) depending on its methods, its uses, its resources and its capacities. Moreover, the organization exploits and maintains the unitary system $(H_i(t), A_i(t))$ according to its own specific interests, interests which converge with the other allies' without, however, identifying with it.

The structure of a system of systems, just like all technological systems, features various types of interactions, including interactions (operator, operator), (operator, artifact) and (artifact, artifact). It should be pointed out that the interactions of the type (operator, operator) and (artifact, artifact) may concern operators belonging to different organizations and artifacts belonging to different unitary systems. These interactions go through particular interfaces of the system of systems which constitute cooperation points between unitary systems and through which transit data flows but also, contrary to what M. Maier claimed in the book quoted below, energy flows (such as with the interconnection of electricity production and transportation networks) and matter flows (such as with the interconnection of people or freight transportation networks). We will call *interoperability interfaces* these specific interfaces which delimit each unitary system within a system of systems, in order to distinguish them from the many internal interfaces of a system of systems. Interoperability interfaces are interfaces through which a system of systems' unitary systems come together. We will see that these interoperability interfaces hold a central position in systems of systems engineering.

2.5. Complexity of systems of systems

In this section, we set out to study the questions found in complex systems, and the complexity of systems of systems. References to the notion of complex systems and the engineering of complex systems are often found in technical literature. The term "complex" is rarely presented as problematic. For example, in a book about systems architecture, and notably about systems of systems architecture (*The art of systems architecting*), M. Maier and E. Rechtin [MAI 02] present this complexity like a clear cut notion, directly accessible through the common meaning (*composed of interconnected or interwoven parts* followed by the footnote: *A system need not be large or costly to be complex*). And yet, in the early 1990s, more than forty years after Von Neumann's founding works on complexity, the biologist Henri Atlan [ATL 91] was pointing out that the term of complexity was far from enjoying a univocal definition, and was more an intuition than a fully founded concept. Almost two decades later, this situation hasn't undergone radical changes, and we still do not have a firm and unique definition of the term "complex". The scientific literature on complexity is abundant. If its pertinence is not to be questioned, it still lacks unity and concerns highly specialized fields, within which are elaborated concepts of complexity which cannot *a priori* be merged together. The question is to determine

if the scientific concepts of complexity are pertinent in the field of technological systems and their engineering. Such is the analysis we will boldly attempt.

Two distinct postulations, which can be juxtaposed in common usage, can be used as evidence.

First, the longer the explanation, design, and manufacturing processes, the higher the number of steps, conditions, decisions, the more complex they are considered to be. The biologist Henri Atlan [ATL 86] had suggested the term *complication* to describe this situation. A process is all the more complicated that it includes a high number of steps, of conditions, of decisions. Thus, a system which hosts complicated processes is complicated. It should be remembered that the complication first concerns the processes, and that a system complicated to design can be simpler to operate and maintain. Therefore, we might think that a civil plane is more complicated to design today than fifty years ago, but that, on the other hand, it is less complicated to pilot.

Secondly, when we say that a process (with a purpose) is complex, we also mean to signify that the issue of that process is uncertain, and that it is not certain whether the process will reach the intended goal. For example, doing a task for the first time without prior training is already a complex process. This can justify, among other things, what has been said about complexity being the measure of our ignorance. The better trained a worker is in the execution of a task, the less complex this task is for him. Symmetrically, producing a set of pieces, answering to high requirements, with a poorly adapted tool can lead to a high degree of waste, and constitutes a task more complex than the one performed with perfectly adapted tools, insofar as the objective's achievement is less certain in the first case than it is in the second. In the first case, the uncertainty concerns our ability to define the means to an end, and in the second, our ability to reach an end with the given means. In that perspective, the complexity of the design or the production of a technological system pinpoints an uncertainty concerning, *a priori*, our ability to reach determined objectives of design or production.

As for natural systems, which have *a priori* no defined purpose, the complexity lies in the uncertainty about our ability to demonstrate, simulate or forecast the behavior of the systems with the means at our disposal: models, observation and measurement devices. From this angle, a system's complexity does not lie in the system itself but in the limits (ignorance, impotence) of our relationship to this system. Thus, the complexity of a chaotic system would lie in our inability to anticipate its evolution rather than in the evolution itself, despite it being determinant.

H. Atlan points out that complex intuition has been the subject of much theorizing [ATL 91] which, for the most part, gravitate around two poles: on the one hand the theory of algorithmic entropy also known as Kolmogorov complexity, and on the other hand a theory of complexity inspired by Shannon's information theory.

Brought back to its simplest expression, the theory of algorithmic entropy defines the complexity of a problem as the minimal calculating time a Turing machine needs to solve that problem, or as the minimum length of the string algorithms (using as sole instructions the instructions of a Turing machine) needed to solve this problem. This theory notably allows a classification of problems (class L, NL, P, NP, Co-NP, etc.) and highlights the degrees of complexity. This approach only concerns solved (and solvable) problems. Therefore we cannot, *a priori*, define the algorithmic entropy of a problem whose resolution is as uncertain as the one of an unsolved mathematic conjecture. Thus, Riemann's hypothesis lacks a defined algorithmic entropy, whereas Poincaré's conjecture, which was in the same situation before 2006, now features one (the length of the demonstration of Perelman's theorem provides a milestone of complexity). This theoretical basis, provided by theories on algorithmic entropy, can demonstrate what we have designated as the complication of a process. To be able to measure this property (complication) of processes, we must find the corresponding instrument and measuring unit (which would be, for technological systems, the equivalent of the instructions of a Turing machine and the number of instructions). Moreover, we believe that *this property is homogenous to an objective cost*, which is like saying that the complication of processes such as the ones of design, manufacturing or exploitation of a system are homogenous to the objective cost of these operations. This proposition is, incidentally, coherent with a subliminal message sometimes contained in the wording: "we are dealing with a complex system!"

The second attraction of complex intuition theorizing is provided by the entropy formula inscribed in Shannon's information theory, which measures the degree of uncertainty linked to the distribution of a random event ($H(x) = - \sum_i p(i) \log_2(p(i))$ if x is a random variable with n states x_i). The quantity of information associated with the occurrence of an event is, in Shannon's theory, by definition a monotonically increasing function of N/n where N is the number of possible cases and n the number of favorable cases. We then have $I = \log_2(N/n)$.

Nam Pyo Suh, in his theory of design (*Axiomatic Design*) [SUH 01] and his complexity theory [SUH 05], comes back to this notion of information quantity I which he thus interprets: for a given technological system, the quantity of information I_{FR} associated with a functional requirement FR corresponds to the probability that the system will meet the functional requirement FR : $I_{FR} = -\log_2(P_i)$. Now, if the system Σ must satisfy a set of functional requirements $\{FR_m\}$, then the quantity of information associated with this technological system equals

$I\Sigma = -\log_2(P_{\{m\}})$ where $P_{\{m\}}$ is the probability composed so that all the functional requirements FR_m are satisfied by Σ . Suh defines the complexity of a technological system (from its design to its manufacturing) as the quantity of information associated with this system. In other terms $C\Sigma = I\Sigma = -\log_2(P_{\{m\}})$. From that point on, the design or the manufacturing of a technological system are all the more complex since their probability of success is weak. That is to say, the lower the complexity of a technological system, the higher its chances of functioning correctly. From this data, Suh deduces the quality criteria of a technological system's design and manufacturing, which are of no interest in the present study. By furthering the analysis of his complexity concept, Suh isolates two components of a technological system's complexity, which he calls real complexity and imaginary complexity (possibly in keeping on with the metaphor of complex numbers). If $\{\Sigma\}$ is the set of the design and manufacturing models of technological systems Σ_M destined to answer the functional requirements $\{FR_m\}$ then he calls the real complexity of the system Σ_M the minimum of all the complexities of the various systems $C_R = \min\{C_{\Sigma_M}\}$ able to meet the requirements $\{FR_m\}$, and imaginary complexity C_I of the system Σ_M the difference $C_{\Sigma_M} - C_R$. In other words, the global complexity of a technological system is the sum of both its real and imaginary complexities: $C_{\Sigma_M} = C_R + C_I$.

Even though natural systems do not belong in his field of investigation, Suh applies his definition of complexity to them in a most interesting fashion: given a natural system Σ , if M is a knowledge model of Σ then Σ is an all the more complex system because the behavioral forecasting taken from the model M is not certain. Just like the imaginary complexity of a technological system C_I , the complexity of a natural system is the measure of our ignorance. Improving the model helps reduce the uncertainty concerning the natural system, and therefore reduce the complexity which is (abusively) attributed to it. The real complexity of a natural system would then correspond to an absolute and unmovable limit of our ability to model and simulate such a system.

The concepts of complication and complexity may be applied as-is to both unitary technological systems and systems of systems. But what about those systems' position on scales of complication and complexity? First, can we claim that a system of systems, producing the same effects as a unitary system, is less complicated than the latter? Then, can we claim that a system of systems, producing the same effects as a unitary system, is more complex than the latter? We deem the answer to be "yes" in both cases.

When operators, owning technological systems, form alliances and group their means of operation together, they hope to increase their own capabilities without having to assume the acquisition costs of the relevant means. In other words, people

seek the integration within a system of systems because it is supposed to achieve positive externalities. For each ally, the global cost of the unitary system, once inscribed within a system of systems, must be inferior to the cost of a (unitary) system which he would personally own and which would provide the same services as the system of systems. Should the opposite occur, the benefits of belonging to an alliance and connecting to a system of systems would be debatable. Moreover, it is empirically obvious that the interconnection of passenger and freight transport networks, electric networks or information networks such as the Internet, is generally much less complicated and costly than the acquisition and development of the networks and infrastructures themselves. People therefore participate in a system of systems *a priori* to achieve scaling effects with as little money and complications as possible.

On the other hand, the nominal complication of those systems of systems (compared with their unitary counterparts) goes along with a higher complexity, as has been defined above. This situation stems from the fact that it is more than improbable that all the requirements, which each ally attributes to the system of systems, will be met. Indeed, meeting the expectations of a co-ally does not only depend on the means he has personally deployed to this end, whether they be infrastructures, rules of conduct, maintenance procedures. On the contrary, he becomes tributary of the availability, or lack thereof, of means which escape his control because they belong to other allies, who follow their own rules of conduct to reach their own objectives, which are convergent, but not identical. Logically, this leads to a greater heteronomy in the conduct of a system of systems than in the conduct of unitary systems. Besides, the maintenance gamut may feature, within a system of systems, a great variability which may lead to a more heterogeneous reliability. The great blackouts of the North-American or West-European electric systems which happened in 2003 and 2004 are there to remind us that systems of systems may be at the heart of great cataclysmic breakdowns, only made more severe by the heteronomy and heterogenousness of these kind of systems of systems. Similar phenomena have been observed in banking and financial systems of systems.

2.6. Systems of systems engineering

If systems of systems distinguish themselves from unitary technological systems by their mode of formation, we should logically expect to find notable differences between unitary systems engineering and systems of systems engineering. Today, unitary systems engineering is well-known, standardized through several standards such as the IEEE1220 [IEE 05], the EIA 632 [EIA 03] and the ISO/IEC 15288:2002 [ISO 02] whose recommendations are largely convergent. A broad consensus on the subject has finally been reached within the community of systems engineers.

IEEE 1220 formulates with precision what (unitary) systems engineering consists of. This process links: “the interdisciplinary tasks that are required throughout a system’s life cycle to transform stakeholder needs, requirements, and constraints into a system solution.” According to ISO/IEC 15288:2002, it includes all of the following technical processes:

- stakeholder requirement definition process;
- requirement analysis process;
- architecture design process;
- implementation process;
- integration process;
- verification process;
- transition process;
- validation process;
- operation process;
- maintenance process;
- disposal process.

To this day, systems of systems engineering does not enjoy the same development and recognition. Several SEI reports underline the bias which might appear when we try to apply “traditional” systems engineering to a system of systems. The technical report CMU/SEI-2006-TN-015 [MEY 06] highlights many difficulties inherent to the development of requirement engineering in the context of a system of systems. Other standards bring up difficulties of “governance” [MOR 06] or “ownership” [CAR 05] within systems of systems. These statements are not surprising when you take into account the absence of a global standpoint presiding over the formation of a system of systems, formation which is a product of the juxtaposition of partial standpoints which see, in the alliance and the cooperation within that alliance, enough benefits to decide to participate in the formation of the system of systems.

Unlike the approach sketched out by ISO 15288’s list of technical processes, which pertains to global engineering, systems of systems engineering must be, first and foremost, an integration engineering, which enables, under certain conditions, the integration of unitary systems within a system of systems, and a cooperation engineering, which enables the concerted operation of each unitary system.

We may imagine that a stakeholder, owner of a unitary system and candidate to the formation of or integration into a system of systems, starts by designing an opportunity study to determine the benefits he might expect from a cooperation with other unitary systems. This activity is akin to a marketing plan, such as are elaborated upstream of unitary system development. Here, this activity does not lead to the elicitation and analysis of the requirements which must be met by the developed unitary system, but to a study on how to modify the unitary system so it can either participate in the formation of, or integrate a system of systems. This study concerns, first and foremost, the interfaces between each of the system of systems' unitary systems. This engineering therefore focuses on those interfaces, which we have defined as interoperability interfaces.

As has been clearly expressed by Maier, when he claims that a system of systems' architecture is composed of its interfaces ("the architecture of the system-of-systems is the interfaces. There is nothing else to architect."), interfaces are at the heart of systems of systems. To be sure, for systems to interconnect, their interfaces must be compatible on several levels, geometric, mechanic, electric, electromagnetic, protocol, in static as well as dynamic terms. Systems of systems may develop on an empirical interconnection basis, but the definition of adapted interconnection standards rapidly becomes a key element which will condition the development of a system of systems, and its complication (its cost).

Allowing a unitary system to participate in the formation of, or integrate a system of systems means rapidly giving it interfaces which comply with certain standards (http for the Internet, certain STANAG – NATO standardization agreement – for strength systems, etc.) and for this to happen, possibly run the system through some structuring works.

The design process of a system of systems therefore consists of designing interoperability interfaces, via or without standards, for one or several unitary systems.

However, the interoperability of unitary systems within a system of systems is not reduced to those systems' ability to connect with each other and transfer information, energy and/or matter flows through those interfaces. As underlined by the report CMU/SEI-2004-TR-004 [MOR 04], interoperability operates on several levels. For example, the LCI (layers of coalition interoperability) model introduced by Tolk [TOL 03] (Figure 2.8) insists on the two complementary aspects of interoperability: technical interoperability, and organizational interoperability.

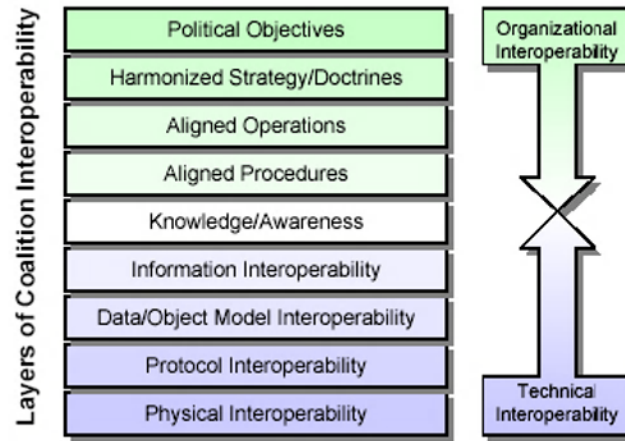


Figure 2.8. *Layers of coalition interoperability (LCI)*

Technical interoperability concerns the behavior of the various unitary systems' various artifacts, which interact through the interoperability interfaces. Organizational interoperability concerns the behavior of the various unitary systems' various operators, which interact either directly or indirectly through the artifacts whose conduct they ensure, and which are coupled through interoperability interfaces.

Systems of systems engineering must therefore include a process of design verification, which will ensure a unitary system's expected or faulty behavior will not lead to the failure of an artifact located in one of the others unitary systems it is cooperating with. In other words, this verification process must ensure that failures will not spread within a system of systems. For it to be so, a system of systems' interoperability interfaces must ensure its *robust partitioning*. Robust partitioning helps confine failures, if need be, but it might also prevent malevolent intrusions and interventions. This robustness feature of a system of systems' partition is not, *a priori*, acquired, and constitutes one of the objectives of systems of systems engineering.

We might also expect systems of systems engineering to contribute to the harmonization of the conduct and maintenance rules of the unitary systems which participate in a system of systems. This harmonization helps reduce the complexity of the system of systems, but we cannot expect such engineering to fully standardize those rules. For it to be so, the alliance at the head of the system of systems would have to transform into one single person (through a process of merger/acquisition,

for example), the convergent but distinct interests would have to merge, and the system of systems would have to evolve towards a unitary system. This evolution towards a unitary system is one possible fate of a system of systems.

2.7. Conclusion

As a conclusion to this chapter, we will recap our main proposals concerning systems of systems. The concept of system of systems is truly the concept of a theory of technological systems, theory which we have developed, and within which it distinguishes itself from unitary technological systems through its mode of formation: whereas unitary technological systems are created on the initiative of an acquirer or a supplier, who defines its expected characteristics, systems of systems are formed on the initiative of voluntary alliances (coalitions or associations) the stakeholders of which, each owning some unitary technological systems, decide to group the systems in order to exponentially increase their capabilities.

From this mode of formation results a set of properties specific to systems of systems, such as the operational and managerial autonomy of first-class subsystems. There exists, therefore, within systems of systems, a heterogenousness and heteronomy which have no equivalent in unitary technological systems. This heterogenousness and heteronomy stem, on the one hand, from the way the allies have convergent but distinct objectives, and on the other hand, each ally personally owns part of the system of systems which he operates depending on his personal capabilities and objectives. A system of systems organizes itself around interoperability interfaces through which its constitutive unitary systems link-up. These interoperability interfaces constitute the system of systems' architecture, following several possible topologies. From these proposals, we can deduce that a system of systems is both less complicated and more complex to design and operate than a unitary technological system offering the same potential.

Systems engineering, such as we know it through its great standards, ISO/IEC 15288:2002, EIA 632 and IEEE 1220, was elaborated with unitary technological systems in its line of sight. It therefore inherits an approach consistent with those systems' mode of formation, but which does not fully adjust to the systems of systems mode of formation. Current systems of systems engineering must therefore reevaluate all the processes in "traditional" systems engineering, with regards to the mode of formation of systems of systems, to focus on interoperability interfaces, technical as well as organizational, which form the true center of a system of systems and must ensure its reliable and robust partitioning.

2.8. Bibliography

- [ATL 86] ATLAN H., *Entre le cristal et la fumée*, Le Seuil, Paris, 1986.
- [ATL 91] ATLAN H., “L’intuition du complexe et ses théorisations”, in Fogelman Soulié F. and Milgram M. (ed.), *Les théories de la complexité: autour de l’œuvre d’Henri Atlan*, Le Seuil, Paris, 1991.
- [BUN 77] BUNGE M.A., “Treatise on basic philosophy”, *Ontology I: The Furniture of the World*, vol. 3, D. Reidel Publishing Company, Dordrecht, 1977.
- [BUN 79] BUNGE M.A., “Treatise on basic philosophy”, *Ontology II: A World of Systems*, vol. 4, D. Reidel Publishing Company, Dordrecht, 1979.
- [BUN 06] BUNGE M.A., *Chasing Reality: Strife Over Realism*, University of Toronto Press, Toronto, 2006.
- [CAR 05] CARNEY D., ANDERSON W., PLACE P., *Topics in Interoperability: Concepts of Ownership and their Significance in Systems of Systems*, CMU/SEI-2005-TN-046, 2005.
- [EIA 03] EIA 632-2003, *Processes for Engineering a System*, 2003.
- [EN 96] EN-1325, *Vocabulaire du Management par la Valeur, de l’Analyse de la Valeur et de l’Analyse Fonctionnelle*, 1996.
- [IEE 05] IEEE 1220-2005, *IEEE Standard for Application and Management of the Systems Engineering Process*, 2005.
- [ISO 02] ISO/IEC 15288-2002, *Life-Cycle Management—System Life Cycle Processes*, October 2002.
- [LEM 98] LEMIEUX V., *Les coalitions, liens, transactions et contrôles*, PUF, Paris, 1998.
- [MAI 98] MAIER M.W., “Architecting principles for systems-of-systems”, *IncoSE Systems Engineering Journal*, 267-284, 1998.
- [MAI 02] MAIER M.W., RECHTIN E., *The Art of Systems Architecting*, CRC Press, Boca Raton, 2002.
- [MEY 06] MEYERS B.C., SMITH J.D., CAPELL P., PLACE P., *Requirements Management in a System of Systems Context: A Workshop*, CMU/SEI-2006-TN-015, March 2006.
- [MOR 04] MORRIS E., LEVINE L., MEYERS C., PLACE P., PLAKOSH D., *System of Systems Interoperability (SOSI)*, CMU/SEI-2004-TR-004, April 2004.
- [MOR 06] MORRIS E., PLACE P., SMITH D., *System-of-Systems Governance: New Patterns of Thought*, CMU/SEI-2006-TN-036, October 2006.
- [ROO 95] ROOZENBURG N.F.M., EEKELS J., *Product Design: Fundamentals and Methods*, Wiley, Chichester, 1995.
- [SAG 01] SAGE A.P., CUPPAN C.D., “On the systems engineering and management of systems of systems and federations of systems”, *Information, Knowledge, Systems Management*, vol. 2, p. 325-345, 2001.

- [SUH 01] SUH N.P., *Axiomatic Design, Advances and Applications*, Oxford University Press, Oxford, 2001.
- [SUH 05] SUH N.P., *Complexity: Theory and Applications*, Oxford University Press, Oxford, 2005.
- [TOL 03] TOLK A., “Beyond Technical Interoperability – Introducing a Reference Model for Measures of Merit for Coalition Interoperability”, *8th International Command and Control Research and Technology Symposium (ICCRTS)*, Washington, DC, 17-19 June, 2003.
- [USC 04] U.S.-CANADA POWER SYSTEM OUTAGE TASK FORCE, *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*, April 2004.

Chapter 3

Contractual Aspects of the Acquisition and Use of Systems of Systems

3.1. Introduction

The technical complexity of a system of systems is also manifest in the legal issues. A system of systems is composed of several goods and services, provided by various legal entities, and calls for coherence and unity. Combining various categories of regulations becomes essential. Thus, the system of systems' structure can include: material goods (tangible assets), studies and software (intangible assets), engineering, counseling and services of installation or implementation (services), public works (civil engineering).

A system of systems is developed by people of various nature (artificial entities or individual, related to public or private law, specialized, or not, in a technical field). The legal qualification of goods and people is an essential prelude to the choice of the contractual organization to implement so as to monitor, with the necessary and achievable flexibility, the system of systems' design, its development, its operation, its maintenance, as well as its evolution and end-of-life.

The scope of the regulations varies depending on the category of people the system is aimed at. For example, a transportation system will not be presented to an end-user, to whom precise information must be given, as waivers of responsibility do not apply to non-professionals; however, it will be presented to professionals, whom the law doesn't protect as much.

If the system of systems processes personal data, such as a ticket reservation system or a military database management system, directives on the protection of such data will be called upon. The system of systems also demands that the property regime relevant to each of its components and to the system in its entirety be properly thought out. The ownership of the equipment can be transferred, and assignments of intellectual property rights are indispensable.

A system of systems may include computer components. The data processing system itself is a collection of equipment and creations of the mind, which can be provided by various entities, such as manufacturers, SSII (IT consulting firms), consultants, business software publishers, hosts, IT infrastructure facilities managers, maintenance providers. Each agent will organize his supplies and services following his own methodology, offer his own set of guarantees, limit his liability according to his corporate strategy, require a method of payment adapted to his objectives and financial capacity.

Such a system can include subsystems of various natures, such as:

- equipment (servers, work stations, telecommunication equipment, desktop publishing tools, etc.);
- software (standard software and specific developments);
- databases;
- files resulting from data processing;
- technical documentation or instruction manuals;
- hosting, maintenance, database updating services, etc.;
- services for the integration of the various components and the adaptation to specific needs, including project management (direction, control and coordination).

Concerning systems of systems¹ used in France, developed for the public corporation, the constraints specific to French law and the French tongue², whose enforcement is mandatory, will have to be reckoned with. A great many number of regulations will have to be taken into account (law, regulations, principles of

1. Dominique Luzeaux, Part 1, Chapter 1, “Systems of Systems: From Concept to Effective/actual Development”.

2. Law of August 4, 1994 relative to the use of the French language:

– law 94-665 of 4/08/94 (*loi Toubon*) relative to the use of the French language, JORF 5/08/94;

– law 2004-575 of 21/06/2004 (LCEN), JORF 22/2004, section 14: every person established in France durably to exercise their activity, and every legal entity whose head offices are established in France, must use French in their e-commerce activity.

European Union law), and their combination and implementation will lead to the creation of precise clauses (government contract, outsourcing of a public service, partnership contract), while following the discretion of the courts and the Council of State. Moreover, we will have to take into account the specificities of the defense district.

The contractual set-up the most adapted to a system of systems will depend on its nature and its complexity. In the case of an order filled by a public corporation, the set-up will have to follow a government contract or a contract for the outsourcing of a public service. In some cases, a solution can be found in the implementation of partnership contracts between public and private agents, as long as the established framework is strictly respected.

This chapter aims to make people aware of the multitude of rights and obligations which necessarily come into play in the design, development, operation and maintenance of a system of systems, as well as the wide range of responsibilities which it entails. Systems of systems therefore constitute an integrated set of components of various nature (section 3.2), designed by people who can coherently combine their diversified skills to meet the expressed needs (section 3.3), with engagements to coordinate (section 3.4), while complying with the unavoidable regulations on ownership rights (section 3.5), according to the most appropriate legal set-up (section 3.6).

It should be noted that this chapter, originally written by a French author for a French edition, is based upon French laws. The United Kingdom and the United States context is different. In the latter, laws may even vary from state to state. However, the main issue we address here is that systems of systems engineering cannot ignore the legal dimension. The reader is invited to refer to their specific laws to find those laws' answers to these questions.

3.2. An integrated set of components of various natures

3.2.1. *Material components*

The material components can vary and, depending on their nature and their purpose, they can obey specific regulations. Such is the case, for example, with security equipments³ on which specific audits must be run. The terms of procurement of some equipment, such as spare parts, will depend on the contractual commitments taken by the manufacturer or the distributor towards their clients, or sometimes toward maintenance providers. The equipment's compatibility will

3. Labour Code, sections R233-83-2 to R233-89-4.

require precaution as to the choice of materials and the guarantee clauses associated with it. The general contractor's obligations towards the maintenance of the equipment will differ depending on whether he acquired them through a sale, a renting⁴ or a financial leasing⁵. Some equipment will require, before it can be used, authorizations or accreditations (electric tools, for example). It will have to meet all the standards concerning its quality⁶ and the respect of environmental regulations⁷.

3.2.2. *Software elements*

The nature of software is not unique. Some display standard functionalities identical for every client. Adjustments and customizations can be made through settings or specific developments. Developments are generally undertaken by someone other than the business software publisher, in most cases an IT consulting firm, as an integrator. Existing agreements between publishers and integrators limit possible interventions on the software to a number of undertakings. Since the software will evolve through time, and might exhibit faults which will be corrected during their useful life, it is essential to draw up maintenance contracts. The contract may be complex since, more often than not, the publisher doesn't open his sources to third parties, and only selected persons are entitled to perform maintenance services.

The software package's maintenance still has to be coordinated with the maintenance of the specific developments, since the publisher has no control over the latter.

More and more, systems of systems include parts taken from free software or software developed from free software. Despite giving service providers and clients more rights than with proprietary software, the obligation to release any

4. Civil code, section 1 719.

5. Law on leasing, n° 66-455 of 2/07/66, JORF 3/7/66, amended by decree n° 67-837 of 28/09/67 and decree n° 2000-1223 of 14/12/2000.

6. New Approach directives. Several directives exist, depending on the field of activity. The list is available at: <http://www.newapproach.org/Directives/DirectiveList.asp>.

Note the directive "electromagnetic compatibility" (CEM) 89/336/CEE of 03/05/89 (repealed by 2004/108/CE) and the directive "radio and telecommunications terminal equipment" 1999/5/CE of 09/03/99.

7. ROHS: decree n° 2005-829 of July 20, 2005 relative to the "composition of electrical and electronic equipments and the elimination of the issued waste", J.O n° 169 of July 22, 2005, p. 11 988, text n° 39.

development and improvement to the community using the free software⁸ becomes a constraint, especially in a field where confidentiality and secret prevail.

Since the software components are designed by different authors, compiling the various documentations will be all the more crucial, so as to provide the users with a single, up-to-date and coherent documentation. Whether they concern techniques, organization, installation, use, instruction, essential manipulations in case of faults, the regulations differ in language, coherence, completeness, and level of detail.

3.2.3. *The human factor*

The use of a system of systems is generally complex, and the general contractor might wish to be assisted by outside business enterprises. In their turn, the enterprises will hire physical persons to provide the services. These persons will, however, remain under the employ of their own company; they will work with the tools, methods and knowledge they'll have taken from or learned in the company, and will therefore not be considered to be under the orders of the general contractor.

Defining all the means and behaviors firsthand is imperative, in order to uphold the boundaries between employer/employee, and between customer/provider.

In the case of undeniable confusion between the teams of the general contractor and those of the provider, these two entities will be liable to sanctions, which can be enforced in the name of unreported employment⁹. If such behavior is damaging to the provider's employees, they will be able to launch an action for redress for illegal subcontracting¹⁰. Vigilance can also incite temp agencies to react against the violation of their monopoly¹¹.

8. See the free software licences. For example, the CeCILL licence released by the INRIA (*Institut National de Recherche Scientifique*, the national institute for information and automation research).

9. Labour code, section L. 125-3: such is the case for any profit-making operation with the exclusive purpose of workforce leasing, if it does not do so within the legal framework.

10. Labour Code, section L. 125-1: "all profit-making operation of workforce leasing which harms one employee, or eludes the enforcement of the law, the regulations, the work convention or the collective agreement, or bargaining, is forbidden".

11. Labour Code, section L. 124: The temporary work contractor is, as defined by section L. 124-1 of the Labour Code: "any individual or artificial entity whose activity is to temporarily supply clients with employees whom they will hire and pay according to a qualification formerly agreed upon." The text enacts that, subject to the provisions of section L. 125-3: "All temporary work activity exercised outside such an agency is illegal."

By way of example, sentences have been pronounced in cases where the sums paid by the customer were calculated not according to the services provided by the operating agents, but to the exact amount of work accomplished by employees which were considered to be rented¹². The client who, under the cover of pretend subleasing contracts, actually took part in illicit workforce leasing¹³, was sentenced. Illicit workforce leasing was attested by the way the provider's employees were put under the client's authority, the way the latter would define which tasks the employees should undertake, the way he himself provided the spare parts, the way the services' price was calculated according to the workforce's cost, and the way the service corporation did not implement any of its own techniques¹⁴.

Moreover, a system of systems could include databases featuring information directly or indirectly linked to individuals. All treatment of personal data is regulated as to the nature of the information which can or cannot be used (interdiction to use racial origins or opinions on politics, religion, labor union or philosophy), as to the formalities to satisfy before the implementation of treatment (declaration to the CNIL, or counseling), as to their transfer outside of the European Union, in order to protect fundamental rights¹⁵.

If people's representations are introduced in the system of systems, such as photographs, sketches, biometrics such as fingerprints, their use shall be strictly supervised¹⁶.

12. Court of Cassation, criminal division, 25th April 1989, report 1 989, p. 435.

13. Court of Cassation, criminal division, 25th April 1989, report 1 989, p. 437.

14. Court of Cassation, criminal division, 15th June 1984, report 19 894, p. 229.

15. Law 78-17 of January 6, 1978, amended by the law 2004-801 of August 6, 2004.

16. - Civil code, section 9: "Everyone is entitled to the respect of his or her privacy. Without prejudice to compensation for injury suffered, the court may prescribe any measures, such as sequestration, seizure and others, appropriate to prevent or put an end to an invasion of personal privacy; in case of emergency those measures may be provided for by interim order."

- Penal Code, section L., subsection 226-1: "A voluntary breach of someone's privacy, through any means, is punishable by a year in jail and a fine of 45,000 euros: 2° Through fixing, recording or transmitting, without said person's consent, the image of a person being in a private place.

When the aforementioned actions have been accomplished in plain view and knowledge of the interested parties without them declaring their opposition, while they were able to do so, their consent is implied."

- Penal Code, section L. 226-8: "Publishing, through any means, an editing of the speech or image of a person without their consent, if it is not apparent that it has been edited, or if it is not explicitly stated, is punishable by a year in prison and a fine of 15,000 euros."

When the aforementioned infraction is committed through written press, tv or radio, the particular provisions of the relevant laws can be applied to determine the responsible party."

All regulations concerning hygiene and safety shall be respected¹⁷. People will also be vigilant as to the credentials of the employed individuals¹⁸.

3.3. Combining people with diversified skills and their contributions

3.3.1. *Diversity of the agents*

The “material” components of systems of systems may be provided directly by the publishers, but also by the agents of a distribution network. The range of obligations will therefore vary depending on the actual knowledge of the customer’s needs and the requirements of the manufactured equipments. Standard and specific manufacturing will also have to be reflected on by the general contractor, who might seek cautionary advice from the providers chosen as assistants to the contracting.

The supplies, works and services may be provided through momentary partnerships, within which the tasks will be divided among the joint venturers¹⁹. Even if the members of such a group elect a spokesperson who will represent them to the general contractor, the coordination and coherence of the supplies, works and services will not be taken care of. Indeed, the spokesperson does not take care of the technical coordination of the services. Such a task must be entrusted to a specific person, for example to one of the joint venturers. The more components there are in a system, the more crucial this question will be.

Moreover, the successful tenderer, whether he is alone or corresponds to a group of joint venturers, will be able to contract out some parts of the works, services or supplies specific to the client. Even though the law clearly entrusts the responsibility of what is contracted out to the subcontracting entity, reality is often more complex. How should the subcontractor’s knowledge be monitored? How can we be assured of their solvency? Even if the subcontractors must be approved by the client, and the terms of their payment agreed on²⁰, their failure during the system’s development or operation can be very problematic: lack of durability, lack of information about the

17. Various provisions of the French labour code.

18. Labour Code, sections L. 143-3 and L. 320 (delivery of a payslip and declaration prior to employment). Section L. 324-9 of the Labour Code forbids “totally or partially concealed work, defined and exercised according to the provisions of section L. 324-10, it is also forbidden to knowingly, directly or through a third party, employ one who exercises a concealed occupation.”

19. There is no official definition of joint venturing. It results from an agreement between several business enterprises. However, it is implied in the books of general administrative clauses and in the forms made available by the Minefi (former French ministry of economy, finance and industry; www.minefi.gouv.fr).

20. Law n° 75-1334 of December 31, 1975 relative to subcontracting.

manufacturing, making reparations uncertain, complexity of a software's sources, making corrections difficult or even impossible, adjustments to the statutory changes, evolutions.

In the construction of any system of systems, it will be essential to anticipate its operation and its maintenance, even if all or part of these services are subject to outsourcing²¹.

There can be cases where several subsystems are designed under various project managements, which raises the unavoidable question of global project management.

The general contractor will not always become the system's owner, even though the system will be designed according to his own specifications. He will be able to use the system, in a way, through a leasing service, or a line service granting him units of operating time.

3.3.2. Project management

Stemming from a need of coherence, project management helps monitor the project of design and operation of the system of systems, coordinate the various agents, and monitor the project's progression in order to anticipate failures and remedy them as they come. But the challenge is to know how to build efficient project management of the components, or even of the entire system of systems²².

There is no unique, rigorous definition of project management (French notion of "*Maîtrise d'œuvre*"). And there is no exhaustive list of project management tasks. The scope of a project is further defined by each new contract. Once again, the notion is eminently variable from one contract to the next, dependent on the agreements reached between the parties. Likewise, in works of civil engineering, the architect does not automatically share management with a legal agent. Project

21. Through government contract, outsourcing of a public service or a contract of partnership.

22. - AFNOR standard n° NF P03-001: definitions, for project contracts subject to private law, of project management and general contracting. The general contractor is the entity in charge of defining his needs, ordering, taking delivery and paying. The project manager is the entity in charge of managing the project, coordinating the interventions, ensuring that everyone respects their obligations.

- The Minefi Guide on general contracting in computer sciences is available in French at: http://www10.finances.gouv.fr/fonds_documentaire/daj/guide/gpem/amo/amo.pdf

- General contracting: "*Maîtrise d'ouvrage, maîtrise d'œuvre dans les projets informatiques*" published by Lavoisier (Syntec).

management can be entrusted to an engineering office, a service provider, or to the general contractor himself²³.

In the event of a system failure, the project manager is the first responsible, since he is in charge of the surveillance of the supplies and of the services provided by the other agents involved on the project, as a “*chef d'orchestre*”²⁴ or an “*ensemblier cotraitant*”²⁵. The project manager is the client’s privileged interlocutor and therefore, legally, his only co-contractor, and he is therefore liable for any failure of the subcontractors to meet their contractual obligations²⁶.

The contractual definition of the project manager’s duties may include:

- the organization and animation of project monitoring committees;
- the writing of the reunions’ reports;
- regular verification that the partners respect the project’s rules of quality insurance, the performances agreed on in a service agreement, the management of apparent faults;
- alert in the event of significant straying away from the project;
- the verification of the client’s comprehension and fulfillment of his obligations, in the name of his obligation of collaboration, and the respect of the financial commitments.

23. Court of Cassation, commercial division, March 7, 2006: “concerning the coordinating obligation resting on the Atos company, which could only be interpreted as an obligation related to the means used (“*Obligation de moyens*”), [...] notes that the Atos company could not monitor the works of the Metaware company in real-time, nor intervene in their development, and that only the IRD company could have a global vision of all the agents, and declares that the latter acted thoughtlessly by breaking the contract that linked it to the Atos company, even though the delays could not be imputed to the company.”

24. “conductor”, following André Lucas’s expression, *Droit de l’Informatique et de l’Internet*, PUF, Paris, 2001.

25. Philippe Le Tourneau, *Théorie et Pratique des Contrats Informatiques*, p. 115, Dalloz, Paris, 2000.

26. Cassation, Commercial Division, June 4, 1991: “But, since after noting that the Progeci company, the project manager, was sole capable of helping the Parent company fulfil its obligations towards its own client, the court of appeal has declared that the Progeci company was responsible of the faults in the system it sold to the Parent company; having thus defined the nature of the contract binding the two companies as the range of obligations subscribed to by the Progeci company towards the Parent company, it has legally justified its decision.”

When a contract is deemed to be “turnkey”, the system’s supplier is charged with managing the project²⁷. The project manager’s obligations concern the system in its entirety²⁸.

Outsourcing the project management does not discharge the general contractor of his obligation to participate in the project, by helping define his needs²⁹, his organization and his objectives; by validating the reunions’ reports and the delivered components; by proceeding, in the contractually defined delays, to his own deliveries and receipts³⁰; by following the payment schedule or by providing evidence, through written documents, of his well-founded reasons for not paying for the works, supplies or services that are not in accordance with the order³¹.

3.3.3. *Competitive bidding*

The difficulty in choosing the proper participants also results from the public corporation’s obligation to open the contract to competitive bidding, in accordance with the regulations on government contracting, based on the corporation’s

27. Court of Cassation, Commercial Division, May 19, 1998: “the court of appeal [...] has been able to interpret the commitment to provide both project management and a “turnkey” installation subscribed to by the A and S company as implying not only the installation of the equipment in the various stores but also their connexion with the central site, the centralization of data being, according to the company, the essential purpose of the system.”

28. Court of Cassation, Commercial Division, May 19, 1998: “the A and S society could only be exempt from responsibility for the difficulties resulting from the evolution of the central computer’s operating system if it had been kept away from the decisions and information relative to the evolution, and unable to fulfil its project management mission.”

29. Court of Cassation, Commercial Division, May 11, 1999: “after noting that the Campocasso company did not draw up specifications, the judgement rules that the responsibility was theirs to define their needs and objectives, by clearly specifying the nature and the importance of the works they wished to realize, so as to allow the suppliers to determine their actual needs and offer them the appropriate supplies, packages and software.”

30. Court of Cassation, Commercial Division, 7th March 2006: “no important technical obstacle hindered the effective start of the operations on the 31st December 1998 and the Atos company fulfilled its technical obligations of result; having noted that the system developed by the Metaware company was only received by the IRD company on the 2nd of December 1998, hence four months late, and having noted that the IRD company itself was late in delivering its goods to the Atos company, the judgement then holds, without reversing the burden of proof, that the stated delay cannot be ascribed to the Atos company [...]”

31. Implementation of the legal principle of anticipatory breach (“*exception d’inexécution*”).

specifications³², without subjecting the specifications to substantial modifications either during the bidding process, or during the drawing of the contract.

When ordering a system of systems, a public corporation should therefore be extremely vigilant during the drawing of the technical clauses, but also during the drawing of the administrative clauses which, in fact, possess a critical legal importance. This notably concerns the description of the deliveries, in their nature and division, their schedule, methodology, the receipt's organization, the choice of guarantees, levels of responsibilities and transfers of ownership.

The rules of competitive bidding, which ensure the respect of the candidates' equality in front of a public order, considerably reduce the possibilities of negotiation.

In public law, it is essential to anticipate all and any purchase which, put together, constitute a functional unit or belong to the same operation, in order to determine the tendering procedure. Nowadays, we should favor allotment to single market³³. This provision enables the command, during the same competitive bidding procedure, of supplies and services, or even works, the realization of which might be entrusted to several separate successful bidders. This provision is highly favorable to the development of systems of systems.

Whether it concerns a government contract, a framework agreement, the outsourcing of a public service or a partnership contract, the public corporation is under the obligation to issue a public notice of competitive bidding, which is fundamental and must include all the characteristics of competitive bidding, including the criteria for candidate choice. We quickly reach the threshold where a notice in the Official Journal of the European Union is mandatory³⁴.

It should be noted that launching a project on the basis of a letter of intent is not valid in public law, and that the bidder will only be able to file for compensation if he can prove that the public corporation led him to believe that it would definitely win the bid³⁵.

32. For government contracts, the specifications include *a minima* the following documents: the tender document and its appendixes including the priced contract, the special administrative terms and conditions (CCAP, *cahier des clauses administratives particulières*) and the special technical terms and conditions (CCTP, *cahier des clauses techniques particulières*). There is no longer an obligation to refer to the general administrative terms and conditions.

33. Code of government contracting, decree n° 2006-975, August 1, 2006, circular of August 3, 2006,

34. For the thresholds, see the code of government contracting, order of June 6, 2005.

35. Council of State.

3.4. Commitments to coordinate

3.4.1. *Effective date and duration of contractual commitments*

The effective date of the contract is also the start of the contracting parties' reciprocal obligations. For government contracts, the effective date is the date called "notification"³⁶. No commitment can be taken nor any expense laid out by the public corporation with regard to an un-notified contract. This date corresponds to the receipt, by the successful bidder, of the contract duly signed and validated by all of the administration's monitoring bodies.

In private law, the contracting parties can agree on the contract's effective date, which can either be fixed, or correspond to the date of the contract's signature, either by the contracting parties, or by the last party if they do not all sign it at the same time. In that case, the shipping delays must be taken into account.

A system of systems combines a contract under public law signed by a public corporation, and contracts under private law, signed between joint venturers, between the contractor and his subcontractors and suppliers. Coordinating the duration of commitments can turn out to be extremely difficult, in particular the management of the receipt and guarantee expiration dates which apply to some of the system's components.

A contract cannot be signed for an unlimited period of time. However, its duration can be specified or unspecified. A contract can be signed for a fixed period of time. It can also be renewable. Two solutions exist: tacit renewal or express renewal. In the case of tacit renewal, the contracting parties must give themselves the possibility of terminating the contract. Generally, an advance notice is planned so the remaining parties can anticipate the consequences of that non-renewal. As a precaution, we might send this advance notice via registered letter with delivery confirmation. The tacit renewal of a government contract is illegal.

For express renewals, the contract is automatically terminated at the end of a period if one or all of the contracting parties, depending on their agreement, have not decided on its renewal. A government contract can only be renewed upon decision of the public corporation. There again, an advance notice and a notice of the decision should be sent via registered letter with delivery confirmation.

36. Code of government contracting, section 12.

3.4.2. *Delivery*

It is essential to clearly define the specific commitments of each party concerning the legal nature of the contract, namely selling, renting, leasing, services, etc., and to have a clear database of all the client's needs.

The description of what must be delivered can be broken down into stages, in some cases can be defined within order slips, in one mandatory and several conditional segments.

The components of the system of systems must also be defined via commitments on performance, completeness, compatibility, etc.

All terms of delivery ought to be specified within the contract, such as the form of the delivery, the number of copies of a study report, the delivery of software in an object file or sometimes in source code, the documentation, etc.

The terms of equipment transportation are specified, notably to monitor the transfer of risks. The risks pertaining to the delivered goods or to the service's results will be transferred to the client on the date agreed in the contract. Any damage or loss of the merchandise will be imputed to its guardian. In international contracts, it is common practice to refer to the incoterms (International Commercial Terms), codified by the International Chamber of Commerce, which determine the moment of the transfer of risk depending on the chosen transportation mode: by air, rail, road, or sea. There is no ban, as pointed out by the incoterms themselves, on using these terms within a contract drawn under French law. The coverage of risks calls for adapted insurance. The transfer of risks also influences the choice of the carrier and the closing of the contract of carriage. It also has an impact on the party who goes through customs and the party who pays the duties.

Partial and definitive deliveries can also have deadlines: deadlines for the delivery of supplies, services and works, deadlines for the delivery of documents, for the communication of information, deadlines for receipt, deadlines for reactivation, repair or correction. These deadlines belong within the contract's full duration. Service deliveries can be associated with quality engagements defined by the contracting parties within a service level agreement. Penalties for the deterioration of the service can be agreed on, so as to prompt the service provider to respect his commitments.

In government contracting, certain kinds of contracts include deadlines planned for in the texts.

If the deadlines are imperative, delays may be sanctioned liquidated damages called “*pénalités*”, or by the rescission of the contract.

The defined liquidated damages will however have to be enforced within reasonable limits. If they are too low, or too high, a judge called on to settle the dispute might deem the liquidated damages to be patently excessive or derisory, and accordingly lower or raise them.

In some contracts, the service provider or the supplier may be prompted to work faster or deliver higher quality by bonus clauses.

Likewise, the liquidated damages and bonuses shall not be patently excessive nor derisory.

Nothing prevents the buyer from planning several liquidated damages clauses depending on the nature of the delays. We should specify in the clause which part of the price it concerns, and the factors of delay (day, hour, week, month, etc.).

3.4.3. *Receipt*

Receipt, also called acceptance in the field of systems engineering, consists of controlling that the delivered equipment, or the provided service, or the work achieved, conform to the client’s needs as they have been expressed.

It is therefore extremely important for the contracting parties to draw documents referencing the definition of the client’s needs, the scope of services or the characteristics of the merchandise on which the service provider or the supplier has contractually agreed. The compliance monitoring will be more efficiently implemented based on these reference elements.

The contracting parties will define, as early as the charting of contractual documents, the terms regulating the inspections, tests, corrections of detected faults, malfunctions or defects, the classification of the faults and malfunctions or defects, the temporary implementation of solutions for their replacement or bypassing. These rules can be written in a receipt file, which will be updated and validated by the contracting parties during the contract’s performance.

The validation and the declaration of receipt must be done by the client; they are one of his principal obligations, along with the payment of the agreed price.

This stage is crucial, for the receipt frees the supplier or the service provider from most of his contractual obligations.

It will be much harder for a client to have his claims recognized after the declaration of receipt than it would have been before it.

Receipt should be accepted expressly and be authenticated by the signature of a written document often called a written statement³⁷.

3.4.4. Financial matters

3.4.4.1. The price

The price must either be determined or determinable. A sale is void if the price is not determined by the contracting parties at the time of its closing. In a service contract, the price can be completely fixed upon signature of the contract, but there is a possibility that its precise determination will vary depending on certain events which can only be controlled during the contract's performance: quantity, application for authorizations, level of performance, technical complexity, evolution of the client's needs, etc.

A price fixed in a definite fashion is generally called all-inclusive. Rates, called unit prices, can be implemented for certain services or works, notably when the deal is carried out through purchase orders.

3.4.4.2. Price variations

The price may vary through a common agreement between parties. Such a variation may result from the carrying out of an escalator clause founded on indexes that have been published and are representative of the trade of one or both parties and fixed by contract. The clauses can combine the application of one or more indexes.

A distinction must be made between the revision of the price which will occur during the performance, and the price discounting contract. The latter allows for the price to be reassessed between the date on which the service provider issues his offer and the effective date of the contract or the notification³⁸.

37. Court of Cassation, commercial division, January 22, 2006: "[...] considering the ordered equipment had been delivered and accepted and gave entire satisfaction to the Imrep company, from which results that the Dai company had fully met its obligation of delivery and was therefore legally justified in its request for the buyer to provide proof concerning other failures of the seller."

38. In government contracts, the discounting is applied to offers issued more than three months before the notification of award.

In a government contract, an escalator clause must always include a fixed part³⁹. In private law, this is left to the contracting parties' discretion.

3.4.4.3. *The payment schedule*

It is also important to combine the payment schedules of the various supplies, services and works which are part of the various components of the system of systems, whenever the general contracting of the system of systems is authenticated.

An advance payment can be made on the effective date of the contract, before its performance. In government contracts, it is called a lump-sum advance payment⁴⁰.

Deposits can be made by the client during the contract's performance, until the balance is paid.

In government contracts, the deposit and balance payments cannot be made in advance: these payments are made on the basis of "performed service".

In public law, the prime rate subcontractors benefit from the direct payment of their invoices by the public corporation⁴¹. In private law, the system of direct action⁴² applies. Payments to the various agents will have to be adjusted.

3.4.4.4. *The payment period*

The contracting parties are also free to determine the payment period within private law contracts. However, if a precise clause is not featured, payments will have to be made every thirty days, when private individuals receive their invoices. In government contracts, the period is of 45 days from the invoice's receipt.

3.4.4.5. *Late liquidated damages*

If the client cannot pay the due sums, without justification, late liquidated damages will be automatically enforced. The rate of these liquidated damages will have to be featured on the invoices.

39. In government contracts, the fixed part is equal to 0.125.

40. The lump-sum advance payment equals to 5% of the total price. The contractor can forfeit it.

41. Law n° 75-1334 of December 31, 1975 relative to subcontracting.

42. If a subcontractor is not paid by the main contractor, he will address to the latter a registered formal notice with delivery confirmation, and send a copy of the letter to the client. If thirty days later the main contractor still has not paid the subcontractor for the performed and accepted services, the client must pay the required sum to the subcontractor.

It is also preferable to have it featured in the contract's clauses. This rate must not be less than a certain amount resulting from a rate determined by the law.

3.4.4.6. *Expenses*

The contracting parties will also split expenses according to their agreement. Those expenses might arise from traveling, logistics, reprography, freight, etc.

3.4.4.7. *Taxes and fees*

Taxes and fees are exercised on commercial transactions and supplies, services or works performed for a public corporation.

Their nature, amount, terms of coverage and their declaration vary depending on the country, the nature of the goods and services, and their destination. If some of the system of systems' components come from a foreign country, the financial impact and the method of payment of these duties and these taxes will have to be taken into account.

3.4.4.8. *Discharge of responsibility through payment*

The basic principle is that payment equals acceptance. Hence a recurring payment equals the validation of the deliveries which it concerns. It can therefore be useful to point out that these payments will be made progressively but might later be challenged if a substantial nonconformity was detected. In public law, the aim is to deftly handle the interim payments and the final partial payments.

3.4.5. **Guarantees**

3.4.5.1. *Contractual guarantees*

After the receipt, or even in parallel with the running of certain tests, the supplier or the service provider can provide guarantees to the client. These are contractual guarantees whose contents, duration and terms of implementation are fixed by the contracting parties. Between professionals, there is no obligation to have any contractual guarantee⁴³.

These guarantees can take on many forms: correction of faults, dysfunctions or defects, replacement of faulty parts, standard replacement of the equipment, lending of equipment with similar characteristics, etc.

43. Since the European directives, a two-year guarantee applies to customers.

We should therefore be extremely clear on that subject in the contracts and deals, and mind the coherence of the guarantees over every component of the system of systems.

3.4.5.2. *Guarantee of hidden defects*

For the whole life of a corporeal good, subject to normal use, the legal guarantee of hidden defects allows us to get a refund or keep the good in exchange of an allowance. The burden of proving the hidden defect is left to the person invoking it⁴⁴.

In French law, this guarantee concerns the selling of goods and rental. It is unknown within service contracts.

It is however heavy to implement and requires a proof of the hidden defect and to take legal action in the two years following the discovery of the defect in order to represent one's rights⁴⁵.

3.4.5.3. *Defects liability and decennial guarantee*

These two guarantees only concern the works. The first means that the contractor must answer, for two years after the handover, for the perfect completion of the works⁴⁶. The second means that, for ten years after the handover, the contractor must repair any defect that might damage the works in such a way as to make it unfit for its destination⁴⁷.

3.4.5.4. *The hold harmless clause*

In the event of the violation of intellectual property rights, the client who sees a third party claim ownership over the works he is making use of might act against the person who gave him the rights to the works⁴⁸.

44. Cassation, Commercial Division, January 4, 2005: "having declared that the Vieules company could not bring the proof that the data processing system was afflicted with hidden defects, the court of appeal did not have to go through with the research invoked in the second branch."

45. Civil code, sections 1 641 and following.

46. Building regulations: defects liability guarantee: law n° 78-12 of 4/01/78, JORF of 5/01/78 p.188, also called "Spinetta law".

47. Building regulations: decennial guarantee, sections 1 792 and following, and section 2 270 of the Civil Code, "law Spinetta".

48. The basic principle of this guarantee was in fact designed for sales (section 1 625 of the Civil Code says that: "the guarantee the seller owes to the buyer has two objects: *the first is the peaceful possession of the sold good*; the second is this good's hidden or latent defects").

This is the case, for example, for maps, diagrams, reports, drawings, models, sketches, music, software, etc. In particular, this concerns the works used through the Internet.

The terms for the implementation of that guarantee can be strictly defined in the contract: deadline for the transfer of the write of summons, information necessary to the organization of the defense of interests, etc. The fees' coverage and their limits are also controlled, as well as the type of replacement the guarantor will be able to provide (modification of the software, delivery of software of substitution).

3.4.6. *The combination of limitations of liability*

When all, or part, of the contractual obligations are not respected, a party can act by calling on the responsibility of the faulty party. In a system of systems, complexity arises from the variety of the works, services, supplies and the complexity of searching for the cause of the faults and malfunctions. The variety of the agents also calls for an in-depth search for the persons behind the fault or the malfunction. The agents will play with the qualification of their commitments concerning the “*obligation de moyens*” and performance. But that distinction is also present in the burden of proof. The same obligation can be an “*obligation de moyens*” or “*obligation de résultat*” depending on the context, or the parties' explicit wishes, or simply depending on the coherence and wording of the contractual documents. The contractor bound to an “*obligation de résultat*” is presumed liable for any fault or malfunctioning reported by the client. In order to be held harmless, he must prove the client's fault (client who did not fulfill his obligation of collaboration, for example) or a third party's fault (a supplier outside of the project who has direct contractual relations with the client) or a case of *force majeure* (external to the parties, unforeseeable, irresistible). If the contractor is bound by an “*obligation de moyens*”, the client must provide the proof of the contractor's fault, the latter having presumably fulfilled his obligations according to the contract and good engineering practice.

As for the services, the provider's “*obligation de moyens*” increases with the random nature of their provision, and his “*obligation de résultat*” increases as they become more precise.

If the principle is that everyone can put the liability of another person into question, the liability must follow certain rules and correspond to actual damage.

Failing that, purely dilatory liability actions will be punished⁴⁹. Claiming a right does not mean one can pervert that claim⁵⁰.

In order to analyze and monitor these risks, the agents are authorized to limit their liability. Thus, clauses of limitation of, or even exemption of, liability can apply. The foundation of liability pertains to civil law. It distinguishes tort liability, which pertains to civil laws and therefore cannot be abridged by a contract⁵¹, from contractual liability⁵².

If it is true that by “natural follow-up of the contractual engagement, the debtor must take responsibility for the acts of people whom he chooses to call on in the performance of a contract, to assist him or substitute to him⁵³”, this foundation is still lacking an essential detail. The debtor’s contractual obligation, legally binding, is in fact only a preliminary condition of the contractual liability whose operative factor is precisely defined as the debtor’s failure to fulfill his obligations⁵⁴. In other words, and this observation explains the specificity of contractual liability⁵⁵, it is not the contractual debt which justifies the liability to compensation, but its non-

49. Code of Civil Procedure, sections on abusive or dilatory procedures:

- section L. 32-1 NCPC: for a dilatory or abusive action: fine of 3,000 euros, in addition to the eventual reparation of damages;
- section 559 NCPC: idem for a dilatory or abusive appeal;
- section 628 NCPC: for a dilatory or abusive appeal: civil fine of 3,000 euros and compensation to the respondent.

50. On the abuse of rights:

- emergence of the theory: as early as the beginning of the 20th century, it became necessary to uphold the social function of rights, and therefore not to grant full impunity to contractors. It is now admitted that the quasi-totality of rights are liable to abuse punishable by law. Therefore, there exists very few absolute rights (an example of those would be ownership);
- criterion of the abuse of rights: an abuse of rights is confirmed when the law is wielded with the intention of harming, for example the construction of works on one’s grounds for the sole purpose of harming one’s neighbor. But the jurisprudence now holds the abuse of rights in a broader way:
 - when the holder of the right shows condemnable levity in the exercising of said right (Cass. Com. 11/10/97). Examples: abusive procedures, right to media criticism (obligation to check the exactitude of their information);
 - in some fields, the principle is even reversed: a right can only be exercised for serious motives. Example: right to dismiss an employee.

51. Civil code, section 1 382: “Any act whatever of man, which causes damage to another, obliges the one by whose fault it occurred, to compensate it.”

52. Civil code, section 1 146 and following.

53. A Bénabent op. cit. 412-1.

54. Fascicle *Jurisclasseur* “Responsabilité civile et Assurances”, fasc. 171-10 or Civil Code, sections 1 146 to 1 155, fasc. 11-10 or *Notarial Répertoire*, “V° Responsabilité civile”, fasc. 171-10, n° 2.

55. A. Bénabent.

performance imputable to a fact which appeared within the debtor's sphere of authority.

It is however mentioned that no limitative clause "must affect any essential obligation featured in the contract"⁵⁶.

In any case, the limit, or the disclaimer of liability, will not apply in the event of a gross negligence⁵⁷.

The project manager's liability can be challenged on the grounds of neglect of his obligation of coordination. However, each of the suppliers, joint venturers, subcontractors, remains liable for their own failures⁵⁸.

In the event where a contracting company entrusts another with the performance of the services in its name and for its benefit, the company taking care of the

56. Court of Cassation, commercial division, October 22, 1996: "[...] resulting from failure to meet an essential obligation [... of the contract...], the contract's clause of limitation of liability, which contradicted the scope of the commitment taken, had to be considered non written."

57. "Considering that the manifest intention of the contracting parties, when a clause of non-responsibility is added to the contract, is not to plan for a simple reversal of the burden of proof in the event of the debtor failing to meet his obligations, but truly to limit the latter's liability to the consequences' of his gross negligence or his wilful misrepresentation, consequences to which he cannot escape; that such is the opinion of the doctrine and the jurisprudence, which require the proof of a gross negligence which can be likened to wilful misrepresentation, for the debtor's liability to be challenged..." CA Douai, Octobre 7, 1954, *Gaz Pal* 1954 &, jurisprudence, p. 302, *RTD Civ.* 1955. 1., p. 121 n° 50, Mazeaud observations.

58. Court of Cassation, commercial chamber, May 3, 1995: "But considering, firstly, that the court of appeal, while not holding the Fiduciaire company responsible for checking the computer's power supply by itself, was able to decide that said company had to ensure that the technical installation was properly cared for by professionals, since it had acted as counsellor to the Janin company in the choice of said computer equipment and its implementation;

Considering, secondly, that the court of appeal has only condemned the Fiduciaire company for the failure to perform its own obligations, and not for the failures of the Bull and Chausson companies;

But considering that, without ignoring the faults of the Chausson and Bull companies, and having, at the sole discretion of the court, accepted the expert's conclusions while adopting the appreciations of the first judges as to the distribution of responsibilities between the various concerned companies, then having noted that the share of the damage imputed to the Chausson and Bull companies had been mended by mutual agreement, the court of appeal did not contradict itself by deciding that the damage not yet mended, whose responsibility had been imputed by the court to the lone Fiduciaire company, did not concern the Chausson and Bull companies."

services will also become liable in the place of the mandating company. Likewise, in the event where the client delegates a service of general contracting (concerning, for example, the definition of needs, the conformity checks before the acknowledgement of receipt, the signature of meeting reports), the representative will automatically bind the liability of the client who will not be able to impute the fact or fault to a third party⁵⁹.

In a system of systems, the difficulty lies in the coordination and coherence of the levels of contractual commitments agreed on by the various agents, suppliers and service providers. If the client is aware of certain limitations of liability resulting from his direct contracting with joint venturers, the same does not apply to the responsibilities accepted by the subcontractors, nor by external subcontractors or suppliers in direct contact with the subcontractors. Thus, in the event where liability is challenged, it is not rare to be faced with some agents opposing inescapable limitations of contractual liability. Moreover, in the event of dire difficulties, the insufficient solvency of some agents prevents real compensation of the damages suffered by the client or by the other agents. As a measure of precaution, insurance subscription commitments should be drawn against certain covered risks as well as the corresponding coverage costs. Those are not sufficient, however, especially in the event of a redress or a winding-up of the people concerned by the decision of the court.

In order to be mended in civil liability, a proven damage must be both certain (that is to say, already done or unavoidable), personal to the one who acts to mend it, and direct (that is to say, a link of causality must exist between the fault and the damage). The direct or indirect character of the damage can be subject to diverging interpretations: some damages deemed indirect can be mended. It is therefore common practice to specify the indirect damages in advance, for example a tarnished brand image, an operating loss, etc. A minute examination of the risks of damage must be led by the buyer, so as to determine, for the client, the nature and scope of the possible contractual events.

One of the particularities of co-contracting lies in the nature of liability established between the various joint venturers. What the buyer is most interested in is the nature of the joint venturers' liability: joint or joint and several. In the case of joint co-contracting, each service provider or supplier will only be liable for his

59. Some authors point out that "the failure attributed to the third party whom the debtor entrusted with performance becomes imputable to the debtor himself, who takes over the actions of said third party" (J. Flour, J.L. Aubert, Y. Flour, E. Savaux, t. 3, op. cit., n° 205); "concerning the creditor contractor, these acts are imputed to the debtor himself, to the point where, if said acts constitute a gross negligence, the debtor will suffer the effects resulting from said character" (A. Bénabent, op. cit., n° 412-1).

share of supplies or services provided to the client. The client will therefore have to be able to prove the origin of a difficulty, and whom it can be imputed to, in order to challenge the liability of the actual defaulting joint venturer.

On the other hand, in the event of joint and several co-contracting, since all the joint venturers are united, the client will be able to take action against any of them if need be. The joint venturers will then turn on each other. Solidarity can apply to all joint venturers, or to a single joint venturer of them. In that case, the joint venturer's liability will be the only one to be challenged, and he alone will have to take appropriate action against the others.

Sometimes the contract must be suspended during its performance. Such is often the case in the event of *force majeure* or a temporary unsupportable failure of a supplier or of the contractor. The obligations are then temporarily suspended. The contracting parties can then specify under which terms they will end the contract if the event endures. The parties can also plan other causes for suspension within their contract. This option will have to be specifically mentioned in the contract, as well as the difference of price between both suppliers, difference which will be covered by the defaulting supplier⁶⁰.

3.4.7. *The end of commitments*

3.4.7.1. *The end of contractual commitments*

A contract may end for reasons of expiration, cancellation, rescission or impossibility of performance.

Impossibility of performance can occur if the contract could not exist because an event prevented its valid execution. Such is the case, for example, of imperfect consent, such as a willful misrepresentation (fraudulent maneuver aiming to abuse the other party's agreement), a misunderstanding on the content, companies which do not have the legal capacity of contracting. This cancellation requires legal proceedings.

In an ongoing contract, i.e. a contract which goes on in time, if services or supplies cannot be challenged (for example maintenance services), the contract is terminated by termination. This means the contract is terminated, but all the services already provided and the equipments already used will have to be paid for, at the right price, if need be in the view of experts.

60. This condition appears in the general administrative terms and conditions.

As for the contracts of instantaneous performance, such as the selling of a corporeal property, the sanction is like cancellation. When the contract is put into question because of a party's failure to meet its obligations, the contracting parties will return to the state they would be in if they had not signed a contract. This will lead to the restitution of the delivered property and the refund of the paid price.

An action in liability can always be filed against the defaulting party, which can be condemned to pay additional damages to the other party.

3.4.7.2. *Reversibility*

The consequences of terminating contractual commitments should be anticipated. Especially when it comes to services which last through time and for which, for example, competitive bidding will be opened to new service providers. It is important that the client keep a certain number of elements from the former contractual relationship. These can be documentations, anomalies, malfunctions or defects statistics, procedures to implement, archived database, etc.

Since reversibility is not a legal obligation, it is essential to write the terms of the reversibility into a contract. The implementation of this period of reversibility will allow the service's continuity⁶¹.

The full scope of the period of reversibility is defined within the contract: definition of the tasks to perform during reversibility, bond of performance during reversibility, organization and monitoring of its development, definition of the technical and documentary elements to transfer, including the databases, along with the transfer of the necessary rights of ownership.

3.5. Ownership rights

3.5.1. *Corporeal property*

It should be noted right away that the transfer of ownership of a property can be dissociated from the transfer of risks which has been studied on the level of the delivery. In a supplementary manner, these two transfers coincide and both happen during the agreement on the thing and the price, as far as the sale is concerned. But the contracting parties can, within the sale, agree on a clause of reservation of title. The goal is to delay the transfer of ownership until after the client has acquitted himself of the full price.

61. For the redaction of a reversibility clause, see the journal of the *CDAF*, Danièle Véret, "La réversibilité", 2006.

The full magnitude of this provision is apparent in the case of the client's receivership or winding-up, before the full payment of the price. This clause allows the seller to reclaim the property. To be valid, this clause must appear in clear terms and be differentiated from the other terms of the contract (in caps lock or print characters).

The transfer of ownership carries the property's license, the rights to make it fructify or garner profits from it, the rights to alienate it⁶².

3.5.2. *The patent*⁶³

An invention is a process which must be materialized by the manufacturing of an industrial good.

The invention is protected if it is new, through confidentiality or a patent application.

When there is a transfer of rights on a patented invention, the transfer must be done by the person who owns those rights.

Once the patent is filed, a certificate from the organization holding it might be produced.

A patent is filed with the National Institute of Industrial Property in France, and in national or international patent offices around the world.

3.5.3. *Copyrights and the particular case of software*

The intellectual creations which are works of the mind are protected if they are original (marked with the author's personality) and can be exploited. They are protected, whether complete or not. Software are protected through copyrights, with some specificities.

The author holds, from the mere act of creation (without formality), the proprietary and moral rights of his work. Among moral rights is the right to respect for his name and his creation. These rights are perpetual, inalienable and imprescriptible. Among proprietary rights are the rights of reproduction, use, distribution, translation, tuning, adjustment, alteration, etc. In Europe, they are

62. Civil code, sections 544 and following.

63. Section L. 131-3 of the code of intellectual property.

protected for 70 years starting from the death of the physical author, or from the date of creation in the case of artificial persons.

In French law, the transfer of proprietary rights must imperatively:

- be recorded in writing;
- include the delimitation of the transferred rights as for their width, destination, geographical zone of use and the length of time during which the transferring party can exercise the transferred rights.

Failing that, the transfer is declared invalid⁶⁴ and the person who does not usually hold the rights is declared a counterfeiter⁶⁵. Transfers can be operated on an exclusive or non-exclusive basis.

The author, the original owner of the copyright, holds the moral and patrimonial rights to his work.

The original title is protected along with the work.

A copy can be given for the private use of the copier, except for software. Access to the software's sources is most often planned for in the contracts. It is a measure of

64. The court of appeal in Paris answered that particular point in a judgement on December 20, 1989: "Since the failure to respect the terms laid out by section L. 131-3 are void *"nullité relative"*, only the authors can argue that a transfer of their rights is void."

65. Jurisprudence construes the section L. 131-3 of the code of intellectual property in a very strict manner. Let us take, for example, the judgement of the 1st Civil Division of the Court of Cassation, rendered on October 9, 1991. The Court of Cassation states that a general transfer cannot be admitted through a blurry clause. It thus reminds us that, according to the terms of section L. 131-3, the transfer of copyright is subordinated under the condition that their field of use is delimited in scope and destination, place and duration. Without trying to interpret the text, it calls forth its principle and applies it in a totalitarian way, without allowing a single exception. The Court of Cassation reminds: "It does not matter that the author has accepted the "principle of reproduction" of his work and perceived a percentage on some of the sales, as long as there is not any contract of transfer abiding with the formalism of section L. 131-3." The court of appeal of Paris also mentioned it in a judgement of May 2, 1975, declaring that "the reality of a transfer of copyright can only be established if it results from specific elements dispelling any doubt on the subject... Such a strict demand is imposed by the necessity to protect copyrights, which are in close relation to the author's personality so that the latter can only be dispossessed if he has agreed to it, and only within the limits of said agreement."

protection, insofar as the sources can be exploited by somebody other than their designer and creator⁶⁶.

Concerning software, the legitimate user can permanently or temporarily create the software's reproduction (downloading), translate it, tune it, adjust it, and conduct any other modification, etc., if those rights are necessary to allow the legitimate user to use the software, according to its destination, including the correction of faults.

However, the author can reserve the rights to correct any fault himself. He can also reserve the right to make a backup copy (not to be used at the same time as the running copy), since the right to observe, study or test might lead the user to create a distinct work without copying.

He also holds the right to come back to the original code (decompilation, disassembling) for the purpose of interoperability (creation of interfaces), but this right requires cumulative conditions:

- only the legitimate user can do it;
- the information necessary for interoperability has not already been made available in a fast and simple manner;
- reverse engineering must be limited to the parts which are necessary to interoperability.

The legitimate user cannot use the data thus obtained for any other end, he must not disclose them to third parties, and he must not use them to develop substantially similar software.

The data acquired during the implementation of interoperability remain confidential and could only be communicated to third parties, general contractors or project managers of systems interoperating within a system of systems, with the written agreement of the author of the software's part which has been disassembled or decompiled.

In the event of juxtaposition of services, for example on the same Internet site, provided by various contractors, the contractual technique will once again be used.

66. Cassation, Commercial Division, January 24, 2006: "for more than a year, the TIC and M.X. company had tried, without success, to enable the use of the software delivered to the Digitechnic company. The court of appeal has deemed that it wasn't established whether the release of the software's sources would have enabled them to find a solution to this problem, and has decided that the prejudice resulting from the withholding of these sources could be analysed as a loss of chances of enabling the use of the software, of which it has demanded the repair without appeal."

The parties agree on their terms of cooperation by defining the types of information and their signification, the alerts and restrictions of use, the disposal process in the event where the brand image is tarnished.

The provisions relative to copyrights do not only apply to software, but also to all the other works such as documents (reports, calculation notes, etc.), plans, diagrams, images, sound editing.

3.5.4. *The databases*

The producers of databases own the rights on the ones they have created, and the ability to grant some rights to a third party, such as a right of conversion.

3.5.5. *Designs and models*

Designs and models can be protected through a filing at the INPI (National Institute of Intellectual Property). In that case, the buyer must check whether the contract grants him the appropriate rights on these works. Indeed, the creator of such works is the only one who can decide on the use of his designs and models.

3.5.6. *Brands and logos*

The trademarks, whether they apply to business enterprises, products or services, along with the logos that represent them, are also filed at the INPI. The brand's filing is preceded by a search of possible former use. To use them, the client will need a deed of transfer, which can be featured as a clause in the contract.

3.5.7. *The domain name*

IP addresses are managed by the ICANN. The “.fr” addresses are managed by the AFNIC.

Domain names are allotted following the rule of “first-come, first-served.” From this can arise a conflict with intellectual property rights on brands, corporate names and other domain names. Abuses can be sanctioned on the grounds of usurpation. Example: registering, as a domain name, a client's corporate name, for which one is creating a site, or the registering of a cartoon character's name, “calimero.org”, which affects the author's ownership rights.

The “naming charter” lays out regulations for the recording, maintenance, transfer and deletion of a domain name.

Actions in unfair or parasitic competition can be led in the event of slamming.

3.6. The most adapted legal strategies

When the buyer is public, he can choose between signing a contract⁶⁷ or a framework agreement⁶⁸, an agreement for the outsourcing of a public service⁶⁹ or a partnership contract⁷⁰. All these contractual set-ups are framed by statutory regulations and cases.

The contract must allow for the fulfillment of the needs expressed by the public corporation within specifications. This therefore implies good knowledge on the nature of the supplies, the works necessary to the development of the system of systems, their volume, their monetary and technical cost allocation base. However, some flexibility is available for the contract’s design. It is possible to contract open-end contracts and options contracts. The allotment of contracts, allowing for a single competitive bidding for the acquisition of supplies, or highly varied services or works, enables the precise segmentation of the description of needs, and to turn to specialized market agents to answer each contract.

In the event where the needs, and most of all the desired result, are not easy to determine *a priori* by the public corporation, in the event where the financial or contractual set-up is not obvious, the contract can be drawn through the procedure of competitive meeting, which allows for in-depth discussion with the various candidates.

By combining all the contracts’ possibilities, it is possible to efficiently define the rights and obligations of each agent, as well as determine the obligations of the public corporation.

This contractual framework puts rather large constraints on the analysis of applications in relation to the established criteria. It is not really possible negotiate with each candidate (since such negotiations would thereby modify the candidates’ offers) in order to avoid breaking the law regarding public contracting; additional

67. Code of government contracting.

68. Code of government contracting, after the addition of the last European directive on contracts.

69. Law n° 93-122 of January 29, 1993, called “loi Sapin”, and law n° 2001-68 of December 11, 2001, called “loi Murcef”.

70. Order n° 2004-559 of June 17, 2004, and decree n° 2004-1145 of October 27, 2004.

clauses, in the event where new needs appeared during the performance of the contract, are closely monitored; any appeal to complementary contracts or identical contracts must be strictly justified. As for the use of public money, within a just management and considering the fact that they answer to government accounting, financial commitments are strictly monitored and must not significantly exceed the allotted budgets.

To implement a framework-agreement means to draw an agreement between the public corporation and the contractor, defining the general principles of the contractual relationship (type of services, rates, liability, etc.). Contracts are signed subsequently, on a case by case basis, each time the public corporation expresses needs which belong within the scope of the agreement.

Considering they represent a new notion for French public law, issuing from the recent application of European directives, their implementation within systems of systems requires some reflection, and an analysis of each case to determine their relevancy.

The outsourcing of a public service is used when the services belong within a mission of public service and generate a strong investment assumed by the agreement's contractor, who is remunerated according to the performance of the service he is providing for the public corporation.

The partnership contract consists of entrusting a service provider, for a long period of time, with a global service relative to the financing, construction, maintenance, operation or management of works or equipments necessary to the public service. This contractual set-up implies that the risks are shared between the public corporation and the joint venturer.

3.7. Conclusion

The contractual set-ups made available to the public corporation have become more varied in recent years. Each contractual set-up has a determined framework, with prerequisites, constraints and terms. For systems of systems, it might be pertinent to combine several kinds of agreements, insofar as possible effects have been anticipated.

Given its complexity, in order for the system of systems to be advantageous and successful, given the particularity of every contractual set-up, which we only briefly mention in this book, and given the recent implementation of the contractual set-ups available to the public corporation, we can only advise people to lead an in-depth reflection on the subject as soon as they need a system of systems.

Chapter 4

The Human Factor within the Context of Systems of Systems

4.1. Introduction

Within a book about systems of systems, the chapter dedicated to the human factor builds on the disciplines of social sciences and applies them to the context of systems of systems. As such, this chapter is not an introduction to these various disciplines. However, readers wishing to learn more about those disciplines will find a selection of related resources within this chapter's bibliography. Moreover, this chapter does not pretend to be exhaustive. Social sciences feature many varied and highly rich disciplines, which can sometimes present a different point of view on a single object. In such a context, this chapter only broaches a certain number of concepts which are relevant to the notion of system of systems.

Within systems engineering, and *a fortiori* within systems of systems engineering, the aspects relative to the human factors, in their entirety, have to this day mainly pertained to the ergonomics of the operating consoles, and to the human-system interfaces. Many documents exist on the subject, which readers may consult, including books by Chapanis [CHA 96], Norman [NOR 90], and Kolski [KOL 97], [MAH 09], and [LEP 03].

When the aspects relative to processes, activities, and organizational dimensions are treated in the field of information systems, they are studied within a framework

built on reductive organization models, which do not take recent works in social sciences and management into account.

Following this introduction, the second part will present a definition of a system and succinctly study the epistemological aspects linked to the notion of system.

The third part will narrow down the issue. What is meant by system and system of systems? In which way are organizations concerned by systems of systems?

The fourth part will present the current standpoint of systems engineering on the notions of process, activity, organization, decision making, and show the important limitations of the standpoint.

The fifth part will study the organizations' complexity, from the point of view of social sciences, and its impact on systems of systems.

The sixth part will present the implementation of the concepts of social sciences within network-centric operations, an example of system of systems.

Finally, the seventh part will identify some good practices which should be implemented in order to broach the aspects linked to humans within systems of systems.

4.2. Definition and epistemological aspects

The notion of a system has several meanings, depending on the adopted point of view, which can either be the one of systems theory, systemics, or systems engineering.

There are many works on systems engineering [AFI 05, ISO 02, MEI 98], as well as systemics [ARA 84, BER 73, DUR 79, LEM 90, MOR 05, VAR 88]. To remain coherent with the studies led in the field of systems of systems, we will follow the definition of systems engineering, while pointing out the limitations of the definition in order to study the dimensions linked to humans, human activities, human organizations, business enterprises.

According to the standard ISO 15 288 ([ISO 02] p. 4), a system is a combination of interacting elements organized to achieve one or more stated purposes.

NOTE 1: A system may be considered as a product and/or as the service it provides.

NOTE 2: In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g. aircraft system. Alternatively the word system may be substituted simply by a context dependent synonym, e.g. aircraft, though this may then obscure a system principles perspective.

As for Meinadier ([MEI 98] p. 24-25), he defines “a system as being a composite set of personnel, hardware and software organized so that their interoperability, in a given environment, fulfills the purposes for which it was designed.”

Finally, for the AFIS ([AFI 05], on the page entitled “the system and its definition” (*Le système et sa définition*),

“a system is described as a set of elements interacting with each other and with their environment, integrated so as to provide the system’s environment with the intended services. Therefore, a system features new properties resulting from the interaction between its components: if you integrate components to create a system, it is indeed to benefit from the synergistic effects resulting from their interaction. The art of systems engineering is to obtain, through interaction, the intended synergic behaviors by containing unintentional emergent behaviors within acceptable limits. In systems engineering, the system definition includes:

- the definition of its subsystems and components (hardware, software, organizations and human skills) and of their interfaces, the base of the sought interaction;
- the definition of their life cycle processes, which enable their design, development, testing, distribution, deployment, operation, maintenance and disposal, and therefore the definition of the products necessary to said processes.”

These definitions of the notion of a system reveal what Morin [MOR 05] calls the *system analysis* in systemic theory. Some authors, in the field of systems engineering, reference systemics, but more often do so abusively, without rigor, and without taking into account the characteristics of systemics which we are about to develop.

Several definitions are in use in the field of systemics. We will mention three of them, before taking an in-depth look at the structuring definition offered by Boulding.

According to Morin ([DUR 79] p. 8), a system is “a global organized unit of interrelations between components, actions or individuals.” According to Ladrière

([DUR 79] p. 8), a system is “a complex object, formed by distinct components linked through a certain number of relationships.” Finally, de Rosnay ([DUR 79] p. 8) defines a system as “a set of components dynamically interacting with each other, organized according to a specific purpose.” This last definition is appropriate for man-made systems, but does not fit the natural systems, unless we stand by a creationist hypothesis of intelligent design, hypothesis which we will not follow in this chapter.

Boulding [BOU 56] (and [DUR 79] p. 25-31, for a synthetic summary), on the other hand, differentiates between several levels of systems, depending on their complexity. The levels are as follows:

- the static structure, for example the framework, the skeleton. This is the structural description of a system, the anatomy of the universe or the pattern of electrons around a nucleus, the anatomy of the cell or the plant ([BOU 56] p. 202);
- the simple dynamic system, with predetermined, necessary motions, such as a clockworks. The dynamics of the universe, or the one of an atom, fall into this category ([BOU 56] p. 202-203);
- the cybernetic system or control mechanism, also nicknamed the level of the thermostat. It differs from the former level in the fact that the transmission and interpretation of information is an essential part of the system. The system will move to the maintenance of any given equilibrium, within limits. This is the homeostasis model of physiology ([BOU 56] p. 203);
- the fourth level, also called the level of the cell, is the one of the open system, which self-maintains through matter consumption. Thus, the existence of the simplest living organism is inconceivable without ingestion, excretion and metabolic exchange. Closely connected with the property of self-maintenance is the property of self-reproduction ([BOU 56] p. 203-204);
- the fifth level, also called the genetic-societal level, is typified by the plant. The differentiated cells which form the tissues enable the division of labor. Thus the leaf is protected by its skin, while the photosynthesis happens in the parenchyma. There is also a sharp differentiation between the genotype and the phenotype ([BOU 56] p. 204);
- the animal level is characterized by increased mobility, teleological behavior and self-awareness. The development of specialized information-receptors (eyes, ears, etc.) leads to an enormous increase in the intake of information. This goes hand in hand with the development of nervous systems, which builds up a knowledge structure or view of the environment as a whole. This is a structuring of information into something essentially different from the information itself. The information is “captured” by the image and added to it, leading to the reorganization of the image

and radical changes in behavior in apparent response to what seems like a very small stimulus ([BOU 56] p. 204);

- the next level is the human level. In addition to all of the characteristics of animal systems, man possesses self consciousness. His knowledge has a self-reflexive quality: he not only knows, but knows that he knows. He has the ability to produce, absorb and interpret symbols. He is also distinguished from the animals by a much more elaborate image of time and relationship. He exists not only in time and space but in history ([BOU 56] p. 204-205);

- the eighth level is the level of social organizations. The unit of such systems is not perhaps the person, but the “role”, that part of the person which is concerned with the organization or situation in question. Social organizations can be described as a set of roles tied together with channels of communication. The interrelations of the role and the person however can never be completely neglected, and the perception of a role is affected by the personalities of those who occupy it. This level is also the level of symbolic and artistic activities, and of the complex gamut of human emotion ([BOU 56] p. 205);

- the last level concerns the transcendental systems ([BOU 56] p. 205).

To Boulding’s classification of system levels, we can add, non-exhaustively, some of Morin’s characteristics of complexity:

- the interdependence of the subject and the object ([MOR 05] p. 51-61);

- a system’s ability to maintain itself in a state of stability and continuity, “the structures remain the same even though the components change.” ([MOR 05] p. 31). Thus, a living organism remains the same from its birth to its death (continuity), and all its cells, all its tissues have changed, its cells have died and been replaced, the tissues have grown, have become differentiated, etc.;

- “the organizational laws of living organisms are not based on equilibrium, but on righted or compensated disequilibrium, on stabilized dynamism” ([MOR 05] p. 31);

- the differentiation made by von Neumann between “the living machine (self-organizing) and the artificial machine (simply organized)” ([MOR 05] p. 43); indeed, “the individuality of the cybernetic object, as an artificial machine, is linked to its organization principle; but this principle is external, created by man.” ([MOR 05] p. 45);

- the systems’ autonomy ([MOR 05] p. 42-46);

- the system’s self-organization and self-production capability, in close relation with their environment, which opens the way to the notion of self-eco-organization,

“the effects and the products are necessary to the process which generates them” ([MOR 05] p. 42-46);

– the differentiation between the programming of an automaton and the strategy of human action, for “everything concerning the emergence of something new is not trivial and cannot be predicted” ([MOR 05] p. 109), which is characteristic of emergencies, the ability to act within an unstable environment.

This brief analysis underlines the differences between, on the one hand, the artificial systems from the standpoint of systems engineering, and on the other hand, the natural systems, human beings and organizations, from the standpoint of systemics. These differences echo von Neumann’s differentiation between the living machine and the artificial machine, Boulding’s levels of complexity, and Morin’s characteristics.

4.3. The issue

Now that we have defined the notion of system, we will specify the notions of system and system of systems, within systems engineering, that is to say artificial systems designed and engineered by human beings, and end on the constraints of organizations on systems of systems.

4.3.1. *Example of system within systems engineering*

Let us use an example. A ticket booking system provides the clients of a transportation company with the ability to purchase tickets in order to travel on the company’s network. This is its purpose. This ticket booking system is a subsystem within a larger system, which offers the company’s client a complete service, from the purchase of the trip to the actual journey, and sometimes optional services such as onboard meals or the transfer of the client’s vehicle. The company’s partners can add to those services, with car rental, hotel booking, parking places, etc. In order to offer the option of booking an onboard meal upon purchase of the ticket, we must link, interface, integrate the ticket booking subsystem to the onboard meal booking subsystem, and the trip subsystem to the subsystem monitoring the preparation and distribution of onboard meals. Such a service can only be provided through the interaction of the subsystems “ticket booking” and “onboard meal booking” within a system of a higher level.

The sale system itself can be broken down into subsystems. The subsystems may be dedicated to selling methods, for example in an agency, on an Internet site, at a booth, etc., or common to the different selling methods, such as a booking central.

The ticket booking system includes:

- end products, i.e. the tickets, which can be material or immaterial;
- components, i.e. the equipments and software needed to create those products (booking central, Internet server, checkout counter, ticket machine, etc.);
- enabling products or systems, which enable the development of this booking system, without however being components of the system (telecommunication networks, test and measurements machines, etc.);
- production and support processes and activities, needed to create those products (define a price policy, sell the tickets, operate the sales equipment, maintain the equipment, treat the accounting and financial data, collect statistics on the sold products, etc.);
- operators, who carry out those activities according to established procedures and trade regulations, using specific skills in terms of general knowledge (knowledge of the rate codes), technical knowledge (knowledge of how to refill an automaton's change machine), and good manners (courtesy towards customers);
- the organization of activities;
- information and data, such as rate tables, the list of nodes within the transportation network, the distance between nodes, visual signs to keep the clients up-to-date about the availability of the offered services and products, etc.;
- resources of varying nature (energetic, etc.), consumables (paper, etc.), necessary to the carrying out of those activities and the manufacturing of those products;
- etc.

First of all, we will identify the process which concerns the operation and maintenance of the selling system's equipments. In order to design, develop and operate those equipments, other – non technical – processes must be implemented, and other skills must be called on.

Should the company sell tickets on the Internet? Should it sell products targeted to students? Should it offer optional services such as onboard meals? If so, to which customers should it offer them? Should the company work with external partners in order to offer further optional services? Those questions are answered through management, business and marketing activities, which, while carried out by people other than the staff taking care of the ticket booking system, are still part of the company. Should the company get funds to offer services to specific customer categories, such as unemployed people? This question is not in the hands of the

company, but of the political leaders, local or regional. In the context of systems engineering, these various agents are called stakeholders.

Already, we can see a diversity of processes, activities, and roles within the transportation company, but also outside of it. Similar activities are grouped into processes, allowing the company to carry out its missions. They are the processes of sale, management, as well as the core processes of people transportation.

In one way or another, the organizational structure of the company reflects the structure of those processes. Under which criteria is the company organized? How are those processes structured? Is there an ideal manner, or even a single appropriate manner, of organizing this transportation company? How can we be sure that the adopted organization is the proper one? Such are the questions asked of management and studied within organizational theory [ROJ 03], and formalized in Mintzberg's book [MIN 82].

In another field, a bank will identify its objectives, its processes, its organization, the activities it must carry out, the necessary skills to do so, the necessary training to acquire and maintain those skills, etc.

An army will organize its processes and its organization so as to fulfill the operational missions which the government, for which it implements this defense policy, will assign to it.

4.3.2. *The notion of system of systems*

The notion of system of systems appears in Boulding's works in 1956 ([BOU 56] p. 202), but it has only very recently been used about software-intensive systems, mainly, but not exclusively, in the field of defense.

According to Maier [MAI 98], a system of systems is defined by the following characteristics:

- the operational independence of the components: a system of systems is composed of independent systems, each with its distinct use and able to provide operational services on its own [MAI 98b];
- the managerial independence of the components: the components of the system of systems are used and managed independently. This managerial independence can be represented by as many general contracting and project management as the number of systems which constitute a system of systems [MAI 98b];

- evolutionary development: the system of systems' development and existence are evolutionary, with functions, purposes and components added, modified, erased [MAI 98b];
- emergent behavior: a behavior which does not reside in any of the system of systems' constitutive systems, emerging from the interactions between systems within the system of systems [MAI 98b];
- geographic distribution: the systems are located in different places and can only exchange information [MAI 98b].

Without being exhaustive, a system of systems can be characterized by its ability to provide a service from beginning to end. To produce this capability, different systems are called upon, which belong to different organizations (criterion of managerial independence), systems designed to perform their organizations' activities (criterion of operational independence). The ability to provide a service from beginning to end can be considered as an emergent behavior, insofar as it is not *a priori* specified and is the product of the integration of the existing systems. Here, the notion of emergent behavior seems to be weak, in comparison with the strength of the notion of emergence within systems theory, when we offer life or social links as examples of emergent behaviors. In that context, new properties, such as growth, the ability to reproduce (self-reproduction) and organize (self-organization), do not exist in the constitutive systems and are by nature different. In the context of systems of systems, the emergent behaviors are functions, the nature of which is not radically different from the functions of the constitutive systems.

Let us take an example in a field where anybody could be a customer. The service aims to provide drivers, in real time, with information on highway traffic and offer them alternative routes in case of traffic jams.

To provide this service, we must first collect data on the state of traffic on the selected highway networks. This information is provided by various sensors scattered over those networks, video cameras displaying the traffic status, detailed reports by the security agents present on the highways, etc. For each network, those pieces of information are consolidated in a dedicated monitoring center. The data is then integrated in a compiling and integration center. This is where an integrated representation of the traffic status is created.

This integrated representation is then transferred to different communication medias aimed towards users. For example, the French Internet site Sytadin (<http://www.sytadin.tm.fr>), the French radio frequency 107.7 FM, variable-message signs displaying predicted travel times to such or such part of the network.

This information can be sent to the navigation systems of drivers who have subscribed to the feeds, via data communication devices such as GPRS. Each navigation system can dynamically compute the driver's itinerary according to the traffic jams. To this end, the navigation system must locate the driver's current position, determine the driver's itinerary between the current position and the destination, determine the location of traffic jams along the itinerary, compute an alternative itinerary in order to bypass traffic jams, offer this itinerary to the driver.

Such a service cannot be carried out through one system which would have been conceived, from the start, for that purpose only, and would be functioning autonomously in order to provide it. On the contrary, we must implement a set of various systems which have all, more or less, been designed independently from one another, with different purposes, and which, once integrated together, will form a system of systems.

Below is a non-exhaustive list of systems:

- the monitoring systems of the various highway networks (for example, the Lutèce and Sirius systems, which respectively concern the Parisian beltway and the suburban highways around Paris), which include data collection subsystems such as traffic sensors, communication network subsystems linking the sensors to the technical information systems of the networks' operators;
- a system designed to compute the surveillance data sent by the various highway networks;
- a system of data telecommunication (for example, GPRS) or of radio broadcasting of updates on the state of traffic;
- a GPS system for the geolocation of drivers;
- the driver's embedded navigation system.

Another emblematic example can be found in emergency situations, where cooperation between firefighters, ER doctors, rescue teams, NGOs, requires the help of different organizations, each with their own objectives, process structure and technical equipment carrying out those processes. The integration of their communication and information systems is akin to the designing of a system of systems.

Likewise, the coalition of several countries to maintain the peace also requires the design of a system of systems in order to carry out those operations.

4.3.3. *Organizations' constraints on systems of systems*

The previous sections have helped us demonstrate several dimensions of the considered systems: on the one hand, a technical dimension of hardware and software equipment, the consumables; on the other hand, an operational dimension, which we can qualify as “non technical”, pertaining to processes, activities, organization of activities, trade regulations, skills, etc., within an organization.

In the case of a system of systems, providing a service from beginning to end requires various systems, identified above. Each of those systems was designed independently from the others, within various organizations, to answer specific purposes, characteristic of each organization.

The organization in charge of the design of the navigation system can be a motor vehicle equipment manufacturer, while the GPS system designed for the American Department of Defense, or the telecommunication system designed for a telecommunication operator, or the monitoring system designed for a highway network's dealer, can all be designed by a technical monitoring systems industrialist.

For the navigation to be efficient, the geolocation via GPS technology must be sufficiently fine and precise. When the civilian world gained access to the GPS technology, such a thing was not at first possible, since the civilian mode (SPS mode) included an error deliberately introduced by the American military, an error which limited accuracy to 400 feet, while the military mode (PPS mode) had an accuracy of 230 feet. Selective availability was deactivated in 2001, enabling accuracy ranging from 10 to 160 feet in the civilian world [LEV 04]. It was therefore clearly a strategic decision and not a simple technical limitation.

Is this navigation service, within the context of intelligent transportation systems, the result of discussion and cooperation between the organizations in charge of the concerned systems, and none other? No. The political motivation was important, both on the French and European levels, and translated into studies on the technical feasibility, technical specifications and incentive measures towards the various public and private partners.

For example, on the European level, the objectives [EUR 03] include:

- the management of mobility;
- the improvement of access for elders and disabled persons in transportation facilities;
- the management of freights and fleets of vehicles;
- the upholding of transportation safety;

- the treatment of emergencies and fires.

From that point on, how are the operational and organizational dimensions, in their broader meaning, of the systems and the systems of systems, treated?

The following section will study the current framework describing the organizations and the processes, from the standpoint of systems engineering, and underline the limits of the framework. The next section will study the complexity of organizations, from the standpoint of social sciences, as well as its impact on systems and systems of systems.

4.4. Current human factors in systems engineering

In systems engineering and systems of systems engineering, how are the aspects relative to processes and organizations, and operational aspects, treated? The following sections will provide examples of the way organizations are understood, treated and modeled.

4.4.1. Designing an organization from the standpoint of systems engineering

In their article, Handley and Levis [HAN 03] compare two organizational architectures, depending on their mission requirements. To this end, they develop an executable model of the organization and validate the model from experimentation results. The modeled process is a decision making process.

The modeling of the organization is based on automata with specific finite states, namely colored Petri nets. An organizational architecture is characterized by the field of responsibility of the person who makes a decision, including the task they must fulfill, the resources they use. For the operator, the aim is to detect the threats which appear on his control screen, characterize them, fight them with a solution adequate to their destruction. The process implemented to fulfill a task consists of a sequence of events, namely:

- appear (a threat appears on the operator's screen);
- detect (the operator's activity);
- identify (the operator's activity);
- attack (the operator's activity);
- destroy (the operator's activity);
- disappear (the destroyed threat disappears from the screen).

This sequence of events is linear, with no possibility of backtracking, and it does not feature complexity such as parallelism between sub-tasks. Each event takes a certain amount of time. On the other hand, the person who makes the decisions must perform a set of task sequences, in parallel, which results in work overload.

The organization and the process can be characterized with the following elements:

- an activity of elementary decision making, performed by an operator;
- the lack of any activity which would require various operators to interact, which is an important thing in the context of systems of systems and its notion of shared situational awareness, a notion we will look at in further detail in an ulterior section: “Sensemaking in organizations and mutual intelligibility”;
- the lack of any problem solving activity (implementing the simple Rasmussen rules): and yet, the information can be confusing, ambiguous, putting high demands on the operator, which definitely turns identification and classification into problem solving;
- an overload resulting from simple activities, not requiring changes of context from the operator, even though an operator is not single tasking. Indeed, the operator must manage several tasks at a time, would it be only the exchanges with other colleagues, the reporting to the line of authority. Many works on cognitive psychology show that changes of context take up a large part of the cognitive resource and significantly influence the work load.

The experiments hold a highly reduced external validity over the issues on organizations. The external validity accounts for a model’s ability to express the richness of the observed phenomenon. The external validity is important when the model is realistic and correctly represents the observed phenomenon. On the other hand, the external validity is poor when the model only incompletely represents the observed phenomenon, and when the incompleteness impacts the comprehension of that phenomenon. Both the models of organizations, with finite state automaton, and the description of the radar operator’s activities as being a strictly sequential series of elementary actions, are not at all realistic.

Under the influence of its environment, the organization restructures, reconfigures itself. New states may appear which cannot be pre-defined. The executable model of an organization will have to take into account the set of characteristics which structure and shape the organization. A human operator monitoring radar is leading a complex activity, which pertains to sensemaking. These characteristics will be studied in the section “The organizations’ complexity from the standpoint of social sciences: impact on the systems of systems.”

Following a relatively similar approach, Browning [BRO 99] applies systems engineering to organizations. In the context of integrated product teams, the breakdown of the product architecture affects the compartmentalization and integration of the developing organization. The organizational interfaces must mirror the interfaces of the product architecture, for the processes and activities which must be led to manufacture the product are interdependent. The integrated product teams who implement these activities and processes are, in their turn, interdependent. These last interdependencies require high coordination efforts from the team. The goal is to find and identify the most appropriate integration level for those integrated product teams. The interfaces can be designed while taking into account the needs in terms of data flows between each integrated product team. This data flow can be regulated following a “just-in-time” logic, with a sufficient level, neither too high, nor too low, adjustable, efficient, documented, measurable and adapted to the project’s task.

The same thing applies to the article written by Faisandier [FAI 06], for whom organizations can be considered as systems. Like a system, an organization can be defined by its purpose, its missions, its objectives. Faisandier analyses the organization using the concepts of systems engineering, and demonstrates the functional and physical aspects of organizations, the aspects of perimeter and interface, as well as those of hierarchical breakdown.

All these authors study formal organizations, within a Taylorian approach. Not one of them studies the organizations’ most important, more pertinent characteristics which allow them to coordinate within a collective action. Likewise, they do not study the environment’s influence on the organizations’ structure. Finally, they do not study the eminently human character of organizations, in which the agents have various interests, various objectives, each agent in simultaneous participation with several organizations, and playing different roles in each of them.

4.4.2. *Social networks and multi-agent systems*

The analysis of social networks seems to offer the right approach for the study of human factors in the context of systems and systems of systems, which are in a way networks of systems.

Carley [CAR 98, CAR 99], bases his work on the analysis of social networks. Let us first look at what is social network analysis.

In social psychology, the concept of a network was used very early on, in the 1930s, and has remained an important component of the discipline ever since. Social network analysis was born from a set of works, on the far end of social sciences and

mathematics. The purpose of these works is to describe the structure of groups, the relationships in a community, such as a tribe of a rural society [SCO 00]. They emphasize the interpersonal relationships existing within the social system. This reflects the informal organization, in opposition with a formal organization such as the one displayed in an organization's organizational chart. These networks take into account the concrete patterns of interpersonal choices, attraction, repulsion, friendship, etc., and feature communication channels through which information is communicated from one person to the next, or through which one individual can influence another.

The analysis of social networks belongs within an approach based upon structuralism and builds on general system theory in order to, on the one hand, treat the group as a system, and on the other hand, comprehend the interdependencies between the group and its environment. These analyses bring out the transmission of ideas and innovations, the spreading of rumors, the cooperation and leadership within a group. They help demonstrate the stability, cohesion, integration within a group, as well as the power play, the conflicts and changes, with negotiations, bargaining, coercion and the establishment of social standards.

Carley [CAR 98, CAR 99] builds on these studies in the field of social networks, as well as on the models of multi-agent networks, in order to take the dynamic aspects into account and clarify the evolution of such a network. This also helps understand the factors of the stability and instability of social networks, in terms of evolution, performance and adaptability. It integrates, apart from the individual dimension, the dimensions of the tasks and the resources necessary to realize those tasks, and uses metrics to describe these phenomena, such as the mental load related to the tasks.

Originally, people in social networks were not treated as active and adaptive agents, able to make decisions, to modify the networks they belonged to. These persons were mere positions within a network. The implementation of the multi-agent technology enables the display of the social and cognitive processes which influence "who is liable to interact with whom", taking into account the cognitive similarity between individuals and their level of expertise.

In that context, the agents learn, take part in events, in actions of social and organizational change which pertain to the network's structure. The agent is no longer studied in terms of position, but in terms of process. The dynamic character of social networks emerges from those actions.

The DyNet tool [CAR 03] is a step towards understanding the way social networks evolve, change, adapt, can be destabilized, and the emergence of leaders. Carley [CAR 03] defines the signs of destabilization, such as the decrease of the

information flow through the network, and the inability of the network's members to reach a consensus, or the length of time needed to reach it. In this way, the emergence of leaders can be linked to the cognitive load, and various types of networks can be compared, for example a centralized and hierarchized network against a decentralized and distributed network.

Carley [CAR 01a] defines an organization's culture in terms of knowledge network. This definition of culture is diminutive and does not take into account the numerous characteristics described below, in the paragraph "Professional, organizational and national cultures in organizations." For Carley, culture as knowledge is shared, argued on, negotiated and distributed by the organization's members. The organizations are then defined as synthetic agents able to learn, which enables the emergence of organizational behaviors and an organizational knowledge [CAR 01a]. Structural learning occurs when the changes happen within the social network, resulting from the addition or withdrawal of an agent, or the maintenance, or lack thereof, of a relationship between two agents. Carley puts these works into practice to examine the possible impacts of unanticipated events, in order to evaluate a threat, and applies them to the context of terrorism.

While the first works on social networks analysis consisted of the treatment of data issued from anthropological, social or psychosocial studies, within an empirical approach, the works of Carley [CAR 99, CAR 01, CAR 03] are essentially rooted in simulation, and do not seem to be based on empirical data. For Carley, the notions of networks similarity, cohesion and stability, based on interpersonal relationships and elective affinities within social networks, only purport to cognition. The attraction between individuals, their ability to interact, depends on the degree of similarity between the individuals' knowledge.

Moreover, even if [CAR 01a] references the notion of culture, with standards, values and rituals, once again only the cognitive dimension is taken into account, exclusively highlighting the notion of knowledge.

Finally, very recently, Sweeney [SWE 05] compared three different methods, including the analysis of social networks, in the context of systems of systems used to military ends.

4.4.3. *Wrap-up on the current human factor in systems engineering*

The various points of view which we have analyzed all belong within a simplistic approach, based on formal models, more or less elaborated, up to highly developed models of cognitive agents. Moreover, they present us with a standardizing logic, meaning that there could be, in the absolute, a good form of

organization, a form which we should strive to follow. They only understand organizations in terms of prescribed processes, and do not take the actual implemented processes into account.

These various points of view have more to do with rational organizations than with natural systems, such as was defined by Gouldner ([THO 04] p. 4). The former are the result of a “closed system” strategy, while the latter are the result of an “open system” strategy, implemented to study organizations. As underlined by Thompson, ([THO 04] p. 9 and 10), both points of view must be taken into account.

The same thing applies to Carley, whose logic is more cognitive than social or cultural, building on a computational model, human behavior being reduced to data processing, which is highly simplistic compared to the set of works on social networks [SCO 00]. The notion of cognitive load also follows a computational model and does not seem to build on any empirical data to justify and validate the scope and limits of this concept on the level of human groups. Non-cognitive elements, such as the interpersonal and informal relationships between individuals, at the base of the works on social networks, as we have previously seen, the shared beliefs and values, the social regulations, the individuals’ character traits, the dominant/submissive behaviors, extraversion, openness or deceit, the trust given to such or such individual and denied to another, all these are not taken into account as parameters that can influence, in any way, the structure and evolution of an organization. When, on the contrary, human organizations are actually influenced by these factors [ROJ 03].

These various points of view do not take into account the works on systems theory, like Boulding’s [BOU 56], Le Moigne’s [LEM 90], or Morin’s [MOR 05], discarding the notions of adaptation, growth, self-organization, self-eco-organization, notions which help grasp the organizations’ complexity. The aspects pertaining to the competition between organizational structures (for example to access some of the organization’s resources), the conflicts which arise in such a context, the way of managing and resolving these conflicts, are not featured in these articles. These organizations integrate humans with their own objectives and interests, their skills and knowledge, elective groupings of individuals. These important elements are not taken into account. Finally, organizations interact with their environment. This environment is social, economical, political, cultural, legal, and includes other organizations such as suppliers and customers, oversight entities, state institutions, etc. These interactions have mutual impacts. On the one hand, the organization modifies its environment, and on the other hand, this environment brings about changes and evolution. Moreover, the organization can be in a situation of conflict with other organizations which do not share the same values, beliefs, purposes, objectives. The set of these elements is studied in works on organizations, Rojot’s among others [ROJ 03].

4.5. The organizations' complexity from the standpoint of social sciences: impacts on the systems of systems.

In systems engineering and systems of systems engineering, how are the aspects relative to the process, the organizations and the operations treated? The following parts will provide examples about the way organizations are understood, treated and designed.

4.5.1. *The organizations' design from the standpoint of social sciences*

If, as we have just seen, engineers are now discovering organizations and want to put their concepts straight into practice, the notion of organizational design is more than 25 years old and rich in numerous discerning concepts not present among the references of recent articles in systems engineering. We will start by sketching out rules, patterns of organizational design.

In his book, *The Structuring of Organizations*, Mintzberg [MIN 82] worked out a broad synthesis of the many works that had been carried out at that time, and offered a structured approach to the description of organizations' structures and functioning, and the way to design them.

Organizations consist of five parts ([MIN 82], p. 35-50):

- the operating core brings together the operators whose work is directly linked to the production of products and services. It produces and provides those products and services;
- the strategic apex determines the organization's mission and strategy, makes sure it efficiently fulfills its mission by meeting the needs of those who control it. It allocates resources, resolves conflicts, and defines roles and responsibilities. It also monitors the organization's limits and its relations with its environment;
- the middle line, composed of executives, links the strategic apex to the operating core. Each link of the hierarchical chain accomplishes and reverberates, on its own level, the work of the hierarchical apex;
- the technostructure is composed of analysts charged with the design and adaptation of the structure. They work out the work flow, the procedures the operators must implement, through the processes' standardization;
- the support staff (logistical functional units) are specialized in precise functions: research and development, communication, human resources. In that regard, they indirectly operate in the workflow.

These five basic parts are linked through various flows: flows of authority, equipment, information and decision process ([MIN 82], p. 51). The organization is ([MIN 82], p. 51-81):

- a system of formal authority, described by the organization's organizational chart;
- a system of regulated flows, including operational flows, control flows and horizontal flows of operational information;
- a system of informal communications, out of the regulated channels of information and decision;
- a system of work constellations composed of co-working operators;
- a system of decision making processes. Programmed decision making processes, daily and standardized, and unprogrammed processes (*ad hoc*) which aim to solve poorly structured problems, are implemented within the organization.

Thus, Mintzberg makes out five means of coordination, allowing the agents to coordinate within an organization:

- “mutual adjustments allow coordination with simple informal communication” ([MIN 82], p. 19);
- “direct supervision describes the coordination mechanism by which an agent becomes responsible of the work of others” ([MIN 82], p. 20);
- standardization of work: “work procedures are standardized when work itself is specified or programmed” ([MIN 82], p. 21);
- standardization of outputs consists of “standardizing work products, for instance specifying the product's dimensions or the level of performance to achieve” ([MIN 82], p. 21);
- standardization of skills: “skills and knowledge are standardized when the workers' training is specified” ([MIN 82], p. 22).

These coordination means depend on the requirements about which processes to implement and which products to treat, to transform. We will study this shortly, within the section dedicated to “The internal and external environment of organizations”.

Mintzberg also distinguishes between nine organizational design parameters, including the design of work stations (the first three points), the design of the superstructure (fourth and fifth points), the design of lateral links (sixth and seventh points), and finally the design of the decision-making system:

- The design of the work station must take into account the tasks to achieve and the meaning of these tasks for the operator. Over-specialization has often ended with failure ([MIN 82], p. 87-96).

- The “formalization of behavior is the design parameter by which work processes are standardized [...]. The operator’s behavior is controlled, reducing variability” ([MIN 82], p. 98-99). The formalization may be linked to the work station, the work flow, or may be achieved through rules. This helps differentiate between the structures of organizations, whether bureaucratic or organic. The former describe standardized behaviors, while the latter are characterized by their lack of standardization” ([MIN 82], p. 97-107).

- “Training concerns the processes by which operators acquire knowledge and knowledge, while socialization describes the processes by which operators are indoctrinated with organizational rules and norms” ([MIN 82], p. 109-113). We will look deeper into this last concept in the part dedicated to “Professional, organizational and national cultures in organizations”.

- Units: workstations are grouped within units according to work characteristics and the activities to be performed. Workstations and tasks contributing to the same process, providing goods or services to the same customers, or sharing the same skills, are grouped together. “The creation of units is the basis of formal authority and of the organization’s hierarchy” ([MIN 82], p. 115). “Creating units stimulates two mechanisms of coordination (mutual adjustment and direct supervision). This also enables coordination by the standardization of results, through the measure of the production unit’s performance” ([MIN 82], p. 117). “On the other hand, the creation of units promotes coordination within groups, while reducing coordination between groups”. ([MIN 82], p. 118). Mintzberg identifies many grouping criteria, such as workflow interdependencies, processes interdependencies, scale interdependencies, or social interdependencies ([MIN 82], p. 123-131).

- The size of units, namely the number of operators whom a leader can directly supervise. There are various kinds of organizational structures: flat, like a rake, with short hierarchical lines, or pyramidal, with long lines ([MIN 82], p. 132-145). There is no ideal structure which could satisfy any situation. In fact, organizational structure depends on coordination needs, induced by the processes. Interdependencies between tasks induce small units, while a high level of standardization allows large units.

- Planning and control systems help regulate the organization’s activity and performance ([MIN 82], p. 147-154).

When the parameters above are not sufficient, connection mechanisms are necessary. Mintzberg, once again building on Jay Galbraith’s works ([MIN 82], p. 155-172), identifies seven connection mechanisms: direct contacts between

leaders, connection roles, task forces, teams, integration roles, relation roles and matrix organizations. These connection mechanisms play a critical role with systems of systems, when different organizations have to coordinate and contribute to global processes to provide a service which cannot be attributed to only one of them.

Decision-making system design takes into account centralized versus decentralized organizations ([MIN 82], p. 173-202).

Mintzberg ([MIN 82], p. 203 to 266) differentiates five contingency factors, and develops the following rules:

- Efficiency within organization design.
- Age and size: the older the organization, the more formalized its behavior. Its structure grows more elaborated and more complex, and its units grow larger, as the organization itself grows. The units get further differentiated as the tasks get more specialized, and its administrative component also develops.
- Technical system: with heightened control over the technical system, the work becomes more standardized, and the operational center becomes more bureaucratic. The sophistication of the technical system leads to a more elaborated administrative structure. The improvement of the logistic staff's qualification decentralizes the organization, and requires more connection mechanisms to coordinate their activities.
- Environment: the environment's dynamics match the structure's organic nature; the more complex the environment, the more the structure is decentralized. On the other hand, a hostile environment goes with a centralized structure. If the organization's markets are diversified, the organization tends to split into units based on these markets.
- Leadership: an increasingly powerful external control over the organization leads to an increasingly centralized and formalized organizational structure.

Based upon these nine design parameters and some contingencies factors, Mintzberg identifies five structural configurations. Most organizations are a mix of these five patterns:

- Entrepreneurial organization ([MIN 82], p. 273):
 - "main coordination mechanism: direct supervision;
 - most important part of the organization: strategic apex;
 - main design parameters: centralization, organic structure;

- contingencies factors: youth, small size, a simple technical system, a simple and dynamic environment, management with a hostile or overly strong need of power, or an old-fashioned structure”.

The entrepreneurial organization is not elaborated, its logistic functions and techno-structure are nonexistent or little developed. Work division is imprecise, with little differentiation between units, and reduced managerial staff. This is an organic structure in which behaviors are barely formalized and coordination is based upon direct planning from the strategic apex, usually reduced to a single manager ([MIN 82], p. 274). “Crisis organization – a variant – appears when the environment is extremely hostile, thereby forcing the organization to centralize itself, regardless of its usual structure” ([MIN 82], p. 276). “Centralization presents a great advantage: it guarantees that the operational center shares knowledge before taking strategic decisions. Moreover, it favors the flexibility and adaptability of the strategic answer: a person can make a decision on its own. But centralization may introduce confusion between the strategic level and the operational level” ([MIN 82], p. 279).

- Machine organization (bureaucracy) ([MIN 82], p. 281):

- “main coordination mechanism: standardization of work;
- most important part of the organization: technostucture;
- main design parameters: formalization of behavior, horizontal and vertical specialization, units generally created according to their functions and large in size, vertical centralization, reduced horizontal decentralization, and planning of the action;

- contingencies factors: elderly and very large organization, non-automated technical system, simple and stable environment, and old-fashioned structure”.

Within this configuration, operational tasks are routine and highly specialized, the procedures highly formalized, with many rules, regulations and formalized communication in the entire organization, and these procedures and rules are elaborated by techno-structure analysts. The units are large, and the tasks are grouped according to functions; the centralization of decision making is relatively important, the administrative structure is elaborated, and there is a clear differentiation between operational and functional” ([MIN 82], p. 282). Taylor removed the power to intellectually contribute from the people who worked in the workshop, thereby ruling out initiatives. He conceived the role of human beings as exactly identical as mechanical parts. This destroyed the meaning of activities and the quality of work ([MIN 82], p. 298).

– Professional organization ([MIN 82], p. 309):

- “main coordination mechanism: standardization of skills;
- most important part of the organization: operational center;
- main design parameters: training, horizontal specialization of work, horizontal and vertical decentralization;
- contingencies factors: complex and stable environment, simple technical system, fashionable structure”.

In this context, coordination is based upon the standardization of skills, training and socialization. This configuration recruits experts who implement well defined, standardized procedures which are difficult to learn. “This corresponds to an environment both complex and stable – sufficiently complex to require procedures with a learning curve of several years, but sufficiently stable for these skills to be well defined, and in fact, standardized” ([MIN 82], p. 324). “The technical system of professional bureaucracies is neither sophisticated, nor automated, and does not really regulate activities” ([MIN 82], p. 325).

– Divisional (diversified) organization ([MIN 82], p. 337):

- “main coordination mechanism: standardization of outputs;
- most important part of the organization: middle line;
- main design parameters: units created according to the markets, performance monitoring systems, reduced vertical decentralization;
- contingencies factors: diversified markets (particularly for products and services), elderly and large organization, managers in need of leadership, fashionable structure.

The divisional (diversified) organization is not complete. It links the strategic apex to the operational centre, but supervises other structures, each of them with its own structure and all functions necessary to its operation, and links headquarters to its divisions” ([MIN 82], p. 338). This reduces interdependence between divisions, such as coordination between units, enabling these divisions to function autonomously. “The main coordination mechanism of this structure is the standardization of outputs. The most important design parameter is the monitoring of performance” ([MIN 82], p. 339). “This configuration leads each division to be more centralized and formalized than if they were acting as independent organizations” ([MIN 82], p. 341-342). This configuration induces an evolution towards a mechanical bureaucratic structure. This configuration is adapted to simple and stable environments;

– Adhocracy organization ([MIN 82], p. 375):

- “main coordination mechanism: mutual adjustment;
- most important part of the organization: support staff;
- main design parameters: connection mechanisms, organic structure, selective decentralization, horizontal work specialization, training, units grouped according to functions and markets;
- contingencies factors: complex and dynamic environment, young organization, sophisticated technical systems, often automated, fashionable structure”.

This configuration is adapted to complex and dynamic environments. This structure is highly organic, with little standardization of behavior. Small units gather experts (project teams). The main coordination mechanism is mutual adjustment. It implements connection mechanisms, such as task forces, permanent councils, matrix structures. This structure does not rely on standardized procedures, but is flexible, informal, organic, and renews itself” ([MIN 82], p. 376-377). “Strategy does not ever really stabilize. It evolves continuously, along with the projects” ([MIN 82], p. 387). “For all that, adhocracy does not only feature benefits. Indeed, people have to live with adhocracy’s ambiguities and inefficiencies”. ([MIN 82], p. 398). Adhocracy is “designed to support extraordinary activities” ([MIN 82], p. 401).

From the standpoint of systems of systems, where many different organizations have to collaborate together, it is likely that each organization has elaborated its structure, according to its age, size, activities and environmental constraints. It is therefore likely that they do not share the same design parameters and do not feature the same structural configurations. This makes collaboration between organizations more difficult. When their activities permit it, with the creation of common standards in terms of work procedures, qualification, training, socializing, achieved performance, collaboration may be favored if they feature the first three structural configurations. But this situation induces constraints, reduces their autonomy and their ability to adapt, and is only viable in stable environments. Faced with strong evolutions, structural transitions may help the structures evolve, but this will be met with resistance ([MIN 82], p. 417).

The most appropriate structures for systems of systems are those which promote interactions and collaboration between organizations, implementing mutual adjustment, such as adhocracy, reducing interdependences, such as divisional organization, and the hybrid structure between these five configurations. Environmental and activity constraints help choose between these configurations. In the case of a long process, not very dynamic or interactive, a divisional organization is more appropriate, allowing each organization to keep its own structure and

reducing the impacts of systems of systems partnerships and interdependences. On the other hand, in the case of a dynamic and interactive environment which necessitates constant coordination, the structure must enable mutual adjustment, according to the evolution of the environment. Temporary structures, such as adhocracy ([MIN 82], p. 395), may be punctually created to reply to a project mixing many organizations, or to a particular situation, such as crisis management.

Mintzberg's design principle is based upon a structural point of view and promotes an organizational design. This design principle has to be complemented with other aspects, more informal. We will now look at these aspects.

4.5.2. *Informal and individual dimension within organizations*

Whether small or large, organizations are composed of individuals, each with their own character and their own objectives, who maintain informal relationships with one another. These informal relationships between their agents influence the organizations' operation. These are the two points which we are about to discuss.

4.5.2.1. *Informal relationships*

Organizations have goals, which have been assigned to them by their creators. With time, these goals evolve and shift. The rules that have been implemented to structure the organization's activities take on a symbolic value of actual objectives. "The initial goals of the organization, the objectives for which realization it has been designed, are forgotten" ([ROJ 03] p. 36). The organization's members build privileged relationships, form subgroups, each with its own objectives, and develop a team spirit, which may prevent any change, or completely separate them from their clients or from the users. Moreover, these subgroups and the objectives they pursue can be linked to stakes within or without the organization. Such is the case, for example, with networks of alumni from prestigious schools, and with trade unions. These situations generate conflicts within the organization. Lastly, the organizations' objectives result in the emergence of the social processes' overt and latent functions. The latter are unconscious and unplanned functions, in competition with the overt functions ([ROJ 03] p. 37). For instance, a political party whose overt function is to reorganize the economy, according to its program, and whose latent function is to control, through the party and its members, the State monopoly, and take over the executive positions.

The results of research led into human relationships shows that leaders and informal groups emerge within production teams, in parallel with the management staff. These informal groups, within the organization, have their own social structure, their internal codes and their standards. Morin ([MOR 05] p. 122) shows

that complementary and antagonistic relations are created between the organization and these groups.

Several group levels can be differentiated within an organization; primary groups of people working together, groups of friends, activity groups out of the working place (members of a common association), and groups formed around questions of common interest, and finally, the organization itself, viewed as a whole. Personal and interpersonal elements influence the organizations' operation. Thus "the sympathy mechanism makes us more inclined to fulfill the requests of people whom we know and like. Sympathy itself is often based on similarity: we prefer what looks like us (opinions, personality, environment or way of life)" ([ROJ 03] p. 283).

The organization's goals and the members' goals can differ, in which case conflicts of objectives emerge. Such is the case when directors tend to maintain their positions in opposition with the organization's members. The directors have at their disposal a set of tools (logistic, informational, legal, etc.), out of reach of the organization's members, which enable them to stay in their position of leadership. Moreover, they seek to obtain external leadership resources, which will allow them to reduce their dependence on the members. Organizations can be built in networks; the party controls the health insurance system which finances the party and the party's trade union, etc. The managers can create an ideology to justify their domination, ideology which they use as a weapon against any opposition, stigmatizing members who, in their eyes, are betraying or threatening the organization's safety. They can also build internal factions through favors and privileges, thus ensuring the loyalty of these factions' members ([ROJ 03] p. 168).

The organization, as much as the members who create these factions, can enact standards, influence the members' values and prescribe their behavior through a system of rewards and sanctions. The former system rewards what is considered as correct, satisfactory, and normal. The latter system punishes what is incorrect, unsatisfactory, abnormal, the noncompliance to these standards. The rewards can be symbolic, such as an appointment, a promotion, or substantial, such as bonuses, or raises. The sanctions, too, can be symbolic, such as harassment, or substantial (or rather insubstantial), such as the absence of a raise. The sanctions may also be brutal. These rewards and sanctions can concern the values shared within the organization, as well as the behavior in terms of performance (productivity), but also other dimensions such as clothing habits, or even the employee's leisure activities. The organization's members adjust their behavior and self-image accordingly. These social regulations work in the same way as the regulations we have previously described to demonstrate informal relationships ([ROJ 03] p. 175). Beyond the control of the members' values and behavior, the organization also influences the decisions made by its members. Indeed, the organization controls the parameters and the information which members use to make a decision ([ROJ 03] p. 177). All these

elements orient the decision making process and *in fine* the decision itself. The works of March & Simon, which we will look at in detail in the following part, focus on that point.

Organizations grow more rigid as they grow older. The number of conformity rules increases. The organizations lose flexibility; they get more rigid and more and more impervious to change.

4.5.2.2. *The agents' bounded rationality and the decision-making process*

March and Simon [MAR 58] have worked, in social psychology of organizations, on communication impacts upon making decision processes, and have defined the concept of bounded rationality. This concept facilitates the description of the rational behavior of individuals and groups.

In large organizations, specialized departments work on specific subjects, such as human resources, production, marketing, management. This specialization creates interdependency between organizations, or between specialized departments within one organization, insofar as the organizations and departments must coordinate. The standardization of situations allows for a higher tolerance towards interdependency and facilitates coordination. March and Simon ([MAR 58] p. 157) differentiate two types of coordination, coordination by plan and coordination by feedback. "The more stable and predictable the situation, the greater the plan's reliance on coordination; the more variable and unpredictable the situation, the greater the reliance on coordination by feedback." Indeed, "to the extent that contingencies arise, not anticipated in the schedule, coordination requires communication to give notice of deviations from planned or predicted conditions, or to give instructions for changes in activity to adjust to these deviations."

The organization's internal communication channels, necessary for coordination, use standardized facts by classifying situations, and enable the selective distribution of information. This classification takes into account the main characteristics of the object or the situation, without reproducing their complexity ([MAR 58] p. 165). The individuals focus their attention on objects which correspond to their patterns of reference, well established and confirmed. The perceptions which differ from these patterns are filtered before they can reach the conscience, and reinterpreted according to these patterns of reference. Parameters monitor the members' perceptions depending on their position within the organization. These pieces of information are simplified, standardized and oriented. These patterns are used to reduce uncertainty. This enables a stable and durable decision making process, avoiding an excessive workload, reducing the resources needed to communicate, and allowing for a simplification of organizational responses.

Individuals and organizations prefer choices which favor the continuation of existing programs and avoid choices which represent changes. Inaction does not ask for resources: it allows for the preservation of energy and money. On the other hand, changes require action, resources, energy and money, and require difficult decisions to be made.

Since the first works of March and Simon, many experimental works in cognitive psychology have demonstrated the diversity of these simplification strategies, most often grouped under the heading of bias and heuristics [SHE 84]. These works by March and Simon, as well as the works on heuristics and bias are taken into account in sensemaking, which we will take a closer look at in “Sensemaking in organizations and mutual intelligibility”.

Within the context of systems of systems, what can be taken from these works, in unplanned situations, is the need for, on the one hand, reactive coordination, and on the other hand, the standardization of facts. This standardization of facts leads to a simplification which disregards elements outside of the reference framework and reduces the ability to detect weak signals outside of the framework.

Works must be led in order to understand how patterns, which are simplification processes reducing the workload, bettering the coordination between groups and organizations, and an increased alertness, apt to perceive weak signals outside of the framework, can be simultaneously implemented.

4.5.2.3. *Collective decision making within small groups*

Collective decision making more precisely concerns complex systems, which cannot be understood by a single person, but does not pertain to group dynamics or consensual decision. In the field of distributed decision making, works have been led on tactical reasoning in crisis management, or in the case of emergencies [RAS 91].

In crisis management, operators face ever changing risks and situations. Their objective is to reach a stable state by curbing, as quickly as possible, the consequences on the persons, goods and economical environment. These crisis situations require coordination by feedback. In emergency situations, studies have shown that, if the internal cohesion, intra-organizational, is rather good, the cohesion between organizations, inter-organizational, is on the other hand rather limited. Communications and the absence of clarity in interactions result in poor cohesion. It is necessary to take an in-depth look into the factors which reduce the cohesion between organizations, and the means to remedy it.

4.5.3. *Internal and external environment of organizations*

The structure of organizations, of their processes and their activity, depends, on the one hand, on the organization's internal environment, in this case the technology implemented to produce the organization's products or services, and on the other hand, on the organization's external environment, in this case the suppliers, clients, oversight entities, in a dynamic and interactive fashion. These are the components we are about to analyze.

4.5.3.1. *Technology, organizations' structuring basis*

Thompson ([THO 04] p. 15-18 and [ROJ 03] p. 134-135) defines three types of technology:

- Long-linked technologies are based on sequential operations, assembly lines for example. The complexity of this kind of technology stems from the necessity of ensuring the regular and reliable following of the steps.
- Mediating technologies establish links between independent individuals, for example insurance companies or communication agencies. Complexity stems from the need to follow standardized procedures and criteria. Indeed, despite the diversity of cases, treatment categories have to be established.
- Intensive technologies: the various techniques used to generate a change in an object, their selection, combination and order of application are determined by the object's feedback.

An organization featuring intensive technology will try to obtain as much power over objects, activities, methods and tools as possible.

Moreover, Thompson ([ROJ 03] p. 136-137) defines three types of interdependence:

- pooled task interdependence, where entities enjoy common resources, such as trading groups;
- sequential task interdependence, such as assembly lines;
- reciprocal task interdependence: the output of one is the input of another, and vice versa. Each unit creates a contingency situation for the other.

These three types of interdependence form a continuum. All organizations feature pooled task interdependence. The most complicated organizations also feature sequential task interdependence. The most complex add reciprocal task interdependence to the two previous features.

Links exist between the types of interdependence and the means of coordination. Pooled task interdependence calls for coordination through standardization, while sequential task interdependence and reciprocal task interdependence requires coordination through mutual adjustments. The latter requires communication between the individuals and the resources for the decision making. When the interdependences are complex, coordinating them and managing the organization becomes more difficult and more expensive. Moreover, the causal relations are no longer manifest and become hard to express and master.

Finally, the types of technology are linked to the types of interdependence. Mediating technology is linked to pooled task interdependence, long-linked technology to sequential task interdependence, and intensive technology to reciprocal task interdependence ([ROJ 03] p. 138-139). Thus, each organization must meet the proper requirements, induced by the technology it implements and by the task environment which it belongs to. In crisis management situations, most organizations implement intensive technologies and reciprocal task interdependence.

From our standpoint on systems of systems, when organizations are brought to cooperate, collaborate, the cooperation capability is largely influenced by the processes and activities led by each organization, which vary according to the technologies and interdependence types.

By linking the works of Thompson with the words of March and Simon, for example in the context of crisis management, where, as we have seen, organizations implement intensive technologies and reciprocal task interdependence, people are trying to standardize the facts, implement schematics, favor coordination between the organizations' groups. The implementation of schematics reduces the need for resources which was induced by the reciprocal task interdependences. One important parameter, identified above, is the level of uncertainty of the applied technology. Another cause for uncertainty lies in the organization's external environment.

4.5.3.2. The external environment influences the organizations' structure and operation

This environment, in which the organization is situated and with which it interacts, is social, economical, political, cultural, legal, and includes other organizations, such as suppliers and customers, oversight entities, state institutions, etc. A company's economical environment includes its suppliers and customers, while an association's includes its members and, less directly, an administration's includes its ratepayers.

To this are added hazards which are specific, or not, to the economical environment and which, indirectly, impact the organization's operation and

structure. Stock market crashes, oil crisis, stresses of weather and natural disasters, the rise of steel and ferric matter, the exchange rates, are as many hazards, cause for uncertainty, since it is impossible to take them into account ahead of time.

Such was the case with the BSE crisis (mad cow disease). In France, the consumption of beef decreased considerably, thus putting in jeopardy the economy of the catering companies specialized in that field. Moreover, the many dimensions of the environment, political, economical, legal, etc., are not separate from one another. The publication of satirical cartoons of Mahomet in Denmark raised a general outcry in the Muslim world. The ethical consequences of these cartoons are important, launching a debate on the boundaries of freedom of speech. Reactions range from boycott to the violent arsons of consulates in the Near-East and the deaths in Pakistan and Libya. Boycott has economical consequences that are not negligible for the enterprises subjected to it. Indeed, this boycott of Danish products within the Muslim world cost around 134 million euros to the companies. Through side-effects, the boycott may have social consequences, such as a decrease in activity and short-time working.

The important technological evolutions in computer sciences, in hardware as well as software, which were at first remote from the field of photography, have turned the photography market upside down in a few years. The economical and social consequences are very important. Kodak had to cut between 12,000 and 15,000 jobs, namely 20% of its workforce. The global enterprise, its structure and its operation are affected.

4.5.3.3. Causal relationships within the context of organizations, and their impacts in terms of engineering

Works on organizations, led within a scientific approach to work or bureaucracy, were based on a closed, determined system logic, in an organization shut off from its external environment ([THO 04] p. 4-6). Moreover, they assumed that there was only one efficient organizational pattern, a pattern which had to be aimed at.

The internal and external factors, described in the previous sections, show that this first model presents important limitations. The organization is not a closed system. The action of the organization towards its environment modifies the latter. The environment, progressing in a determined way and evolving from the organization's actions, modifies it in return. As demonstrated by Thompson ([THO 04] p. 10-12), both points of view must be taken into account.

Situations of nonlinear relationships, including positive or negative feedback loops, are described in terms of interactive complexity by Perrow ([WEI 95] p. 130). Crisis situations appear when unlikely events happen simultaneously. Moreover, if the performance requirements are higher, the individuals focus their attention on the

central aspects of the task affecting the performance, and neglect the peripheral aspects, the weak signals. Crucial information on the interactions between the task's components risk being forgotten, misunderstood or ignored ([WEI 95] p. 130).

As for Morin [MOR 05], he describes this phenomenon in terms of eco-self-organization: the organization self-organizes in close interaction with the environment; they evolve together.

This situation of circular causality has important consequences in terms of systems engineering and systems of systems engineering. Indeed, systems engineering holds the hypothesis that the environment is *underlyingly* stable. The links between the system and its environment are described in terms of interfaces, on a structural basis, rather than in terms of interactions, from a dynamic perspective. Moreover, during the design of a system, it is particularly difficult to plan all the interactions between the system and its environment, all of the system's impact on the environment and all of the environment's impact on the system.

Within this context, two essential aspects must be treated. On the one hand, in terms of systems engineering, it is necessary to specify and design a sturdy technical device, apt to support interactions with its environment, interactions which cannot *a priori* all be defined and some of which are produced by unlikely events. On the other hand, in order to take the organizational aspects into account, it is necessary to enrich the means, methods and tools of the adequate systemic concepts [LEM 90, MOR 05], as well as of the major works on organizations [MAR 58, PER 84, ROJ 03, WEI 95].

4.5.4. Professional, organizational and national cultures in organizations

Culture, as defined by ethnologists, has an important impact on the organizations' operation. "Culture fashions a complex framework of national, organizational and professional attitudes and values within which groups and individuals function... [This is a] natural and unquestioned mode of viewing the world" ([HEL 98] p. 1).

Studies show that, in the aviation and medical fields, many human errors "involve failures in communication, decision making, interpersonal conflict and teamwork" ([HEL 98] p. 17). The differences in team performance depend on the leaders' personalities. The cultural and organizational factors which are characteristic of the company influence the organization's performance. In the aviation field, many accidents concerning Western technologies arise from misunderstandings about the English language, unfriendly human-computer

interfaces, and difficult human interactions between people coming from Western countries and people coming from other countries.

4.5.4.1. *Professional culture*

Professional culture is built on standards, values, attitudes specific to each profession. The members of a profession all have special expertise. The process of acquiring such expertise requires the novice to undergo a lengthy, demanding training, a socialization process during which the new members are indoctrinated into the professional culture.

The profession's standards and values are showed as examples by the older members to the newer recruits, who take them over and, in their turn, while they gain experience, communicate them to even newer members. The profession's culture is manifested in its members by their sense of community and their commitments. The individuals, who go through a severe initiation in order to become a member of a group or an organization, reduce the cognitive dissonance by proving that the group they are joining is worth their trouble. "The positive aspects of professional culture, including prestige, contribute to a positive self-concept in the work domain and to self-esteem" ([HEL 98] p.33).

On the other hand, negative aspects, including an impression of invulnerability, also become an integral part of the self-concept. "The resistance of self-concepts to disconfirming evidence can explain why attitudes about personal limitations seem to fall on deaf ears and why change proceeds at a slow pace" ([HEL 98] p. 33). "Many of the errors and conflicts observed in the operating room are at the point of intersection between different groups" ([HEL 98] p. 40).

4.5.4.2. *National culture*

Work behavior can be influenced by elements which do not pertain to professional standards. Cross-cultural psychology studies phenomena such as leadership, communication, risk perception, stress, decision making. These phenomena can have relevance for team performance. The efficiency of leadership can vary from one culture to the next. The same leader behavior which can be seen as rude and inconsiderate in one culture may be interpreted as paternalistic and encouraging in another ([HEL 98] p. 55).

The preferred communication style in some countries is direct and specific ("say what you mean, and mean what you say") while in others it is indirect and relies on the context to carry its full meaning. This is the case when saying "yes" just means "I've heard you", and the expression of disagreement and discussion are socially repressed ([HEL 98] p. 55). The modes of conflict resolution also differ from one culture to the next, from avoidance of conflict to the search of compromise, to

confrontation ([HEL 98] p. 55). Differences in power distance can lead to hostility. In some cases, this can lead to a break in communication and team work.

4.5.4.3. *Organizational culture*

Helmreich and Merritt ([HEL 98] p. 109) define “organizational culture as the values, beliefs, assumptions, rituals, symbols and behaviors that define a group, especially in relation to other groups or organizations.” “Each organizational culture is unique and socially constructed” ([HEL 98] p. 110). “Cultural strength relates to whom and how many accept the dominant values, how strongly or intensely the values are held and how long the values have been dominant” ([HEL 98] p. 111).

This discipline looks at questions such as “do people’s perceptions of their organization and Management affect their performance?” ([HEL 98] p. 116). “A demoralized and cynical pilot group which believes that Management will compromise the crews’ safety for profit [...] and that their own suggestions for improving safety will be ignored by an uncaring Senior Management” presents a “greater willingness to deviate from company mandated procedures, even when those procedures are designed to maximize safety.” ([HEL 98] p. 116-117) “Management can direct cultural shift by specifying the desired actions and reinforcing the appropriate norms, but the efforts of Management in this direction must be sincere. While it may be possible for Management to direct people to change their work behavior, it cannot direct people to change their values. And without the underlying values and beliefs in place to guide the behavior, any Management-directed behavior shift will be short-lived. When employee groups feel that they cannot trust Management [...], they will reject with suspicion any new initiatives, including training and new safety procedures” ([HEL 98] p. 124-125).

Perhaps the most difficult cultural challenge for any organization is unifying the values, beliefs and practices of employees from merging companies. [...] ‘The way we do things here’ is no longer the same ‘here’, and members from both organizations struggle to resist the changes. [...] An integrated organizational culture can be characterized by subgroup cooperation, a strong corporate identity, a positive organizational climate and high employee morale, all of which have a positive impact on service and safety” ([HEL 98] p. 121-122).

4.5.4.4. *Integrating organizational, professional and national cultures*

“There can be a variety of interrelationships among national, professional and organizational cultures which are ultimately reflected in the behaviors at the ‘sharp end’. National culture is the most distal element of the model and the one least amenable to change. It can influence the organizational culture, for example in the forms of communication and leadership practiced” ([HEL 98] p. 134).

4.5.4.5. *The influence of culture in the apprehension of uncertainty*

A study [SUT 04] on 44 officers performing support and stability operations in Bosnia-Herzegovina helps determine the contribution of culture and personality to cognitive readiness and response to uncertainty. These 44 officers form a multicultural team, from Canada (3), Germany (4), France (2), the Netherlands (4), Italy (2), New-Zealand (1) and the United States (23), teamed in three groups, English, Germanic and Romance, implementing the function of control and command. This support and stability operation pertains to asymmetrical threats, from peacekeeping to violence management, and is characterized by the growing use of information technologies and increasing uncertainty. There are, at the same time, individual differences in the need for certainty, and intercultural differences in terms of avoidance of uncertainty.

The study is about the following main dimensions:

- the type of personality, featuring five scales of reference (activity/energy, aggression/hostility, sociability, neurosis/anxiety and impulsiveness/risk taking);
- individual preferences for using cognitive structuring to reach a higher level of certainty;
- the individual ability to implement cognitive processes compatible with the need for cognitive structuring;
- individual differences in the response to uncertainty, featuring the three following responses: emotional responses (anxiety and sadness), cognitive responses (based on order, planning and structuring), and the desire for change (individuals like new things, change and what is uncertain).

The individual preference for using cognitive structuring in order to reach a higher level of certainty does not only depend on individual differences, but also on cross-cultural differences. The most important difference in this need for a higher level of certainty and planning ability is found between the English and Romance groups. The first group presents a higher score than the second one concerning the individual preference for using cognitive structuring in order to reach a higher level of certainty. In such a context, we could surmise that the members from the United States, which were predominant in the English group, have a greater need for certainty in decision making situations, and prefer the use of cognitive structures to achieve a higher level of certainty. On the other hand, the Romance group presents a low level of individual preference for using cognitive structuring in order to reach a higher level of certainty. This group presents more relaxed behavior, and uses a higher number of stereotypes to reach a higher level of certainty. Finally, while the English and Romance groups display a preference in working with other people, the German group displays self-confidence and a firmness of spirit.

“The cognitive facility to react appropriately to an uncertain environment is critical for the mission’s success, especially in situations within which the exchange of information, the definition of roles and responsibility, the coordination and supporting behavior characterize the work of multicultural teams.” [SUT 04]

This study suffers from its lack of participants and the overwhelming presence of members from the United States, and should therefore be confirmed and broadened. However, the results show the importance of individual and cross-cultural differences.

4.5.4.6. *Culture and technologies*

Technologies are introduced into organizations which have their own culture, their own ethical values, depending on the members which compose them. They are not fixed and evolve with the social, economical, political and cultural context of their home countries.

Kirke [KIR 04] shows the consequences of the introduction of an aerial drone system within the feminizing British army. When the drone system was designed, there were no women within the Royal Regiment of Artillery. The drone system features a ground data terminal, heavy to manipulate and operated by a team of two people, half a mile away from the ground control station.

This team is composed of men, since women cannot manipulate the ground data terminal. The same thing applies to the recovery of the drone system’s flying part. Women take care of the less arduous missions of logistic and control within the ground control station. This situation could not be anticipated during the system’s design. The distribution of activities between women and men depending on their physical capacities does not comply with the military custom which demands that the staff rotate and perform different tasks, in order to get a good grip on the trade. Moreover, this distribution of activities between men and women threatens the ethics of the army concerning the sharing of burdens.

[KIR 04] differentiates two processes implemented whenever deeply rooted attitudes, norms, values, are stirred by a forced change: cultural drag, and cultural precession. Cultural drag is the process implemented when the culture of an organization is not structured to quickly and easily adjust to new conditions in the environment. In that case, the organization’s members tend to adapt to a slower rhythm than the one expected.

Within cultural precession, the organization’s members move towards different directions, unplanned or unexpected. Kirke analyses the way the introduction of new technologies, particularly information technologies, for the implementation of network-centric operations, overturns the techniques and tactical uses of the battle

field, and their consequences with regard to military customs. Those customs pertain as much to power and prestige, as to the cohesion of units, formal or informal, the feeling of belonging, ideas, rules, behavioral conventions, written or unwritten.

The implementation of capabilities made possible by information technologies has consequences:

- a very strong increase of available information;
- the ability to quickly reorganize an operational team of command and control, for example to facilitate the *reach back*¹ of information to the headquarters, or even to another continent, which allows commandment to keep in close contact with the operational or tactical staff, while being physically remote;
- the ability to reconfigure responsibilities for the function of command and control, meaning that the best placed commanding unit can be given the reigns to the battle, regardless of its rank and space of responsibility;
- the increase of the maneuver's rhythm resulting from heightened capabilities.

In this context, Kirke [KIR 04] draws the following observations, in terms of cultural drag and cultural recession processes:

- individuals will not profit from the new abilities to distribute the information;
- virtual teams will not come into existence, because of a lack of trust between members who have never met and do not know each other;
- the ability to implement a function of command and flexible control will be subordinated to the cultural surmises about the hierarchy's legitimacy;
- the rhythm increase will not be efficient because the senior command will not be in the right psychological conditions to exploit it, and the junior command will refuse to make high level decisions.

He goes on to make the following suggestions:

- make sure that the data flow is designed to take into account the social structures of the army, among others, the relationships of hierarchy and the informal relationships;
- allow the members of virtual teams to meet up before deployment;
- design new, more flexible, procedures of command and control, within the existing hierarchical relationships;

1. *Reach back*: the ability to access information backwards, the use of technologies which enable commandment to access information away from their location.

- strengthen the headquarters with a senior rather than junior staff.

4.5.4.7. *Synthesis of the cultural aspects in the context of systems of systems*

From these works, we learn that the organizational, professional, and national cultures have an influence on the mode and the expression of leadership, the communication modes and the ability to collaborate. This is the result of a process which integrates trust, personal and collective identity and the morale of the individuals.

The processes of cultural drag and cultural recession help us understand the reactions to the introduction of new information technologies, among others, within network organizations.

As we pointed out in the definition of systems of systems, their design and their operation call into play different organizations which can represent different organizational, professional and national cultures. Peacekeeping operations done in coalition, just as well as multinational interventions to face natural disasters, call upon systems of systems and the intervention of different organizations.

4.5.5. *Sensemaking in organizations and mutual intelligibility*

After having treated the cultural dimension of organizations, we will now study the notions of sensemaking in organizations, and mutual intelligibility. These notions address, among other things, the way a group elaborates a shared representation of a situation.

4.5.5.1. *Sensemaking in organizations*

Within an organization, people attribute meaning to the situations they are faced with, and give structure to what was previously unknown to them. This meaning helps them understand their environment, and act upon it. This sensemaking process is characterized by the following elements [WEI 95]:

- someone notices “something”, in an ongoing flow of events, something in the form of a surprise, a discrepant set of cues, something that does not fit;
- this “something” is understood retrospectively, after the break in the flow of events. Everything which might affect memory and recall shapes the meaning which is given to that “something”. The past is reconstructed with the knowledge of this “something’s” consequences. This reconstruction varies depending on whether the consequences are favorable or unfavorable. Therefore, things have never happened exactly as in the memory we have of them;

- plausible explanations are offered to explain the reason for this “something”. Cognition and action go hand in hand. The apprehension of this “something” is not contemplative. The environment within which these events occur is not fixed, and its existence is not remote from the people. The actors help create this environment, with which they are faced, in which they are located. In return, their actions are restricted by the environment. These processes and activities are interdependent, and the change is continuous;

- these explanations are communicated, made public. This “something” is then turned into an object, therefore the “noticing”. It becomes the subject of communication, involving social phenomena such as trust, distrust, lies, gossip, loyalty and commitment;

- the sensemaking process is a continuous process. The interruption of people’s daily activities attracts attention, generates emotions. These emotions, in turn, disrupt the cognitive process. People with improvisation skills display fewer emotions, and their emotions are less extreme;

- sensemaking is built on a reduced set of elements. The characteristics taken from the situation are made to represent the global situation. Moreover, those characteristics depend on the context. The situation’s outstanding, deviant, unpleasant, extreme, unusual characteristics are favored. We find there the characteristics of the schemas [MAR 58, SHE 84];

- sensemaking relies on sufficient, plausible, convincing and consensual information rather than on precision and accurateness, once more within a logic of bounded rationality [MAR 58].

Weick [WEI 95] presents concrete cases, among others, in the context of command and control systems. These systems connect people who look at things in different ways. At a high level, people use strategic thinking and weigh the risks in their holistic properties. At a low level, a local level, people are more tactical. In this case, audacity and the element of surprise are crucial. The points of view on each level differ drastically, just as the readings of the same events. This creates confusion, which does not stem from ignorance, from a lack of information which would lead to a lack of meaning, but from the way various people attribute various meanings to the same event. Moreover, the meanings may contradict one another.

In such a context, confusion cannot be resolved with more information but with a different kind of information, built within a face-to-face interaction. This interaction helps resolve confusion, insofar as people may debate, clarify or promote their points of view, rather than just provide a heap of information. Moreover, this interaction helps build trust between people, and reduce uncertainty. The richness of the face-to-face relationship helps the perception of complex events, namely

invention and innovation, something which is much reduced when the interactions go through computer systems.

Moreover, Weick ([WEI 95], p. 176-177) differentiates between mechanistic and organic structures. The first are characterized by a rigid framework and the implementation of routines. While the second feature a flexible structure, better able to accommodate those instabilities. Mechanistic structures may be implemented within a stable environment. However, in an unstable environment, in which the routines are inadequate and decision making pertains to problem solving, organic structures are more adequate. We recognize the properties of bounded rationality [MAR 58], and the acknowledgement of contingency [THO 04].

Weick ([WEI 95], p. 177-178) then highlights the growing links between sensemaking and information technologies.

“Existing programs tend to focus on what is judged *a priori* to be “controllable”, which means that information needed for improvisation, reframing, or repunctuation is not available. The observer is trapped into the conclusions coerced by the technology and has neither the time nor the data to question or override what appears to be a compelling synthesis” ([WEI 95], p. 178).

Indeed, information technologies are based on a rationality of decision, not on a rationality of action, or a rationality of narration. This situation increases the probability of interactive complexity and normal accidents, such as described by Perrow [PER 84].

Finally, Weick ([WEI 95], p. 188-189) details the mechanics of sensemaking. If each experience is singular, sharing meaning becomes difficult. Therefore, individuals do not share meaning, but experiences, actions and activities in common and moments of conversation. If they use personal labels and categories, the meaning will be harder to share. If, on the other hand, they use common labels and categories, the meaning can easily be shared. A common adventure is an excellent way of building a team. It is a new, common experience, for which predefined labels or categories do not exist. The individuals will therefore build a shared meaning in order to transcribe a shared experience.

Within the application of sense-making to the context of systems of systems, Weick's works can be articulated around the elements we have previously studied. We recognize the notion of schemas, already understood with the works of March and Simon, works on which Weick builds his study, thereby enriching them. We also recognize the confrontation of differing points of view which do not get resolved through added information, but through a trust built within face-to-face

interactions. This notion of trust is just as important for Helmreich and Merritt [HEL 98] as for Kirke [KIR 04, KIR 05]. Finally, we can see the works of Thompson [THO 04] and Perrow [PER 84] on organizational structures and decision making.

4.5.5.2. *Mutual intelligibility of the situation*

This issue, introduced by Weick, is treated in a slightly different manner by Grosjean [GRO 05].

The mutual intelligibility of the situation, also called situational awareness is, on the level of human factors, a key-element of the network-centric operations which we will discuss shortly.

Within one team, the operators can communicate through speech (interaudibility) as soon as their surroundings are a little loud, and they are close or getting closer. They may communicate through gestures, postures, facial expressions (intervisibility), as long as they can see each other, something which depends on the spatial disposition of the various stations. Communication may be intrusive (an operator hails another) or unintrusive (an operator speaks to no one in particular, or points out something with his hand and thereby attracts the attention of others to the element). These exchanges help them publicly define what must be seen and heard, and the categories of the situations, environments and elements. In this way, they also define the situation as “a common interpretation framework which does not exist outside of this process. This is the basis on which are built mutual intelligibility and the sharing of meaning” ([GRO 05] p. 84).

A lot can be learned from Grosjean’s study [GRO 05] on the command and control station of the A line of Paris’s RER network (a network of suburban railways). Grosjean demonstrates how the operators build a mutual awareness of a situation, and one origin of possible misunderstandings. The spatial layout influences the operators’ access to information, but also their ability to communicate and make their behavior understandable, meaningful to others ([GRO 05] p. 88). If some operators have access to dedicated sources of information which make them autonomous, an operator usually depends on the others to gather necessary information ([GRO 05] p. 91-92). This spatial layout and the availability of information differ according to the corporate culture and the priority defined for each operator’s activities ([GRO 05] p. 87). Lastly, each operator pursues his own specific activity, on a specific object, and both activity and object are the center of his focus ([GRO 05] p. 92-93).

Misunderstandings occur when the operators communicate on a common situation without the same interpretation framework. The same problem occurs when the agents are working on diverging activities and their only way to

communicate on their progress is to interfere with the other's activity ([GRO 05] p. 94).

Grosjean demonstrates that, on London's Bakerloo Line, the co-presence of the operators "allows them to build a common situation within which each one may pursue their activity without having to interrupt the other" ([GRO 05] p. 93).

In the same perspective, Swain and Mills [SWA 03] demonstrate how team members communicate with each other in order to elaborate and maintain a representation of the situation. They use implicit communication strategies when faced with a high level of stress. This requires them to have the same knowledge about the events happening around them. These implicit communication strategies help reduce the communication and coordination load within the team. These authors demonstrate that teams which are used to working with one another have less difficulty implementing implicit communication strategies in new situations than newly formed teams.

4.5.6. Impacts of the introduction of information technologies within organizations

The introduction of the latest information technologies within organizations has had important impacts. These new technologies modify the sources and flows of information. Those who have access to such technologies treat the information and therefore achieve greater power. For people who already have power, this increases it. On the contrary, traditional management, which doesn't have a high command of such technologies, loses power to the ones who have. Those technologies thus take the place of political instruments [VAS 90].

Moreover, the introduction of highly structuring computer systems, such as integrated management software, greatly modifies the distribution of tasks and the organization of work. This introduction goes hand in hand with works which aim to describe, standardize and model the organizational processes, and design activity flows (the famous workflows). The organization's structure is modified so that it corresponds to the structure of the integrated management software, regardless of the organization's former structuring components.

Moreover, the introduction of such technologies, especially those of cooperative work (CSCW: computer supported cooperative work), greatly modifies the communication between individuals. Through the use of messaging systems, communication is less personal, but also less respectful of social norms. Decrease of the social feedback, present in face-to-face communication, such as winking, nodding, smiling, generates difficulties to understand each other and coordinate

communication and common actions. Moreover, this type of impersonal relationships penalizes extroverted people, who are more eager to have personalized relationships [VAS 90].

4.5.7. *Network-centric organizations*

Under this heading, Rojot ([ROJ 03] p. 314-317) studies organization networks and interorganizational structures. This enables the pooling of resources, such as the research and development process. Relational coordination must be implemented, based on interpersonal trust relationships, and those relationships must be established in time, thanks to a mutual learning process. This enables the development of interdependence dynamics between the parties. Quoting Wacheux, Rojot lists the following common rules:

- get personally involved;
- allow time for training;
- respect your partner to achieve mutual trust;
- accept sacrifices to create a non-zero-sum game;
- contractualize;
- take the partner's problems during evolution into account (flexibility);
- make sure the partner has the same expectations;
- develop off-work relationships with counterparts;
- accept differing reactions from the partner;
- respect the partner's interests and independence;
- get the partner to endorse every decision, even if this turns out to be a tactical maneuver;
- design a common-marketing plan.

Each participant within the network has his own objectives, be they economical, politic, scientific, etc. A partial common goal must be discovered and formulated so the network can digest it and constitute itself. This calls for the design of a collective representation, based on a minimum temporary partial point of convergence, of simplification, which will help translate the various points of view. The elaboration of this collective representation is influenced by many elements which are external to the issue of network construction. These elements pertain to the main currents in disciplines and methodologies, which are widely acknowledged and accepted, what is politically correct and ethically acceptable.

Dagnino [DAG 04] demonstrates that the enterprises do not do everything on their own and are deeply rooted in a relationship network weaved between them. This enterprise network is based on the exchange of resources and capabilities. This system often emerges, unexpectedly, from spontaneous interactions between enterprises. Within this network, creation, transformation, changes, cannot be completely planned and governed. Such a thing requires the ability to operate flexible adjustments, the emergence capability of unplanned behavior, which presents unexpected consequences. Indeed, the system is continuously shaping and reshaping itself, within a process of self-organization and self-design. The system, which is a network, never displays any finite boundaries, nor any predefined evolutionary paths. It presents the same behavior as an open system and of organizational closure, in a meaning akin to Varela's operational closure [VAR 88]. It is therefore in coevolution with its environment. This notion is akin to Morin's notion of eco-self-organization [MOR 05].

The network builds on:

- intense social interactions between individuals, teams, groups, enterprises;
- exchange of knowledge which help the elaboration of a shared knowledge reference table;
- deeply rooted commitment to the network's establishment, to work together and contribute to the entire network's performances.

The network's enterprises' integration patterns show that the enterprises differentiate themselves, each taking on a specific role within the network. The value elaboration chain and its processes are not located on the level of the individual enterprise anymore, but on the level of the network.

In the context of systems of systems, this notion of organizational networks is important, and helps demonstrate the implementation and the operation of a system of systems, which calls on several organizations. The organizations might have various countries, various cultures, and various trades.

4.6. Social sciences implemented within the context of systems of systems

The notion of system of systems finds its largest echo within the military field. This field also provides examples on the application of social sciences to systems of systems.

The systems of systems are implemented within network-centric operations (NCO, also called network-centric warfare). These network-centric operations favor the use of new information technologies on the various military levels, in order to

develop a common operational picture, or even a common relevant operational picture. To avoid any confusion, ambiguity or misunderstanding, the various operators who work on the various levels of the chain of command must be able to communicate by building on a common context, a common reference, a common image of their environment.

Creating this common image is not only a question of displaying the same representation on each operator's computer screen, or making intensive use of both instant messaging and formatted emails. A tactical operator does not need the same information as an operational operator. Within a command center, the various operators have various tasks, specific objects of focus, and end up coordinating and sharing information in order to elaborate a common image, a shared awareness of a situation. Within a command station, the communications, broadly speaking, of the operators, are influenced by many factors, as pointed out by Grosjean.

Our first example will present the impact which information technologies have on network-centric operations, while the second will study the network-centric operations conceptual framework.

4.6.1. *Impact of information technologies on network-centric operations*

In the context of network-centric operations, the impact of new information and telecommunication technologies is very important.

Baker [BAK 02] identifies the following elements:

- Tactical representation: nowadays, the assets of defense systems are not the weapons, but the sensors. To achieve a better tactical situation, the appropriate sensors must be used. In return, the use of sensors heightens their vulnerability and the vulnerability of the global defense system. The use of sensors must be optimized while taking vulnerability into account.

- Information overload: this refers to the human mind's inability to take into account and absorb all the information, especially in the case of a network information system. This is further heightened by the necessity of reaching a decision within a minimum delay in order to treat the undifferentiated flow of information compiled by all the sensors. This information overload lengthens the decision making delays, leads to incorrect decisions, or sometimes to the absence of any decision. This echoes the works on bounded rationality (March and Simon [MAR 58]), which we have previously studied, and mutual awareness (Grosjean [GRO 05]), but also on sensemaking (Weick [WEI 95]), as well as the limits and constraints induced by such technologies, limits and constraints which are propitious

to the emergence of interactive complexity and normal accidents [WEI 95], [PER 84].

- The collapse of communication lines and micro-management: the tendency of the operational commander to directly communicate with the tactical units, bypassing the intermediate commanders, and to focus his efforts on the tactical level. This translates into the constant temptation for the operational commander to command through directives rather than through intention or planning. This may confuse the tactical operators and the chain of command. The ease with which the operational command can access and control the subordinates' actions through an information system has been highlighted in many studies. This leads to micro-management: by focusing on the tactical image, the operational commander diverts his attention from the operational and strategic levels he should be monitoring. Moreover, the operational commander is not trained, and does not master the latest technologies, techniques and procedures of the tactical level; hence a major risk of double inefficiency.

- The loss of autonomy: the increasing use of technologies is heightening people's dependency on technologies. In this respect, the systems' installation manuals do not provide any downgraded information if the system is not available, breaks down or is neutralized by a combat action.

- Endurance: if the network-centric automated systems may operate without interruption, the human commander who uses these systems, on the other hand, can only operate periodically. The human mind cannot stay focused on an object indefinitely without losing some of the capacities needed to reach efficient decisions.

To these problems, Baker [BAK 02] offers various solutions. Among which are the following:

- Improved information presentations: the presentation of information (as tables, in opposition with the diagrams) influences the decision. Nowadays, the systems are designed within a "design then train" approach, which forces the operators to adapt to the system. Numerous works show that the simplification of the presentation reduces the cognitive load.

- Information filtering and clustering: most of the cognitive load comes from unnecessary and non-relevant information, which are communicated to the operational commander. By their design, the information systems provide all the information to every network member. The operational commander wades through all the available information to find the pieces which actually contribute to the elaboration of the operational situation. While the common relevant operational image aims to reduce this representation to the sole relevant information, no guide or standard exists to help create it. Some information, relevant on a tactical level, such

as available weapons, must be filtered past the operational command, for such information does not hold any relevance on that level.

In this context, the works of Grosjean [GRO 05] and Sperber and Wilson [SPE 89] take on a particular importance. The filtering and clustering of information must be designed while taking into account the use the operators will make of the information (what are their purpose, who are they useful to, how are they used, etc.), but also the totality of the context known to the operators, context on which the operators lean to understand this information and reach a decision. This context is enriched with the operators' experiences, their beliefs, their expectations. It is this context, among other things, which helps us to understand the tactical situation, elaborate an image which is common to all the operational commanders, and finally reach a decision.

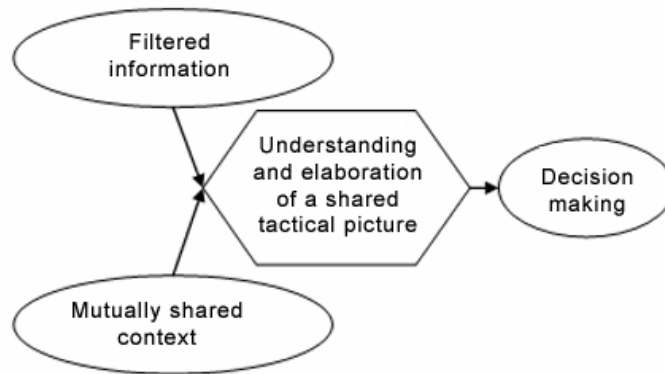


Figure 4.1. *Process leading to decision making*

Finally, from a different point of view, studies are currently led to elaborate common pivotal ontology, which would enable different applications to communicate within a service-oriented architecture [SMA 05]. It would seem opportune to merge these technical studies with the ones led in the field of human factors, including works by Weick [WEI 95], Grosjean [GRO 05], Kirke [KIR 04, KIR 05].

4.6.2. Example of the network-centric operations conceptual framework

The document NCO conceptual framework [EBR 03] demonstrates which aspects of social sciences should be taken into account during the implementation of network-centric operations.

The main stakes are:

- the sharing of information;
- collaboration;
- the quality of information, including its ability to be communicated and shared;
- the mutual intelligibility of the situation, the shared awareness;
- the ability to mutually synchronize;
- the ability to promptly command.

Ultimately, this ought to improve the level of efficiency, the synchronization of the groups' actions and the synchronization of decisions.

In the context of network-centric operations, the document identifies four domains, which are:

- “the physical domain, where strike, protect and maneuver take place;
- the information domain, where information is created, manipulated and shared;
- the cognitive domain, where perceptions, awareness, beliefs and values reside and where, as a result of sensemaking, decisions are made;
- the social domain: interactions between and among force entities” [EBR 03 p.11].

To achieve such results, we must improve the ability to share information. The ability relies on the networks' characteristics and therefore depends on the network's quality, including, on the one hand, the network's scope, quality of service, security and reliability, and on the other hand the ease with which new nodes can be integrated within the network. On the informational level, the aim is therefore to improve:

- the quality of organic information, which is not shared by all and not available on the networks, and features attributes such as correctness, consistency, precision, relevance;
- the quality of individual information, featuring the same attributes;
- the degree of shared information, featuring the same attributes as the quality of individual information, to which is added the attribute which concerns the ratio of information common to the various entities;
- the degree of information “shareability”, with attributes such as the scope of the collected information posted on the networks or the ease of use.

Beyond this informational level, sensemaking must also be studied; it brings us back to a complex cognitive and social process, hence an individual and social issue. On the individual level, sensemaking features degrees of intelligibility, awareness and individual decisions. On the level of the group, the degree of shared sensemaking includes the degrees of mutual intelligibility (characterized by the ratio of intelligibility elements common within and between each community of interests), shared awareness, and collaborative decisions. Finally, on the social level, the quality of interactions between individuals, groups or organizations, leans on:

- individual characteristics (degree of aversion to risks, individual propensity to trust others, individual expectations about the reliability of others, identification level);
- organizational characteristics (authority, members' diversity, autonomy level, relations between pairs and authority, degree of collective participation and length of a common experience);
- individual and organizational behaviors (cooperation, efficiency, synchronization, involvement, ratio between the efforts provided to maintain the team and the efforts provided to accomplish the task entrusted to the team).

This emphasizes the means by which organizations can exchange information, conciliate different perspectives, and reach both a common comprehension and a common vision. The following diagram illustrates the links between the elements of the conceptual framework of network operations.

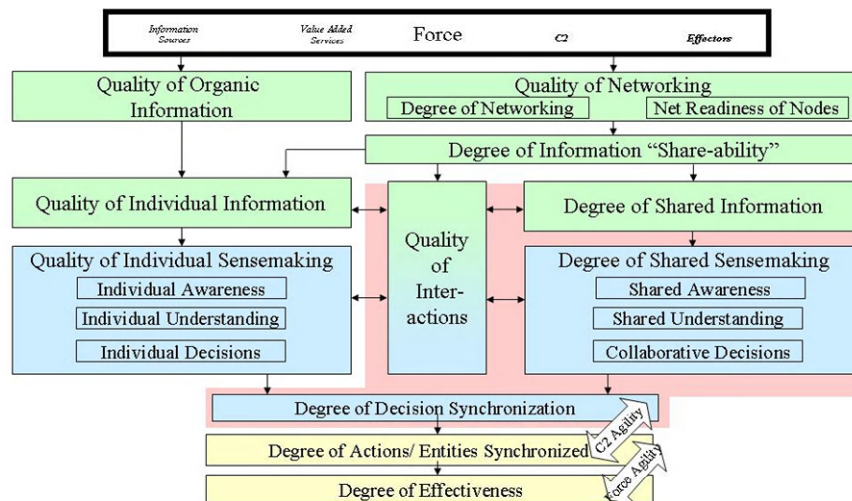


Figure 4.2. *Diagram of the network-centric operations conceptual framework ([EBR 03] p.4)*

These network-centric operations concepts build on the concepts and disciplines of social sciences. The notions of sensemaking, collective experience, trust granted, identification and common culture are all present. Indeed, the characteristics we have just identified, both on the individual and organizational levels, explicitly or implicitly echo the organizations' complexity which we have previously studied.

This NCO conceptual framework document is a major reference on which other documents about network-centric operations in the military field are built, such as *The Network-centric Warrior: The Human Dimension of Network-centric Warfare* [WAR 04] and *Exploring New Command and Control Concepts and Capabilities* [SAS 06].

These documents show, if there was a need, that to look at human factors in the context of systems of systems, we must lean on the concepts of social science.

4.6.3. *Impact of network-centric operations: from technical to organizational interoperability*

Achieving interoperability within network-centric operations is not only a question of designing techniques to make several platforms communicate. It is also a matter of different organizations having to collaborate and become coordinated. Beyond technical interoperability, we must work on organizational interoperability. What do we mean by that?

In the context of crisis management, such as the intervention of several countries in Indonesia and Sri Lanka, after the tsunami, various organizations coming from various countries have to coordinate. On the one hand, they end up using common resources, and therefore competing for the use of resources. On the other hand, they must tighten their collaboration when the activity of one organization depends on the activity of another's. For example, organizations contributing to emergency construction and food aid depend on the organizations which handle logistics. Moreover, the services provided by those logistics organizations are shared by several other organizations, competing over a limited resource. In such a case, the agents within those organizations must regulate their activity, synchronize with one another.

Hazel and Bopping [HAZ 06] underline the aspects which must be treated to reach organizational interoperability, namely "doctrine, legal frameworks, technology, command philosophy, and rank/skill parity." The authors offer to complete the models LISI (levels of information systems interoperability) and OIM (organizational interoperability model), which study technical and organizational

interoperability on a macro or global level, by adding psychosocial and organizational aspects of a micro and meso level.

They identify the context, the identity and the expectations as being of particular importance, since they are linked to the national modes of interaction which have impacts on the operation of a coalition. They offer three possible modes of interaction, interpersonal means which groups implement in order to heighten the awareness of others' activities and regulate their own activities accordingly. These modes of interaction are: observer (a group member simply observes another group's activities), liaison (a group member is assigned to another group to undertake basic activities necessary for ensuring unity of purpose), and finally embeddedness (a group member becomes part of another group for all intents and purposes). They demonstrate how liaison or embedded personnel highly facilitate the organizations' interoperability within a coalition. Moreover, these agents may exploit their personal relationships, or the ones they have within the organizations.

They highlight the fact that the most interoperable component is the embedded agent. Moreover, human beings are a force's most flexible component. Thus, solutions to the stakes of interoperability may be based on personnel rather than hardware.

What's more, the environment within which those organizations must operate, in crisis situations, is complex, dynamic and uncertain. In such a context, the operating organizations must be flexible and adaptive, as advocated by Henry Mintzberg [FID 06, MIN 82]. "The role of IT in modifying and extending the range of coordinating mechanisms and structural configurations is also described" ([FID 06], "Executive Summary"). Building on Mintzberg's works and Groth's extension of those works to the information technologies, Fidock [FID 06] explores the consequences of those technologies on the organizational structures of a coalition command center.

Groth ([FID 06] p. 3-6) expands Mintzberg's five coordination mechanisms, taking into account the capabilities offered by information technologies. This leads to the following correspondences:

- mutual adjustment: implicit coordination by database;
- direct supervision: system-supported supervision;
- standardization of outputs: no identified correspondence;
- standardization of work processes: on the one hand, programmed routines, and on the other, hyper-automation;
- standardization of skills: system-supported.

Groth's extension helps evaluate the implications of information technologies for current and future organizational structures ([FID 06] p. 12-14). He explores the way information technologies modify and extend the influence of the technical systems' factors of contingency.

In that regard, Groth ([FID 06] p. 12) gives the following examples: the use of a database to increase coordination between geographically distributed teams, as well as what he calls the meta-organizations, or extensive enterprise, which help couple separated organizations more tightly together, such as the links between car manufacturers and their suppliers.

From the works of Mintzberg and Groth, Fidock measures the impact on military structures, in terms of decentralization/centralization, that is to say between a higher autonomy of forces on a tactical level, making them able to self-synchronize, or a lower autonomy of forces on a tactical level, to the benefit of the implicit coordination of the central level, to achieve better understanding on a broader context. On the other hand, geographically remote commanders will not have the same evaluation of the local tactical situation than the commanders located on the field. The identification of the best solution calls on a CD&E approach, such as implemented in the battlelabs, and which we have described in Chapter 1.

Moreover, this impacts the structures of power and authority, and, indirectly, the desire to retain power, the management of resources.

Groth ([FID 06], p.15) matches Mintzberg's structural configurations thusly:

- the joystick organization with the simple structure;
- the flexible bureaucracy with the machine bureaucracy;
- the interactive adhocracy with the adhocracy.

Groth ([FID 06] p. 15) adds two configurations which were made possible by information technologies. They are:

- a meta-organization is composed of separate organizations that are tightly coupled to each other through unified computer systems (supplier clusters in the automotive industry), which correspond to network-centric organizations;
- an organized cloud, which is a large organized entity, able to operate in a highly coordinated fashion, due to implicit coordination, such as an airline seat booking system.

What is the implication of information technologies in the command centers? This is the question Fidock is trying to answer.

He demonstrates ([FID 06], p. 19-24) that the communication of objectives, orders, depends on the one hand on the explicit meaning formulated within the order, within the message, and on the other hand on the implicit, tacit meaning, which leans on the operators' shared knowledge, elaborated through common experiences, of indoctrination, organizational culture. This is more than implicit coordination, induced by the simple sharing of a database. The operators also share systems of value, a culture and deep knowledge, all of which, for Mintzberg, pertains to socialization. They also share reciprocal commitments, "moral debts" such as described by Zacklad [ZAC 06].

Information technologies have an extremely limited impact on the improvement of the implicit coordination of such tacit comprehension. Even if those information technologies help improve the ability to elaborate a shared representation of the situation, coordination will necessarily go through mutual adjustment to solve unusual, poorly structured problems. As it turns out, in the context of such crisis situations in which systems of systems may be used, the highly dynamic and unpredictable nature of the environment means the problems are always unusual, always poorly structured.

In light of the increasing importance of information technologies in command centers, and the support they require, Fidock ([FID 06], p. 22) suggests the creation of information managers.

Fidock points out ([FID 06], p. 23) that the design of the organizational system should be entrusted to social sciences experts, who can help system and software engineers with the design of the technological system, as well as managers with the implementation of changes.

4.7. Recognizable good practices in the field of organizations

The various elements we have so far identified allow us to sketch a first list of good practices in the field of organizations, good practices which must be validated through experimentation in order to achieve higher relevance and operational validity.

These good practices are as follows:

- design a professional database of operators [GAL 07], common to the various stakeholder organizations, and design systems which follow that database;
- implement Wacheux's common rules, achieve coherence between values and regulation strategies (sanction/reward);

- favor organic, flexible organization structures, able to take into account instabilities within a troubled environment;
- implement exchanges between organizations, collective activities of elaboration of shared values, mutual commitment and acknowledgment, enabling the agents of the various organizations to develop a common experience, construct a shared meaning and favor the creation of primary groups;
- design trainings based on the conducting of dynamic processes, implementing reactive coordinations through mutual adjustments and improvisations in reply to unexpected events, rather than on planning;
- implement a linking system between organizations [HAZ 06];
- take into account the diversity of the organizational, professional and national cultures of the concerned organizations and individuals;
- entrust the design of organizations to experts in social sciences (organization sociologists), who will help the engineers in charge of designing the technological system;
- design technological systems capable of supporting the aforementioned operating modes, namely capable of helping the implementation of dynamic processes (the technological system must adapt to the operation of the actual organization, and not the reverse).

4.8. Conclusion

This chapter does not pretend to exhaustively present the state of the art of the structuring and dimensioning aspects of social sciences in the context of systems of systems. Some aspects, such as those concerning the cognitive load, have not been addressed. However, we have tried to demonstrate how the performances of a system of systems are closely dependent on organizational, cultural, communicational aspects and aspects of coordination, which must be taken into account, and the need not to limit ourselves to a reductionist approach to organizations, such as the one currently used in the field of information systems and systems engineering.

The documents *NCO Conceptual Framework* [EBR 03], *The Network-centric Warrior: The Human Dimension of Network-centric Warfare* [WAR 04], *Exploring New Command and Control Concepts and Capabilities* [SAS 06], and *Organisational Structure and Information Technology* [FID 06] are the basis on which we should lean to take into account the human factor within the context of systems of systems. Indeed, they articulate the technical aspects of information

systems to the cognitive, social, organizational and cultural dimensions of network operations.

Other studies should be led, on the one hand to go further into the current studies in the field of social sciences, pertinent in the context of systems of systems, for example the transition between several structural configurations depending on the environment's conditions and the need for coordination between organizations, and on the other hand to integrate the studies and skills of social sciences within the engineering process of systems of systems.

4.9. Acknowledgments

I most heartily thank Dominique Luzeaux, without whom I wouldn't have started the adventure which led to this chapter's redaction. I also extend warm thanks to the many people with whom I was able to converse, people I could not give an exhaustive list of, but who will not fail to recognize themselves.

4.10. Bibliography

- [AFI 05] AFIS INTERNET SITE, <http://www.afis.fr>.
- [ARA 84] ARACIL J., *Introduction à la dynamique des systèmes*, Presses Universitaires de Lyon, Lyon, 1984.
- [BAK 02] BAKER M., "Human factors in network-centric warfare", *Final Report*, Naval War College, Newport 2002, available at: <http://www.stormingmedia.com>.
- [BER 73] VON BERTALANFFY L., *Théorie générale des systèmes*, Dunod, Paris, 1973.
- [BOU 56] BOULDING K., "General systems theory: the skeleton of science", *Management Science*, n° 2, 1956.
- [BRO 99] BROWNING T., "Designing system development projects for organizational integration", *Systems Engineering*, vol. 2, n° 2, 1999.
- [CAR 98] CARLEY K., "Organizational decision making and distributed information", *Systems Engineering*, vol. 1, n° 1, 1998.
- [CAR 99] CARLEY K., "On generating hypotheses using computer simulations", *Systems Engineering*, vol. 2, n° 1, 1999.
- [CAR 01a] CARLEY K., HILL V., "Structural change and learning within organizations", in Lormi A. (ed.), *Dynamics or organizational societies: Models, theories and methods*, MIT Press, Boston, 2001.

- [CAR 01b] CARLEY K., LEE J.S., KRACKHARDT D., “Destabilizing networks”, *Connections*, vol. 24, n° 3, p. 79-92, 2002, available at: <http://www.insna.org/Connections-Web/Volume24-3/Carley.web.pdf>.
- [CAR 03] CARLEY K., “Dynamic network analysis”, 2003, available at: [http://www.si.umich.edu/stiet/researchseminar/Winter %202003/DNA.pdf](http://www.si.umich.edu/stiet/researchseminar/Winter%202003/DNA.pdf).
- [CHA 96] CHAPANIS A., *Human factors and Systems Engineering*, John Wiley & Sons, New York, 1996.
- [DAG 04] DAGNINO G.B., “Complex systems as key drivers for emergence of a resource-and-capability-based interorganizational network”, *Emergence: Complexity & Organization*, vol. 6, n° 1-2, 2004, available at: <http://emergence.org>.
- [DUR 79] DURAND D., *La systémique*, PUF, Que sais-je?, Paris, 1979.
- [EBR 03] EVIDENCE BASED RESEARCH INC., *Network-centric Operations Conceptual Framework, Version 1.0*, Evidence Based Research Inc., Vienna, November 2003.
- [EUR 03] EUROPEAN COMMISSION, *Intelligent Transport Systems, Intelligence at the Service of Transport Networks*, brochure published by the European Community, available at: http://ec.europa.eu/transport/its/index_en.htm.
- [FAI 06] FAISANDIER A., “Examen du concept de système; considérer les organisations et les entreprises en tant que systèmes”, *Actes de la conférence AFIS*, 2006.
- [FID 06] FIDOCK J., *Organisational Structure and Information Technology (IT): Exploring the Implications of IT for Future Military Structures*, DSTO, Edinburgh, 2006.
- [GAL 07] GALARA D., “Langage d’expression des Besoins en Informations des Métiers d’Exploitation”, *Surveillance, Sûreté et Sécurité des Grands Systèmes, Actes du Forum Académique AFIS’07*, Nancy, 2007.
- [GRO 05] GROSJEAN M., “L’awareness à l’épreuve des activités dans les centres de coordination”, *@ctivités*, vol. 2, n°1, p. 76-98, 2005, available at: <http://www.activites.org/v2n1/grosjean.pdf>.
- [HAN 03] HANDLEY H., LEVIS A.H., “Organizational architectures and mission requirements: a model to determine congruence”, *System Engineering*, vol. 6, n° 4, 2003.
- [HAZ 06] HAZEL G., BOPPING D., “Linking NCW and coalition interoperability: understanding the role of context, identity and expectations”, *Human Factors in Network-Centric Warfare Symposium*, Sydney, Australia, 2006.
- [HEL 98] HELMREICH R., MERRITT, A., *Culture at Work in Aviation and Medicine: National, Organizational and Professional Influences*, Ashgate, Aldershot, 1998.
- [ISO 02] ISO/IEC, “Systems engineering – system life cycle processes”, *ISO/IEC 15288:2002*, 2002.
- [KIR 04] KIRKE C., “Organizational culture: the unexpected force”, *Journal of battlefield technology*, vol. 7, n° 2, July 2004.

- [KIR 05] KIRKE C., "Organizational culture: can system designers ignore it?", *Proceedings of the IEE and HFI DTC Symposium on People and Systems: Who are We Designing For?*, p. 9-15, London, 16-17 November 2005.
- [KOL 97] KOLSKI C., *Interfaces Homme-Machine*, Hermès, Paris, 1997.
- [LEM 90] LE MOIGNE J.L., *La modélisation des systèmes complexes*, Dunod, Paris, 1990.
- [LEP 03] LEPREUX S., ABED M., KOLSKI C., "A human-centred methodology applied to decision support system design and evaluation in a railway network context", *Cognition Technology and Work*, vol. 5, pp. 248-271, 2003.
- [LEV 04] LEVER J., "Unintended consequences of the global positioning system", *Systems Engineering*, vol. 7, n° 3, 2004.
- [MAH 09] MAHATODY T., SAGAR M., KOLSKI C., "State of the art on the cognitive walkthrough method, its variants and evolutions", *International Journal of Human-Computer Interaction*, 2009.
- [MAI 98a] MAIER M., "Architecting principles for systems-of-systems", *Systems Engineering*, vol. 1, n° 3, 1998.
- [MAI 98b] MAIER M., "Architecting principles for systems-of-systems", 1998, available at: <http://www.infoed.com/Open/PAPERS/systems.htm>.
- [MAR 58] MARCH J.G., SIMON H.A., *Organisations*, Wiley & Sons, New York, 1958.
- [MEI 98] MEINADIER J.P., *Ingénierie et intégration des systèmes*, Hermès, Paris, 1998.
- [MIL 04] MILLS V., SMITH R., "Short- and long-term effects of participation in a cross-cultural simulation game on intercultural awareness", *DSTO*, Edinburgh, 2004.
- [MIN 82] MINTZBERG H., *The Structuring of Organizations*, Prentice Hall, 1979.
- [MOR 05] MORIN E., *Introduction à la pensée complexe*, Le Seuil, Paris, 2005.
- [NOR 90] NORMAN D., *The Design of Everyday Things*, Doubledays, New York, 1990.
- [PAJ 00] PAJEREK L., "Processes and organizations as systems: when the processors are people, not pentiums", *Systems Engineering*, vol. 3, n° 2, 2000.
- [PER 84] PERROW C., *Normal Accidents*, Doubledays, New York, 1984.
- [PIA 75] PIAGET J., *L'équilibration des structures cognitives*, PUF, Paris, 1975.
- [RAS 91] RASMUSSEN J., BREHMER B., LEPLAT J., *Distributed Decision Making*, John Wiley & Sons, Chichester, 1991.
- [REA 97] REASON J., *Managing the Risks of Organizational Accidents*, Ashgate, Aldershot, 1997.
- [ROJ 03] ROJOT J., *Théories des organisations*, Eska, Paris, 2003.
- [ROU 05] ROUSE W., "A theory of enterprise transformation", *Systems Engineering*, vol. 8, n° 4, 2005.

- [SAL 04] SALEMBIER P., ZOUINAR M., "Intelligibilité mutuelle et contexte partagée, inspirations conceptuelles et réductions technologiques", *@ctivités*, vol. 1, n° 2, p. 64-85, 2004, available at: <http://www.activites.org/v1n2/salembier.pdf>.
- [SAS 06] SAS 050, "Exploring new command and control concepts and capabilities", *Final Report*, prepared for NATO, January 2006.
- [SCO 00] SCOTT J., *Social Network Analysis*, Sage Publications, CA Thousand Oaks, 2000.
- [SHE 84] SHERMAN S.J., CORTY E., "Cognitive heuristics", in Wyer R.S. and Scrull T.K. (eds.), *Handbook of social cognition*, vol. 1, Lawrence Erlbaum, Hillsdale, 1984.
- [SMA 05] SMART P., SHADBOLT N., CARR L., SCHRAEFEL M., "Knowledge-based information fusion for improved situational awareness", 2005, available at: http://eprints.ecs.soton.ac.uk/11065/01/Fusion_2005_Final_Draft.pdf.
- [SPE 89] SPERBER D., WILSON D., *La Pertinence : communication et cognition*, Les Editions de Minuit, Paris, 1989.
- [SUT 04] SUTTON J., COSENZO K., PEIRCE L., "Influence of culture and personality on determinants of cognitive processes under conditions of uncertainty", *9th International Command and Control Research and Technology Symposium*, Copenhagen, September 2004.
- [SWA 03] SWAIN K., MILLS V., *Implicit Communication in Novice and Expert Teams*, DSTO, Edinburgh, 2003.
- [SWE 05] SWENNEY K., "The assessment of non-physical human factors in the context of the naval capability evolution process", *INSIGHT*, vol. 8, n° 1, 2005.
- [THO 04] THOMPSON J., *Organizations in action*, Transaction Publishers, New Brunswick, 2004, first edition 1967.
- [VAR 88] VARELA F., "Une approche de l'étude de l'autonomie et de la complexité", in Schwarz E. (ed.), *La révolution des systèmes : une introduction à l'approche systémique*, Editions del Val, Fribourg, 1988.
- [VAS 90] VASKE J., GRANTHAM C., *Socializing The Human-Computer Environment*, Ablex Publishing Corporation, Norwood, 1990.
- [WAR 04] WARNE L., ALI I., DOPPING D., HART D., PASCOE C., *The Network-centric Warrior: The Human Dimension of Network-centric Warfare*, DSTO Information Sciences Laboratory, Australia, July 2004.
- [WAS 00] WAGENHALS L.W., SHIN I., KIM D., LEVIS A.H., "C4ISR architectures: II. A structured analysis approach for architecture design", *Systems Engineering*, vol. 3, n° 4, 2000.
- [WEI 95] WEICK K., *Sensemaking in Organizations*, Sage Publications, CA Thousand Oaks, 1995.
- [ZAC 06] ZACKLAD M., Une approche communicationnelle et documentaire des TIC dans la coordination et la régulation des flux transactionnels, work document available at: <http://archivesic.ccsd.cnrs.fr/>, 2006.

Chapter 5

Space Communication and Observation System of Systems

The following chapter aims to illustrate, as simply as possible, how the notion of system of systems is consubstantial to the space systems and the daily services provided by these satellites: telecommunication-television broadcasting, guidance-navigation-dating, but also meteorology-oceanography-geography, which rely on Earth observation abilities, some of which are dedicated to the military intelligence of certain countries.

5.1. The dual context of omnipresent information and the commoditization of space

The increasing need for telecommunication services available in all places and at all times, as well as multimedia distribution, localization and navigation, converters, telecourses, telemedicine, etc., drives the development of new communication systems essentially based on wireless technologies. Concurrently, the growing interest in making such services available within regions where the telecommunication infrastructures are very limited is an answer to the modern problem of digital divide.

This is how the satellite communication and global navigation systems provide access to strategic technologies which have an important economic and social impact in the following fields: data transmission for air, rail and sea transports,

natural disaster forecasting, humanitarian aid, crisis management, etc. These fields of application require the transfer of images, videos, voices, and the performance in terms of robustness, transmission delay and security must be accordingly high.

Moreover, the impact of the new information and communication technologies has created a new paradigm which elevates the information and communication system to the level of corporate strategy. No longer a simple infrastructure of interconnection and service networks, doubled with low-level data transfer services, the so-called information and communication systems have gained a real strategic value. This concerns all the various fields of application, civil and military.

The United States have sought to profit from this new paradigm in the field of defense, by adopting the so-called Information Warfare doctrine, based on the notion of information superiority (sometimes called *full information dominance*). The underlying idea is that conflicts, whether armed or not, are for the most part fought and won on the battlefield of information. The documents *Vision 2010* and *Vision 2020* focus the American military strategy (but also the military strategy of NATO and of the vast majority of countries) and the control of information flows. Initially approved in the second half of the 1990s, and despite being criticized after September 11 and the Iraqi crisis which started in 2003, these documents are still topical and represent the logical progression of the technical objectives defined by William Perry as early as 1978:

“to be able to see all high-value targets on the battlefield at any time;
to be able to make a direct hit on any targets we can see; and to be
able to destroy any target we can hit.”

The path to meeting these objectives goes through the creation of the system known as C4ISR (Command, Control, Communication, Computers, Intelligent, Surveillance, Reconnaissance), which regroups the data and puts them at the commanding authorities' disposition. In its broadest definition, this system includes the decision components, as well as the components of the weapon systems, providing them with adequate guidance and navigation. In theory, its extensive use gives access to full connectivity and in turn to the establishment of an unbroken situation awareness from one end of the chain of command to the other, and helps achieve more precise targeting and increase the theater's depth. Let us look at a concrete example: statistically, during World War II 4,500 bombers would each have to drop two tons of bombs in order to destroy a target the size of a house. Today, it would only take a few cruise missiles launched several hundred miles away. The aforementioned system, which obviously fully pertains to the class of systems of systems, as has been mentioned in Chapter 1, helps look at military operations as OODA loops, consisting of four steps: observe, orient, decide and act. The enemy is first observed, then this information helps the commanding authorities

orient their choices; the final decision is reached through a decision making process, after which the action finally happens (whether it be further observation to complete the information, or the actual fight). The result is studied, and another loop begins. The goal is obviously to chain OODA loops faster than the enemy, so as to keep the upper hand.

From a technical standpoint, three things can help improve those loops' speed and performance:

- Bandwidth: during the first Gulf War in 1991, the coalition had a bandwidth of 100 Mb/s. In Afghanistan in 2001, capacities went up to 800 Mb/s, they reached 7 Gb/s in Iraq in 2003, and bandwidths in the range of 16 Gb/s are expected in 2010.

- Precision of observation: the first goal is to distinguish details as small as possible so as to be able to not only detect, but also recognize possible targets and threats (for example, being able to tell apart a bus and an armored personnel carrier), and in some cases identify them (for example, distinguish one type of vehicle from another); the second goal is the ability to localize fixed and mobile objects within a reference three-dimensional space, with metric accuracy, either to improve the comprehension of the observed scenes, or to optimize the resulting geographical data and avoid, for example, targeting errors and the associated “collateral damages”.

- A comprehensive view: intelligence must be as extensive and detailed as possible, the objective being constant real-time surveillance, which entails demands both in spatial and temporal coverage.

All three requirements call on the use of space: telecommunication, localization and remote sensing satellites are therefore tools inherent to the very nature of this revolution of military affairs.

5.2. The technical view: an interconnection of ground-based and space-borne systems

A space system is by nature a distributed system whose orbital segment, which can be composed of several satellites, is a component which cannot, in itself, fulfill a mission.

Even if it only plays a fleeting part in the satellite's life, the launching infrastructure itself can be apprehended as a system of systems composed of a “spaceport”, to use the appellation of the Kourou base, of the launcher itself, and also of the trajectography and telemetry centers scattered around the surface of the globe. These tools may even be mobile, such as the French Minister of Defense's Monge naval building, in order to adapt to specific trajectories.

In a steady state, a space system includes *a minima*, besides the orbital segment, a ground control or station keeping segment, and a user ground segment. If the orbital segment is the most visible and therefore most critical, the control segment is just as essential for the maintenance of the function, and can if need be extend the life of the space-borne component, by guiding the satellites on particular orbits, for example. Lastly, the user ground segment can be located in several places and made up of several interconnected systems.

5.2.1. Telecommunication and navigation satellite systems

In order of complexity, we first find the geostationary telecommunication satellites, which are, in (excessively?) broad outline, elevated cell towers (22 400 miles high!). Then come the radio navigation and communication systems, which use low earth orbiting satellite constellations. These capacities require a global approach of the constellation, which is constructed through increments: constantly, new satellites are being purchased, other satellites are in use, and some satellites are reaching their end-of-life, all spreading over several technological generations.

With regard to telecommunications, the needs, civil as well as military, are constantly increasing: as an example, the ambition of the United States in the military field is to multiply bandwidth by ten in years to come. This is achieved through the sharing of resources between civilians and military forces, and an ever growing migration towards civil resources: whereas 75% of the war theatre telecommunications had transited through military satellites during the First Gulf War, only 40% did so during the war in Kosovo and 20% during the Second Gulf War. Such sharing of resources demonstrates the criticality of a global approach where the use of systems is mostly about the management of the availability and capacity of a service which uses various systems, in space and on the ground.

It should be noted that the civil or military space telecommunications only represent one of the components of the telecommunication system: the ground segments include fixed earth stations, and in the military field, mobile earth and sea stations. The interconnection of the various networks ensures the function of global communication. In the military field, national authorities and the authorities deployed on warfare theaters can therefore command the troops regardless of the distance, discarding the local infrastructures.

This all seems normal to us, since we use that function daily. It calls on physical interconnections, constant signal exchanges following various protocols, interoperability on various levels, security software to guarantee the data's integrity, and the constant upgrading of some of its components. Incidentally, we should remember that, on the one hand, the life of any satellite is subject to the laws of

physics, and therefore not is eternal, and on the other hand the constant need for improved performances requires technological upgrades to be performed on the various equipment, which is not easily achievable with space components. The configuration management of the global technical architecture is all the more critical, as well as the management of its obsolescence with regard to the expected evolutions in service quality.

Localization by satellite, at the base of the navigation function so widely used in daily life and in military operations, follows this scientific principle: a cellular phone equipped with a specific receiver can be localized within a universal reference system, based on its distance from at least four satellites in simultaneous visibility; its speed can be calculated via the measurement of the Doppler effect of the frequency emitted by these four satellites. Therefore, to localize anything anywhere on the planet, one only needs to have access, on each point, to a minimum of four satellites, which implies spatial coverage via a satellite constellation, as well as a signal reception system and a minimum calculation capacity to deduce the necessary information from the compiled data. The American GPS (global positioning system), and the Russian GLONASS (in English, global navigation satellite system) have thus been designed to provide millions of civil and military users with information on their position at any time, anywhere on the globe. These systems also transmit a precise time reference used in many applications, for example to synchronize cell phones with the communication networks' base stations, or to track the position of a mobile user.

Nowadays, the American GPS constellation has the monopoly on that service. Its general architecture consists of three major segments: the space segment, the control segment, and the user segment, both of which are on the ground. The space segment is composed of 24 satellites arranged so that each point of the globe receives signals from at least six of them almost constantly. The control segment is composed of five ground stations scattered over the world: each of the constellation's satellites completes one orbit of the Earth in 12 hours and carries a precise clock which enables the exact dating of any transmitted signal. The user segment is represented by receivers, which can be carried manually or embedded in vehicles. The standard localization precision is in the range of 100 m, and can be corrected by a factor of 100 by a receiver whose fixed position is known. Nowadays, more than 100 different types of receivers can be found on the market, with varying sizes.

Ever improved, the GPS system is a major leverage tool for the United States, in particular in times of conflict, insofar as a great many combat and weapon systems use GPS for localization, navigation and synchronization. This is why the European Union has decided to develop the Galileo system, which should be operational around 2013. China and India each have the same objective of acquiring an independent global navigation system.

The example of GPS navigation and its connected services, whether it be traffic maps in our cars or distress beacons for adventurers, is the archetype of the interconnection of numerous systems scattered geographically in space and on the ground, which were manufactured separately and can be used independently, and provide new services when grouped within the same architecture.

5.2.2. Space-borne remote sensing and observation systems

Those systems gather information, on the globe's surface and in the atmosphere, and compile the signals within various electromagnetic tapes, often translating them into images. An important characteristic of satellites in charge of transferring images is the ground resolution: nowadays they often achieve a resolution of 1 m. Thus, *Ikonos*, which was put into orbit by the United States in 2000, provides commercial images with a resolution of 80 cm, and *Quickbird*, launched in 2001, achieves resolutions of 61 cm. In both cases, private initiatives are encouraged by the United States Government, and the images are at the basis of services provided to users, simultaneously processing space observations and complementary information: we only need think about Google maps, or other services recently accessible via the Internet.

If remote sensing satellites were originally used by the military, in particular to gather intelligence on nuclear powers and guided missiles during the Cold War, and to implement the treaties on arms controls, nowadays those satellites are just as much civil as military, and result in cooperative exchanges of capacity, as we will study in the following paragraphs.

The systemic aspect of space-borne observation and telecommunication became fully apparent during the Kosovo Conflict in 1999: the targets were defined from the fusion of data collected by observation and spy satellites, the missiles sent to destroy said targets were guided by GPS, the results were evaluated by the spaceborne and airborne intelligence network, the command systems, California-based in the case of the United States, were in permanent liaison, and the media coverage was also passing through space systems! Moreover, the space and ground components worked complementarily with airborne systems such as UAVs (unmanned aerial vehicles) and reconnaissance aircrafts, in particular for the collection of intelligence.

The following conflicts, Afghanistan in 2001 and the Second Gulf War in 2003, confirmed those trends. In terms of technical performances, the integration of these various resources within a system of systems, in addition to the technical improvement of the various components, helped drastically shorten the OODA loops, going from about 48 hours in 1999 to 10 mins in 2003.

5.3. Search for functionality and capacity

It should be noted that in the case of telecommunication/broadcasting and radio navigation systems, the user ground segment cannot be dissociated from the service. In the first case, it enables the transmission or reception of information to where the user stands. In the second case, it is the presence of the radio navigation receiver where the user stands which enables precise pinpointing of that user's location. In both cases, the satellite system only provides a generic capacity, or even part of that capacity, the other part being at the hands of a component acquired by an end-user which might belong to the public at large (antenna and decoders for satellite television, or GPS receiver) and therefore might be acquired in a logic largely independent from the one who presided the contracting or design of the satellite system.

On the other hand, the use of Earth observation satellite systems relies on "centralized" processes, since the end-user will have to file his request for observation with an organization which, if it does not possess the information in its database, will transmit the request after managing the priorities upon programming of the satellite.

However, in order to optimize the system's performances (see above), the ground segments of control, station keeping and mission, relying on a network of telemetry stations, are scattered on the surface of the globe. These infrastructures, while belonging to different organizations, are "mutualized".

The global design of a satellite observation system greatly varies depending on the desired capacity. The capacity can notably be described in relation to the following terms: life expectancy, the observable zones and the priorities between said zones, the revolution rate, the quality of images (ground resolution, spectral, radiometric, level of noise, geometric quality, etc.), the nature of the elaborated products (panchromatic, color, stereo couples, etc.), the hours of exposure (heliosynchronous orbit, or not, and if so, the hour of passage over the equator, geosynchronous, etc.), the satellite's agility (ability to observe everything within a cone around the nominal line of sight, and to rapidly change the line of sight), the incidence of the exposures. These characteristics are strongly dependent on the chosen platform, instrument and orbit. However, the arbitrations are also dependent on the connected infrastructures, notably terrestrial. Besides the ability to insert the space launchers on a given orbit from a given launch area, the ground systems have an impact on the satellite's programming delay (time between the moment the user asks for an image and the transfer of his demand to the orbital segment), on the information's age (time between the moment the exposure is taken and the moment it is made available to the user) and therefore on the delay of access to information (time between the moment the image is requested and the moment it is made

available to the user). As an example, the ESA is using the stations network ESTRACK, which shares the Kourou (French Guyana) and Kiruna (Sweden) stations with the CNES, which also uses, for systems such as SPOT 5, the stations of Aussaguel (near Toulouse, in France) and Hartebeesthoek (South Africa), in order to optimize station keeping and mission programming. In its transient stage, the CNES also uses the stations of other agencies: Wallop Island (USA) and Poker Flat (Alaska) for NASA, Okinawa and Katsuura (NASDA, Japan), Prince Albert (CCRS, Canada). The vocation of centers such as CNES's PASO (architecture panel of orbital systems) is to enable the definition of these global architectures. Like the battle-labs mentioned in Part 1, Chapter 1, these centers use simulation tools to optimize the border/grounder compromises, but also the platform/orbit and cost/performance ones.

The GMES project (*Global Monitoring for Environment and Security*), a joint initiative of the ESA (European Space Agency) and the European Union, is the proof of a strong European desire to federate and rationalize Earth observation activities in Europe. This project consists in a set of thematic services, whose first components should be operational as of 2008, and which will leverage the existing and future infrastructures, but also help develop assets for the collection and distribution of data, and integrate these data within environmental monitoring and prediction systems. It also plans to ensure long-term continuity and the upgrading of the space infrastructures needed for the gathering of said data. Besides providing Europe with a trustful, precise environmental information system, this initiative also contributes to the common policy of security and defense, *via* dual uses (civil and military) of some spatial resources in particular.

5.4. A logic of exchange on an international scale

It should be noted that in the fields of weather forecasting and oceanography, among others, the observed phenomena can be explained by coupled multi-scaled mechanisms (micro, meso and macro); corollary from that coupling, the forecasts only have a local interest even though they call upon the collection of data on the entire surface of the planet. The need for information on various scales and about mechanisms which are not always discernible from space leads to the use of observation tools *in situ* just as much as space components. The acquisition of these various components represents a width of investment which no single country can sustain in time. Moreover, scientific and industrial stakes lead countries or regional organizations to finance the constituents of both components. Coordination happens within international organizations (answering to the United Nations). Besides the distribution of requirements between the various components and constituents, and the coordination of schedules so as to assure the permanency, or even the

improvement, of performances, the main challenge lies in the ability to exchange data.

The issue is similar in the field of imagery for national defense and security, where resources are sometimes shared with civil resources. In the following paragraphs, we will study the organizations which are created in order to optimize the value chain of the observation and telecommunication system of systems, following that organizational dimension.

5.4.1. *The GEOSS program*

The GEO (Group on Earth Observations) is an international group launched on a voluntary basis following the calls for action by the 2002 World Summit on Sustainable Development, and the G8 group of the leading industrialized countries. It represents a step towards meeting the goals set by the United Nations Millennium Declaration, furthering the implementation of the obligations linked to the international environmental treaties.

The GEO offers a framework within which partners can develop new projects and coordinate their strategies and their investments. At the end of 2007, the group's membership included a total of 71 governments as well as the European Commission, and 46 intergovernmental, international and regional organizations as participating organizations. It coordinates efforts within the GEOSS (Global Earth Observation System of Systems) program.

GEOSS's objective is to become a global Earth observation system, coordinated and maintained through time in order to improve the surveillance of the state of the planet, the comprehension of the processes which govern the Earth and the forecasting of the Earth system's behavior.

The GEOSS system of systems is a global public infrastructure which must generate environmental data and analyses in near-real-time, for the benefit of a wide range of users and decision makers. Its purpose is to interconnect the existing and future observation systems, whether they be floating buoys monitoring the oceans' temperature and salinity, meteorological stations and balloons recording air quality and rainwater trends, sonar and radar systems reckoning the fish and bird populations, seismic and GPS stations recording movements in the Earth's crust, some 60-plus satellites observing the Earth from space, or early warning systems, for example against tsunamis; it must also interconnect numerous numeric models used for various simulations and forecasts. GEOSS seeks the interoperability of all these tools and also aims to reduce costs and promote international cooperation.

Its ambition is to provide information useful to the nine “social benefit areas”: natural disasters, health, energy, climate, freshwater resources, weather forecasting, ecosystems, agriculture and biodiversity. More precisely, the aim is to:

- reduce the loss of property and human lives resulting from natural or human-induced disasters;
- understand the environmental factors which impact health and well-being;
- improve the management of energy resources;
- understand, assess, predict, mitigate and adapt to climatic changes and variability;
- improve the management of freshwater resources through a better understanding of the water cycle;
- improve the weather information, forecast and warning;
- improve the management and the protection of the terrestrial, coastal and marine ecosystems;
- encourage sustainable agriculture and fight against desertification;
- understand, monitor and preserve biodiversity.

In addition to the interoperability of the measuring and calculation tools, the collected data must also be pooled together, insofar as one set of data can be useful to several users, just as one user might need several datasets. On a technical level, this requires, on the one hand, the use of common standards so each existing or future component can communicate with the other systems, and on the other hand the certainty that each user will adhere to the exchange principles of the system of systems, for the data, the metadata and the products. For example, on the level of data and metadata, one challenge is to coordinate the socio-economic variables; in more prosaic terms, a common model of ground elevation, providing a stable, accurate, homogeneous and global geodetic point of reference, is also necessary to efficiently compare the set of observations. Besides these data models, the GEO competent technical groups are also developing a data quality strategy, and implementing better practices relative to the calibration and validation of the sensors and the data.

The standards used within GEOSS are recorded as technical specifications sanctioned by the participants and based on non-proprietary, recognized international standards. Insofar as a certain number of GEOSS’s constitutive systems will ultimately follow their own path, and thereby acquire operational independence, the real demand lies in standardizing the interfaces through which the various systems connect to GEOSS’s other components. Moreover, the descriptions

of the components, services and standards are formally recorded in a document managed by GEO's adequate technical groups.

In addition to this reflection on interconnection standards, GEOSS also provides true collaborative infrastructures, such as sensor networks, which enable communication between geographically scattered sensor platforms. Likewise, a concept of virtual constellation is in development, and will enable the coordination and correlation of the measures provided by the various satellite networks contributing to GEOSS. Lastly, the output data compiled by GEOSS are accessible via a portal, with a view to presenting them to decision makers and the set of users of Earth observations.

One of the objectives of GEOSS in the field of sustainable development is to implement true capacities at the disposal of the international decision makers, which will allow them to manage and protect the natural resources and drive the private sector in the same direction. For example, GEONETCast is a system providing environmental data in near-real-time, from earth, sea, air and space observations, transferred to the users via a network of four communication satellites. The potential users include organizations established in countries with limited access, or no access at all, to high-speed internet. One of the capacity objectives of GEONETCast is therefore to help the users, among whom the ones susceptible to belonging to the decision chain, identify their top priority needs and train themselves to use potentially useful data. Such issues pertain more to governance than to technical and functional demands, even if the demands are obviously essential in reaching the determined capacity objectives.

5.4.2. *Necessary governance of the GEOSS program*

GEOSS is governed by a plenary consisting of all members and participating organizations, which meets at least once a year at the level of senior officials, and periodically at the ministerial level. An executive committee composed of twelve elected representatives from the five GEO regional caucuses (three for each of the American, European and Asian continents; two for the African continent; one for the Commonwealth of Independent States) pilots the GEO activities in-between plenary sessions. Four co-presidents elected by the members of GEO preside over both the plenary and the executive committee.

The 10-Year Implementation Plan, adopted in February 2005, sets GEOSS's vision, its scope, its priorities in terms of capacity and techniques, its governance structure. Moreover, it defines 107 objectives to meet within two years, 82 objectives within six years, and 56 objectives within 10 years. Committees and

workgroups are created to implement these various points, and define action plans with precise deadlines.

GEOSS is, as stated by its acronym, a true system of systems: the *in situ* space observation systems, as well as the information systems which transform and transfer the data to decision support systems, will retain their initial missions as well as their own modes of governance. The standardizing elements which we have mentioned in the previous paragraphs will help resolve this conundrum, and their definition as well as their acceptance is part of the governance principles. Moreover, the international GEO group is in negotiation with national and international organizations for the acquisition of a certain number of dedicated radio frequencies, in particular for the transmitting of certain satellite measurements.

If the notion of architecture is frequently brought up, it should be pointed out that it is mainly inscribed within a bottom-up approach, going from the systems to a global capacity. Its principal aim is to insure the programs' coordination and the implementation of shared standards for the production and sharing of data. On the other hand, there is no engineering infrastructure aimed at a top-down approach, going from the need to the allocation of demands to the constitutive systems. This idea is notably found in the key documents with such assertions as: "The success of GEOSS will depend on data and information providers accepting and implementing a set of interoperability arrangements, including technical specifications for collecting, processing, storing, and disseminating shared data, metadata, and products."

We should however notice that the set of space systems is not yet thought out as an intrinsic "space component" to which is "collectively" allocated part of the performances expected from the system of systems. To this day, the main limitation is essentially budgetary. It also stems from the implemented governance, which relies on international organizations treating a given issue, weather forecast for example, or regional organizations such as Europe, which contributes to GEOSS via GMES (Global Monitoring for Environment and Security).

5.4.3. *Capacity exchanges in the military field*

Things are slightly different within the military field, as we are about to see. For example, the American approach to space-borne military intelligence consists of defining the operational capacity of the space component, and achieving it by relying on the possible interactions between satellites; for example by using a communication satellite as the relay of an observation satellite. In Europe, the approach somewhat differs, essentially for historical reasons relative to the recent

constitution of the European Union as an entity featuring one common policy and a desire for the common acquisition of systems.

In 1999, during the meeting of the European Council in Helsinki, the European Union decided to achieve autonomous action on the military level, leading to the creation of the European Security and Defense Policy. This ability to quickly deploy military forces capable of leading operations at corps level was achieved from conventional contingents provided by the various countries. But this force must be provided with the means crucial to autonomous action, namely the ability to listen and observe, and more generally to gather intelligence. There is still some way to go, since the intelligence approach is still fundamentally national, even though exchanges between services have been going on for a long time. A real European Intelligence doesn't yet exist, but one of the first steps in that direction is the sharing of resources which were up till now jealously withheld, namely the satellites and the data they compile.

To this day, the French own the military observation satellite *Helios*, which provides optical imagery data. The first generation, with the launch of *Helios I A* in July 1995 and *Helios I B* in December 1999, was designed in cooperation with Italy and Spain; each of these nations can order images via these satellites for its own benefit. The second generation, with *Helios II* and a first satellite launched in 2004, followed on from the previous cooperation strategy with the added partnership of Belgium, Greece and Germany, and saw an improvement in image resolution, faster exposures, the addition of infrared capacities and an improvement of the ground stations in order to answer the demands of end-users following the feedback of the previous generation. Germany owns the *Sar-Lupe* system, a constellation of five satellites, launched between 2006 and 2008, which use synthetic aperture radars and therefore achieve very high resolution. Italy owns the system *COSMO-SkyMed*, composed of four satellites, the first of which was put into orbit in June 2007, and which also provides high-resolution images through the use of synthetic aperture radars. The bilateral treaties which have been established in recent years regulate the exchanges of images between these countries, so each of them can have, in theory, access to an operational Earth observation capacity at all times, day and night and regardless of the weather, thanks to the complementarity of the various sensors.

The objective is to further those advances, and achieve a common system of space-based imaging: the common operational need has been defined, and the end-of-life of the current systems is planned between 2014 and 2017. Rather than relaunching independent space programs on the level of each nation, we ought to combine each country's capacity, which is the aim of the future MUSIS system (multinational space-based imaging system), born from the cooperation of six European nations: France, Germany, Belgium, Spain, Greece and Italy.

Besides improving the sensors' performances, this system will enable a single access to the system's various space-based imaging components: optical sensors, radar, infrared, hyperspectral. To this day, we cannot program the various satellites through one single tool, but the MUSIS system should give an operator the possibility of programming the satellite the most adapted to his needs from a single ground station, and it should also reduce delays between the programming of a satellite and the acquisition of the image by the authorities. This might be possible through the use of relay satellites, programming and receiving stations located on the field of operations, as well as the sharing of data through the creation of a common image and information database. We are faced with a system of systems logic of capacity, in which we must go much further than the simple juxtaposition of several chains, respectively composed of space and ground segments, and achieve real integration of the ground components providing a common service which optimizes the available space resources.

Moreover, the various space telecommunication systems which have been deployed by certain nations within NATO *a priori* feature gateways to interoperability, which can enable mutual support and coverage extension.

5.5. Conclusion

Spaceborne telecommunication and observation systems require the implementation of many systems, both in space and on the ground, in order to provide functions of capacity evolving through time. In that way, the definition of systems of systems, such as has been given in Chapter 1, naturally applies.

To further add complexity to the previously discussed space system of systems, and in particular the technical and functional architectures, let us mention DARPA's (Defense Advanced Research Projects Agency) System F6 program (*Future, Fast, Flexible, Fractionated, Free-flying Spacecraft United by Information Exchange*) in the United States, which aims at fractioning the traditional monolithic satellites into a cluster of smaller satellites, each weighing under 650 pounds, launched separately but later flying in formation, and interconnected through wireless links, or maybe even capable of physically binding, in order to create a single virtual satellite. Small satellites within the cluster could exchange information and power. The calculation capacity would also be distributed and could potentially be heightened through the addition of new modules. The global objective is to reduce the risks of destruction and increase the robustness of the functions carried out by space systems, while adding flexibility and evolutionary capacities so as to prevent obsolescence.

5.6. Bibliography

- [BLA 05] BLAMONT J., Espace et défense, Presentation during the colloquim organized by the Académie des technologies in honor of Mr Hubert Curien, Paris, September 15, 2005.
- [GEO 05a] GEOSS, *The Global Earth Observation System of Systems (GEOSS) 10-Year Implementation Plan*, ESA Publications Division, The Netherlands, Bruce Battrick (ed.), February 2005.
- [GEO 05b] GEOSS, *The Global Earth Observation System of Systems (GEOSS) 10-Year Implementation Plan Reference Document*, ESA Publications Division, The Netherlands, Bruce Battrick (ed.), February 2005.
- [KNA 07] KHALSA S.J.S., NATIVI S., AHM T., SHIBASAKI R., THOMAS D., *The Global Earth Observation System of Systems (GEOSS) Interoperability Process Pilot Project*, IGARSS, 2007.
- [MinDef] MINISTÈRE DE LA DÉFENSE, portal accessible at: <http://defense.gouv.fr>.
- [MUS 08] MUSQUÈRE A., “La DARPA veut fractionner les satellites”, *Air & Cosmos*, n° 2116, March 14, 2008.
- [OM 08] OBAIDAT M.S., MARCHESE M., “Recent advances in global navigation and communication satellite systems”, *IEEE Systems Journal*, vol. 2, n° 1, 2008.

Chapter 6

Intelligent Transport Systems

6.1. The field of intelligent transport

6.1.1. *ITS*

The acronym ITS stands for *intelligent transport systems*. It can be used in two contexts. As *ITS systems*, the form most commonly used, it refers to the new transport products, services and systems which call on state-of-the-art technologies, notably in the field of computer sciences, communications and electronics, which are called “intelligent” because their essential functions are based on qualities generally associated with intelligence: sensory capacities, memory, communication, processing of information and adaptive behavior.

As ITS, the acronym refers to the intelligent transport system, in constant evolution, which integrates, on the technological level as well as the institutional, all the current means of transport of passengers and freights and regroups the new subsystems, products and services of intelligent transport.

6.1.2. *The systems in use*

Whether it is for the transportation of people or freights, many systems are in use.

The systems vary greatly: transport operation systems, toll systems, traveler information systems, law enforcement systems, systems for the sharing of information and cooperation between operators, etc.

The passengers or the freight are, by definition, the central data of the various systems.

The traveler unknowingly follows all the concepts exploited by ITS systems, such as multimodality, interoperability and other new information and communication technologies.

The “*Homo Mobilis*” checks the weather and the traffic on his television, takes his car to the train station, pays the highway toll and the parking, rides a train, then a subway and when he reaches work he is ignorant of the fact that he has been the subject of an adapted treatment provided by more than a dozen systems.

Year after year, transport facilities call on more and more advanced technologies, to offer more and more diversified services and answer the great challenges of sustainable security and mobility: optimization of the traffic flow, enforcement of regulations, traveler information services, computer ticketing, emergency management, management of freight and fleet, etc. These technologies rely on advanced and complex information systems, whose continuity and evolutionary nature must be ensured.

Moreover, the challenges of intermodality lead the agents of the various means of transport to work together. These evolutions in the field of transports call for new approaches in the running and management of projects in order to avoid the rollout of systems lacking real compatibility and evolutionary capacities.

The early creation of guidelines and organization of the transport systems are therefore essential to the interoperability and durability of the systems and the investments. It is essential that the general contractors, and in general any agent of a project, have access to a simple method, model, and tools that will assist them in thinking faster and more efficiently.

The concept of systems of systems is, in that field, a vector for the improvement of customer service quality.

In the field of systems in general and information systems in particular, the term “architecture” has many meanings. The intelligent transport systems are systems cooperating to provide services to their beneficiaries, mainly users and operators. The increasing interconnection of the transport systems and therefore of the ITS systems, and the implementation of the national and international (European in our

case) regulations, bring about the definition of a multiproject architecture as a rule for a cooperative approach between the stakeholders. Indeed, the success of the architecture approach will depend on the support of every agent, and will only be established if everyone's needs and demands are taken into account.

Concepts from the fields of computer science and information systems can be applied to the organizations running ITS systems, in particular when architecturing the system. System architecture offers a system of reference and a framework from which a system, or a set of systems, can be constructed and organized. It enables the identification of the various components of a set of information systems, the delimitation of the services provided by each of them, and the identification and qualification of the information flows spreading between their components. Lastly, it recommends the proper way(s) for the system(s) to be constructed.

The interfaces are a sore spot and one of the first factors of fragility in any project: the architecture must help control the interfaces between subsystems from the start, in terms of functionality but also of costs and management (definition of the messages, nature of the exchanges, time limits, etc.).

The mere notion of cooperation between ITS systems demands that certain principles and rules be specified. In the simplest projects, two stakeholders negotiate those rules in private. In projects of a higher complexity, the number of stakeholders calls for those rules to be defined on a higher level, in order to apply to everyone.

To start with, the approach must remain "functional": it defines what must be done, before describing "the way it must be done". This helps take into account the technological evolutions and various modes of organization, and leaves maximum freedom of choice and optimization on the level of each subsystem.

6.1.3. *An international approach*

The first projects on intelligent transport system architecture were conducted in the United States. Europe also launched a project focused on ITS which led to the creation, in October 2000, of the KAREN framework (*Keystone Architecture Required for European Networks*). The FRAME project (*FRamework Architecture Made for Europe*) followed and led to the creation of a coherent methodological set. The documents and the framework can be found on the project's site: www.frame-online.net.

Each country has worked on its national project while following the design principles that had been defined for FRAME. Other countries, aware of the stakes of

interoperability on the efficiency of the transport networks, have designed similar tools: the United States, Canada, Japan, Australia, etc.

Aware of these stakes, the French Department of Transport, with the help of agents in the field of transports, has recently launched a project which aims at favoring the transport systems' interoperability through the definition of enterprise architecture of the information systems. This led to the ACTIF project for "*aide à la conception de systèmes de transports interopérables en France*" (the framework architecture for intelligent transport in France). ACTIF works in close cooperation with the various ongoing projects, in order to reap benefits from foreign experience and help the improvement and coordination of methods, models and tools, notably for transborder transport services. A true partnership has thus been implemented with the European architecture FRAME.

6.2. ACTIF

ACTIF enjoins general contractors of transport systems to design architecture approaches for their projects so as to enable the organization of durable systems that can more easily communicate with one another ([RIN 05], p. 27). It provides them with:

- a method to build their projects within a complex context in which several agents, several systems must communicate with one another (standardization of the information systems' architectures and their interfaces);
- a model, coupled with that method, which builds up on the collective experiences in the modeling of transport systems and their interfaces; nowadays, eight fields of activity related to terrestrial transports are modeled;
- license-free tools, simplifying the method's implementation and the use of the knowledge provided by the model.

The following functional fields are already available for use ([RIN 05], p. 27):

- a method for the architecturing and design of transport systems;
- functional models explained and well-documented;
- simple license-free tools to manipulate those models on each project.

ACTIF has standardized the architecturing approach of the transportation systems within a methodological guide. This document, aimed at general contractors and transport systems designers, explains in detail the various stages of the architecturing process along with the tools used and the resulting end products

([RIN 05], p. 27). This project management method helps ask the right questions and find the right answers, even before the projects' technical design ([RIN 05], p. 27):

- identification of the project's environment, the interfacing systems and agents;
- identification of each stakeholder's needs and constraints;
- functional description of the system: individual responsibilities and functions;
- description of the data exchanges between agents and systems.

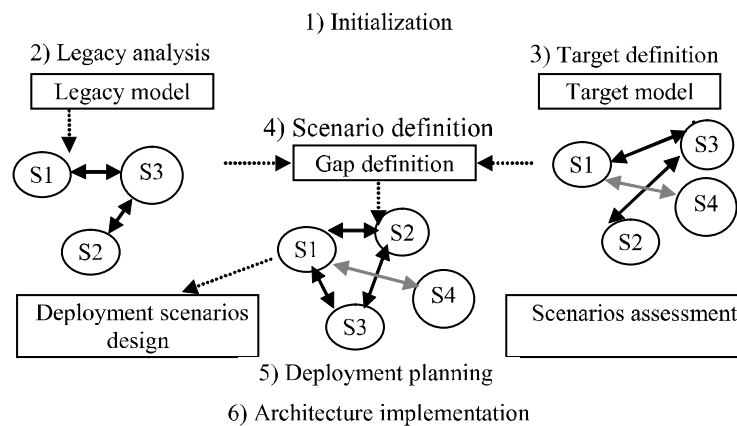


Figure 6.1. *The ACTIF method [DEN 07]*

6.2.1. *The ACTIF perimeter*

The ACTIF model is split into nine functional areas which model, using a common lexicon and grammar, the functions and information exchanged between the functions of the transport systems. This model is the product of lengthy studies and has benefited from the experience of agents from each of the concerned trades. It thus helps improve efficiency in the modeling and analysis of transport systems.

Nine main functional areas were chosen ([ACT 08], in the “Model” section):

- DF1: provide electronic payment means;
- DF2: manage emergency and safety services;
- DF3: monitor transport infrastructure and traffic;
- DF4: operate public transport;

- DF5: provide advanced driving assistance systems;
- DF6: provide information on traffic;
- DF7: enforce regulations;
- DF8: operate freight and fleets;
- DF9: manage shared data.

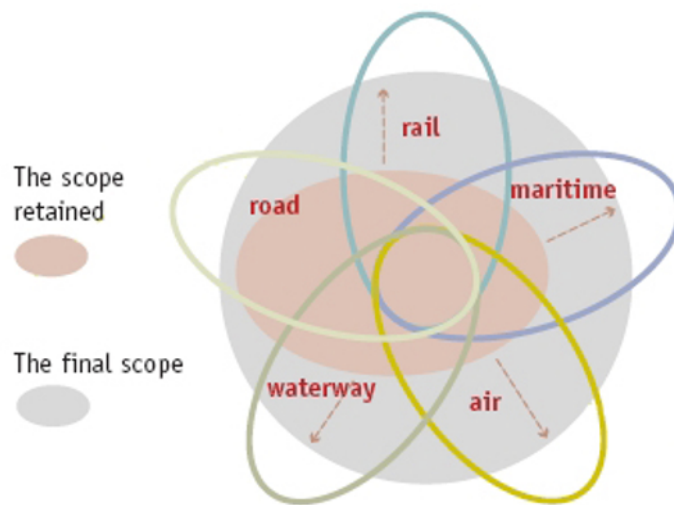


Figure 6.2. *ACTIF perimeters and fields*

Each model offers a simplified representation of the operating logic of the various activity fields and of the interfaces to be implemented between those fields and their environment.

The modeled objects are characterized by ([RIN 05], p. 28):

- the functions of data compilation, treatment and distribution;
- the external systems with whom data must be exchanged;
- the data exchanges with the external systems and between functions;
- the databases holding the information;
- the norms and standards relevant to each object.

The model can be accessed on the ACTIF site through several diagrams:

- the functional views, which present the services, the messages exchanged and the required interfaces (notably to communicate with other means of transport);
- the logical views, which provide a simpler view of the functional model of interfaces between the various fields. These groupings correspond to the functions' locations in organizations or physical systems (management and operation center, vehicles, etc.);
- the thematic views, which present, for a given process, the appropriate set of functions, messages, standards and recommendations.

A set of documents details the general modeling logic, the implementation of that logic on each trade, and the potential use of the model.

6.2.2. The ACTIF model

The ACTIF model represents the business model reference architecture for the field of intelligent transport systems.

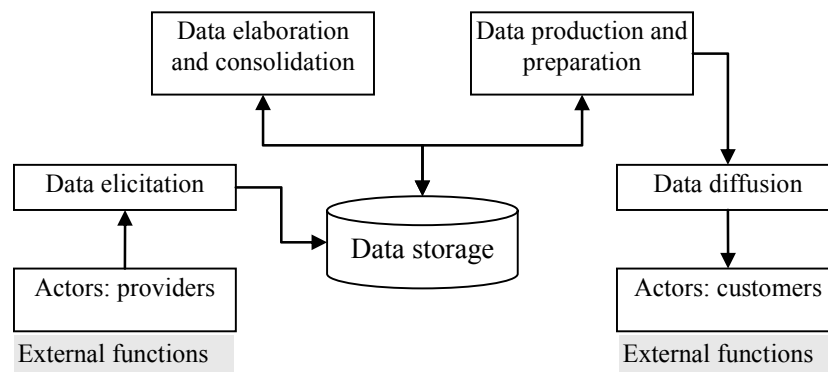


Figure 6.3. ACTIF model of generic architecture [ACT 08]

The model lists each functional field as well as their connections. The “Browsing the Model” part of the ACTIF website gives access to the model’s various components. This page aims to lay out the terminology and concepts necessary for comprehension, navigation and use of the model.

The system of systems thus created helps a general contractor define a coherent functional set.

6.3. Practical application

The entire ACTIF method-model-tool concept will be put to practical use in the design of a multimodal information system. This example brings together several functional fields and multiple agents, thus illustrating the interest of a system of systems.

6.3.1. *Context*

The various agents of a region's transport system wish to design a multimodal information system. Such a tool will provide global information about the transport possibilities through the integration of the various networks' offers, and the promotion of public transport systems to the users.

One of the goals of such a system is to favor intermodal and/or multimodal traveling habits, and thus contribute to an increase in the use of public transport systems.

Typically, the implementation of a multimodal information system involves a high number of agents (institutions or network operators) who may each have their own databases or plans for the design of passenger/client information systems.

These agents are also involved in the design of a multimodal information system as suppliers of information or as potential users of the data held within the system.

6.3.2. *Architecture approach*

During the entire elaboration stage, but also during the project management, an organizational approach called "system enterprise architecturing" or "multi-project architecture" must be implemented. This approach helps identify the agents, determine the functions and analyze their operation, notably through data flows.

The stakes of such an approach is to put forward, before complex systems are actually designed (systems which are necessarily expensive to implement and operate), the various questions that should be pondered on to structure each system's operation, in order to keep expenses to a minimum, evolve along with the technical and political environment, cooperate and exchange data.

The first step consists of identifying the agents operating on the field, such as:

- the various local administrative subdivisions: in decreasing size, the region, the general councils, the towns and community of towns;

– the authorities at a national level: the transport organization authorities (AOT, short for *autorités organisatrices de transport*), the DDEs (*Direction Départementale de l'Équipement*, the French infrastructure management office).

The second step consists of identifying the needs. Each agent shares his expectations and needs both in terms of coordination and about the implementation of the regional project.

An analysis is carried out to define the elements already owned by each agent, so as to compile, on the one hand, the mapping of the operating systems, and on the other hand, the data which will form both the stable and dynamic frameworks.

During the various meetings, the following needs are expressed:

- provide global information about the available transport means on the region's level;
- have the information circulate through means other than the Internet;
- integrate interregional or transborder issues;
- favor the use of public transport;
- favor the implementation of the transport policy;
- allow users to plan their travels with no necessary knowledge of the various operators' transportation networks;
- help have a clearer view of the demands (whether they have been satisfied or not);
- take into account the accessibility needs of users with disabilities;
- inform the users about possible disruptions;
- take said disruptions into account;
- inform the users about fares and travel costs;
- enable the exportation of the framework's data and the calculated data.

Expressing the expectations the agents have concerning the multimodal information system helps classify the various needs into modules. Each module describes a data or a set of data defined in collaboration with the agents of the target system and which might be integrated into the future information system.

6.3.3. *Modeling*

The modeling was achieved using the OSCAR software tool, provided by ACTIF.

To fulfill its purpose, and keeping in mind the previously described concepts of a multi-project architecture, OSCAR offers the following services:

- definition of the subsystems: the user creates the subsystems of his multi-project architecture, by basing them, or not, on the components of the ACTIF reference architecture;
- definition of the stakeholders: the user creates the stakeholders of his multi-project architecture, by basing them, or not, on the suggested components of the ACTIF reference architecture. The suggested components are deduced from the definition of the stakeholders, according to the data flows described in the reference architecture;
- definition of the projects: the user creates the managed projects in its multiproject architecture and assigns managed subsystems or stakeholders to them;
- definition of the links: the user selects the links among those automatically created between subsystems and stakeholders, still based on the data flows described in the ACTIF reference architecture. In the case of stakeholders outside of ACTIF, links are not automatically created. The user is therefore able to fill out his architecture by creating links, modifying the existing links or even hiding the links offered by the software;
- creation of diagrams: the user can create diagrams and have a graphic image of his multiproject architecture. Two types of diagrams are offered:
 - global architecture diagrams implement the coexisting subsystems and relationships (the links between subsystems),
 - subsystem diagrams illustrate, for each subsystem, its components in terms of functions, exchange flows between functions, as well as the links with the other subsystems in the architecture;
- generation of documents: the user can generate different types of documents in order to translate the whole set of data within his created multiproject architecture.

The tool's purpose is to enable the declension of the generic ACTIF components into "real world components", corresponding to systems that are already in use, or will be implemented for the users. This requires the introduction of the following concepts:

- a multiproject architecture: a multiproject architecture defines the components and data exchanged between the existing or planned ITS systems for well-defined applications within a well-defined geographical zone (city center, urban area, department, region, etc.);
- subsystems exist within the multiproject architecture: the subsystems are first defined from the functional subdomains, then from the framework architecture's functions and databases. Since these functions are associated with the user needs within the framework architecture, a list of the fulfilled user needs can be deduced for each subsystems;
- stakeholders exist outside of the multiproject architecture: the stakeholders are defined from the ACTIF terminators and the functional subdomains which contain functions linked to functions of the multiproject architecture's subsystems;
- projects exist outside of the multiproject architecture: a project is a grouping of subsystems and stakeholders, within or without ACTIF;

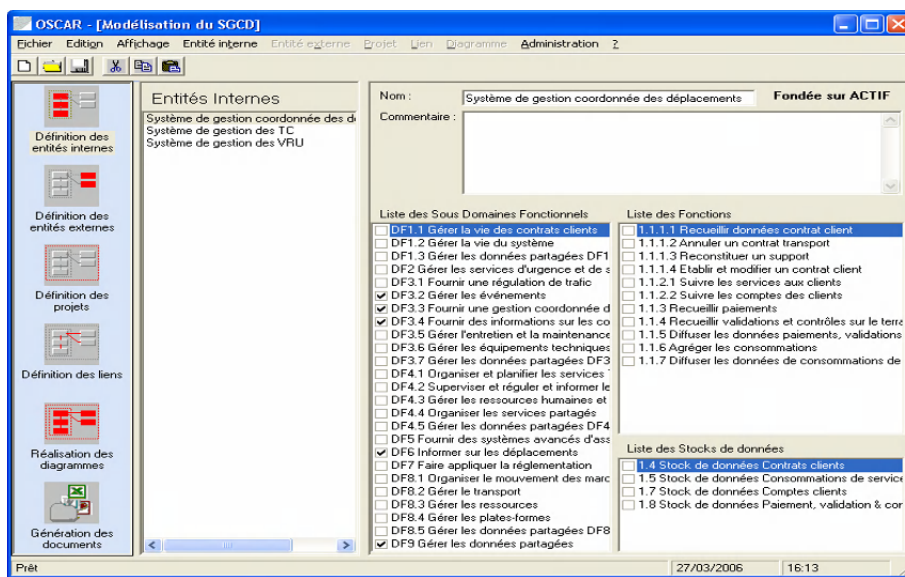


Figure 6.4. Modeling within the OSCAR tool [ACT 08]

- links exist between two entities (subsystems or stakeholders): a link is a directional flow between two entities, modeling the presence of an interface between these entities. It groups the framework architecture's logical flows which exist between these entities (more precisely, the logical flows created by an entity's functions or its related external systems and sent to another entity's functions or

related external systems). Traceability between the logical flows of ACTIF gives access to the standards. The subsystems/stakeholders are linked by data flows which must also be instantiated. The connections between entities will be represented by the links, formed by the ACTIF flows, and this traceability will give access to the ACTIF standards.

Figure 6.4 displays a window of the OSCAR tool including, for a given function, different sections, among them the stakeholders with whom the function interacts, the list of the functional subdomains, the list of functions and the list of databases.

6.4. Conclusion

The concept of systems of systems is particularly essential in the world of transport, where many public and private agents are sharing data and managing services aimed at the users.

The user, customer of the transport of passengers or goods, stands at the heart of functional domains with high interoperability demands.

To achieve ever higher performances within a fully developing competition, administrators have had to develop new systems. The implementation of the most accessible systems for customer use, such as information and computer ticketing, has helped accelerate the establishment of common standards.

The French Department of Transport's answer was ACTIF, a system of systems applied to Intelligent Transport and assisting the design of interoperable transport systems. This method-model-tool set is a good concrete example of the concept's application.

6.5. Bibliography

- [ACT 08] ACTIF WEBSITE: <http://www.its-actif.org/>, 2008.
- [DEN 07] DENIS Y., "ACTIF : Aide à la Conception de Systèmes de Transport Interopérables en France", *Journée PREDIM*, Grenoble, 22 October 2007.
- [JAN 07] JANIN J.F., *ACTIF*, Departement of Transport, Equipment, Tourism and Sea, Lyon, 2007.
- [NAR 04] NARDUZZI F., *Diagnostics de projets avec la méthode ACTIF*, SETEC ITS, 2004.
- [RIN 05] RINIÉ E., "e-quippement : application des technologies de l'information et de la communication", Conseil général des ponts et chaussées, report n° 2004-0185-01/1, Lyon, 15 June 2005.

Chapter 7

Systems of Systems in the Healthcare Field

7.1. Introduction

The medical field is undergoing major changes. Clinical exams are more and more often instrumented through the implementation, direct or indirect, of computerized systems. The same applies for the administrative data exchanged between hospitals and social security administrations when you (the patient, us!) are hospitalized. The ophthalmologist runs a digitalized dilated *fundus oculi* examination, which he stores in the patient's file, in his computer. The patient uses his health insurance card so the health professional can create a file for the reimbursement of the medical procedures by social security, and pays the pharmacist with the card of his complementary health scheme. Those are only the most visible aspects of the evolution, rooted in the information world, which the medical field is currently going through.

The medical field is characterized by the diversity of its organizations (public or private hospitals, pharmaceutical industry, technical medical systems industry, health insurance funds, etc.), its agents (health professionals, working in health facilities or private practices), these agents' activities (medical or paramedical agents, social workers, etc.), and the technical systems supporting their activities. The diversity of organizations, agents, activities and technical systems, favors their interaction, to exchange data, design healthcare networks, record the procedures undergone by the patient through his entire life and manage the healthcare systems.

We can already see the outline of the services which the systems of systems – independent systems which, when interconnected, provide new services and enrich the value chain – could be expected to provide in the medical field.

This chapter is broken down into five parts:

- the first part defines the challenges, in terms of capability, around which systems of systems are designed in the medical field;
- the second part characterizes the medical field's systems in terms of services provided;
- the third part studies the coordination of the agents' activities within the healthcare networks;
- the fourth part looks at the development of the information technologies and their interoperability, at the heart of the issue of healthcare networks;
- finally, the fifth part studies the difficulties met in the field.

This chapter was elaborated from information sources and interviews (in alphabetical order): Karima Bourquard, Yves Constantinidis, Claude Pourcel, Jean-Claude Sarron and Michel Veret.

Currently, many initiatives are led in this field and this chapter does not claim to cover them all, but rather to shed light on the issue of systems of systems in the healthcare field.

7.2. From capability challenges to the design of systems of systems

Population development (increasing number of dependent elder people), societal (refocus the healthcare system on the patient) and financial (reduce the costs of healthcare systems and optimize the use of resources) evolutions, as well as the development of healthcare networks, all require the global healthcare system to undergo major changes.

Indeed, the efforts in placing the patient back at the heart of the healthcare system, the shortage of practitioners, the search for the continuity of service and the implementation of information technologies bring about a search for increased capability in order to:

- improve patient care quality, whether on a medical level (quality of care) or a social level, as well as improve quality of life (for example by helping the patient stay at home whenever possible);

- improve the coordination of the various medical and paramedical agents, working in hospitals or private practices, who contribute to the medical care services, through the optimization of processes and channels;
- successfully manage, optimize and use hospital resources (to fully profit from an expensive scheme);
- garner expertise wherever it may lie, without its localization penalizing patient care quality;
- successfully handle the ageing of the medical profession;
- improve the care's cost/performance ratio.

This heightened capability can be achieved through several means, including the development of healthcare networks outside of the hospital, and the integration of the various healthcare activities within one continuous service, focused on the patient. This can be achieved through the implementation of information technologies.

Our discussion is in the context of systems of systems, such as defined by Maier (Chapters 1, 2 and 4). Indeed, the agents of the healthcare networks, the health facilities, the general practitioners, etc., are independent on an operational level, each with their own activity not dependent on the others'. They are also independent on a managerial level, since physicians can work in private practices, and health facilities can either be public or private organizations. The medical and paramedical agents are scattered within a more or less important geographical and demographical basin. Network-centric operation enables the provision of services which no individual agent could deliver. These networks are organized thematically: pediatric networks, oncology networks, etc. Lastly, in order to provide their services to the patients, the medical and paramedical agents must coordinate their activities and exchange medical and paramedical information about the patients.

This system of systems features a major and structuring characteristic. Its purpose is to provide services to the patients, who are an integral part of the healthcare process.

Figure 7.1 places the patient at the heart of the healthcare process, which features a set of health professionals, working either in hospitals or private practices. A network can feature several hospitals and several private practitioners within one population pool.

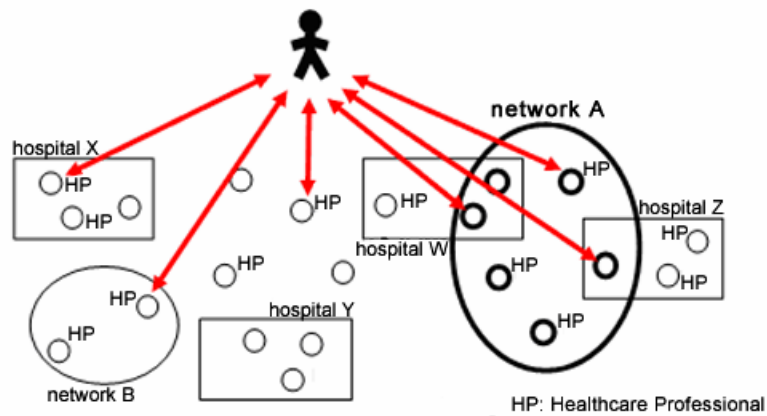


Figure 7.1. Patient at the heart of the healthcare system (source [GIU 01])

If a network can feature several health facilities, a hospital can contribute to several networks. Likewise, private practitioners can contribute to several networks, as demonstrated in Figure 7.2.

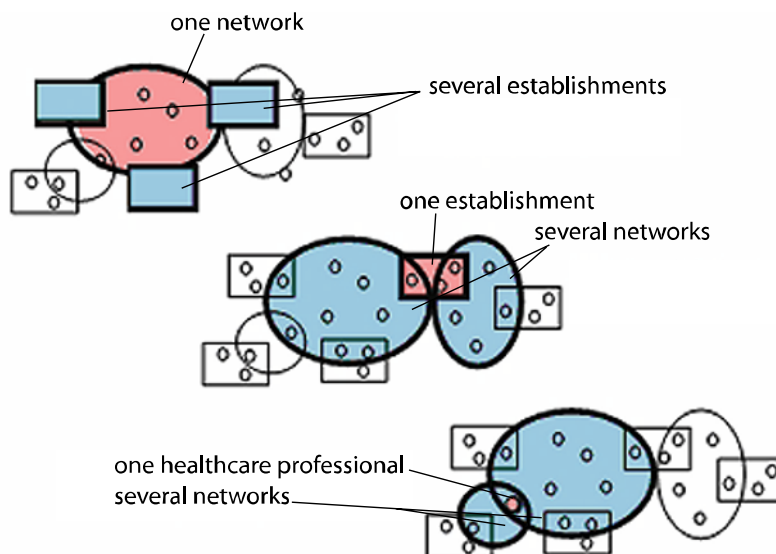


Figure 7.2. Hospitals and health professionals within and at the intersections of the healthcare networks (source [GIU 01])

We delimit three sets of themes to study in order to achieve those goals and improve capability:

- the service provided to the patient, the main characteristic of systems in the healthcare field;
- the coordination of the various medical and paramedical agents activities, on an organizational level, in order to provide that service;
- the development of information technologies, to support such coordination and exchange information.

To this purpose, the information systems of the various agents of the healthcare network must be interoperable.

These are the three themes we are about to discuss.

7.3. Personal service, the main characteristic of systems within the healthcare field

If we look back at the definition of the notion of system that we have given in Chapters 1 and 4, a system is characterized as being a product or service, implemented by the final user, the processes and activities needed to supply that product or service, the resources needed for its design, its production. Moreover, we differentiate the “system to do” and the “system to be done”. The former qualifies the product, as well as the related data. The latter qualifies the set of means, processes, methods, tools, which need to be implemented in order to do, to realize the system to do.

With the healthcare field, we are not in a context of industrial production, traditional of engineering approaches, but in a context of service supply, currently identified as pertaining to service science, management and engineering ([ABE 05, HID 06, SPO 06]).

The service is therefore defined as a system of interacting parts, featuring people, technologies and commercial exchanges, with the user in direct contact with and participating in the service. Service science is multidisciplinary, calling on disciplines such as anthropology, cognitive psychology, information sciences, cognitive sciences, science education, human factors, industrial engineering, organization sociology, law, mathematics, economy and social sciences.

Abe ([ABE 05], p. 11-12), differentiates four types of services:

- services centered on specialized skills and knowledge with little connection to commodity goods, such as porters, hairstylists, gardeners, teachers, accountants;

- services which provide commodity equipment, or even goods, such as performing arts, theaters, movie theaters, museums, restaurants and caterers;
- financial services;
- services which provide information not manipulated by human beings, such as the storage or supply of information by press agencies.

In that context, the healthcare services belong to the first and second category: they include, on the one hand, services centered on the physician, the paramedical specialist, and on the other hand the services which call on a convalescent home, medical laboratory or emergency call center.

The main characteristics of a service are ([SPO 06]):

- its intangible character, which means the service provided is most often immaterial, not physical;
- simultaneousness, insofar as it is simultaneously created and used;
- its perishable character, since the service is used as soon as it is created and is therefore not preserved;
- the customer, the service's user, actively contributes both to the creation and the use of the service;
- the difficulty to determine the quality of the service: indeed, the service's quality is closely dependent on the interaction between service provider and service user;
- heterogenousness: the same service can have different results, depending on the provider and the state of the service's user.

What are the impacts of this "service" dimension in the healthcare field?

The service's user, that is to say the end user as defined by engineering, is also the "product", which we have previously qualified by "system to do". Indeed, the purpose of the medical procedures is to modify the state of the patient (the "product"), to "fix" him. But this "system to be done" is radically different from the products, such as they are designed in the industrial field. There is no engineering, no design of a human "product". The human "system" is intrinsically and radically different from the object system of systems engineering. There is therefore no expression of need, specification, design, development, integration, inspection, validation, in relation to the documents concerning the specification, design, operation and disposal process.

Moreover, the patient is an autonomous subject. He actively participates in the medical service, taking or discarding the medication prescribed to him, abiding by the prescribed diets or going on with dangerous behaviors, etc. Indeed, the medical agent is not alone in providing the service which will modify the patient's health status. The patient is co-responsible and a co-producer of this service. This service is constructed within an intersubjective relationship between the medical agent and the patient, who is at the same time the end user, the product to transform and a contributor to the service. In some way an attachment between the practitioner and the patient, a relationship of trust is built. The intimately intersubjective character of this relationship has important impacts, on the service's realization as well as its quality, and finally on the patient's health.

In the healthcare field, the "system to be done" is made up of the set of agents within the medical field who contribute to the service's realization. We can therefore differentiate two main types of services: on the one hand, general practitioners, and on the other, hospitalization. These agents have organizational structures, operating modes, processes and process instrumentations, all completely different from one another. This aspect helps structure the design of the system of systems, and we will study it in section 7.4.

Just like the engineer uses data related to the product, the medical agent designs, updates, exchanges, stores data about the state of the patient, of the "product". This is the patient file. This information also includes components typical of the implemented processes and procedures. For example, the results of a medical analysis include the protocols and developers, necessary elements to the interpretation of these results, the same way oral and rectal temperatures are both treated differently. All the information included in the patient file is unique to each individual and pertain to medical secrecy; as such, their protection is a critical factor in the elaboration of technological solutions to facilitate their treatment, storage, or exchange. The communication of these medical data must therefore be secured through mechanisms that will ensure their authentication, confidentiality, integrity and non-repudiation (see Chapter 12, section 12.21. Appendix L). The use of such personal information and the development of tools to treat them must be filed with the CNIL¹. Moreover, each patient's ID must be unique. Finally, this information remains important during the entire life of the patient, and must therefore be conserved and available all through his or her life, currently more than 100 years².

1. CNIL: Commission nationale de l'informatique et des libertés (French Data Protection Authority).

2. The current life expectancy of a French citizen is higher than 81 years old, according to the numbers published in March 2008.

The data concerning the patient are pertinent for a set of agents who participate in the medical or paramedical acts, or even, sometimes, for social workers. But in these situations, the agents only have access to subsets of these data.

Moreover, if the medical data and their exchange form the backbone of patient care's continuity, the multidimensional character of the medical act means there are many more information exchanged. Besides the individual medical data (description of the medical and surgical procedures, analysis results, etc.) and the information exchanged during teleconsultations, telesurgery or telesurveillance, all pertaining to medical secrecy, there is also a non-negligible administrative part (records of performed medical procedures, billing, fee-for-service), as well as collective information of an epidemiological nature, gathered from the individual medical data, turned anonymous and aggregated, which can lead to health hazard reports and do not pertain to medical secrecy since they are no longer referencing anyone in particular.

7.4. Coordination of the medical and paramedical agents, in hospitals and in private practices

In order to ensure the continuity of the care given to a patient, the various medical and paramedical agents must coordinate their efforts, before hospitalization, during hospitalization, in all relevant departments (intensive care, radiology, surgery, specialized departments), and after hospitalization (follow-up, home care).

As we have pointed out, these various agents have different social statuses, from highly structured organizations such as hospitals and rehabilitation centers, to private practices and general practitioners, statuses which have an impact on the agents' activities, and on their way of managing activities. As a result, it also has an impact on the coordination capacities of these various agents. What does this coordination consist of?

The hospital is an organization, with its own structure, its roles and responsibilities, its operational services and auxiliary services, such as defined by Mintzberg (see [MIN 82] and Chapter 4, for a summary of Mintzberg's works and a definition of the various structures studied in the present section). This is not the case with general practitioners (doctor, nurse, physiotherapist, medical laboratory, etc.) who are numerous and, most often, either have a liberal status, sometimes working within private practices, or are categorized as small office/home office (SoHo), or small and medium enterprises (SMEs).

Within a health facility, the processes, the activities, are interlinked and governed by the state of the patients. A patient's discharge is another patient's

admission and vice versa, which calls on intensive technologies ([ROJ 03], p. 134, [THO 04], p. 17-18) and reciprocal interdependence ([ROJ 03], p. 136, [THO 04], p. 55). These are complex and dynamic processes such as described by Thompson [THO 04] and Mintzberg [MIN 82]. These characteristics determine the structure and operation of a health facility.

The hospital, according to Mintzberg's model, is mainly a professional bureaucracy. The formation required to work there on an operational level is long (medical studies or nursing schools). The work protocols are strict. The hospital is often structured into independent departments, one for every specialty (cardiology, pediatrics, obstetrics, ORL, etc.). Most of those departments tend to function autonomously, in order to reduce contingences, and are managed by a doctor, a chief resident. The development of a cluster-based organization is making this structure less automatic. Each department is then organized following a simple structure. The contingencies are reduced through the search of total control of the patient, following the logic of a total institution such as described by Goffman ([ROJ 03], p. 135, [THO 04], p. 43). The activities of the transversal departments, such as radiology, intensive care, or the internal medical laboratory, are highly dependent on one another, but these departments also define a set of constraints for the other departments, particularly in terms of resources management and use. These departments are not often in direct contact with patients, and the aforementioned characteristics do not concern them as much.

In that way, the medical laboratory, which is highly automated, would be more akin to a mechanistic bureaucracy. The strategic summit of the hospital, on the other hand, usually is not managed by a doctor but, in the case of French public health facilities, by an administrator. It is possible to have, on that level, a divided structure. Strong social differentiations can be found within the hospital, in terms of roles, responsibilities, and practices between each agent. This differentiation also translates into privileged social networks, as much within the hospital as without, in everyday life as well as in organizational and professional culture (Chapter 4).

We have seen how the patient's active participation in the realization of the service is characteristic of the medical field, and possesses a strongly intersubjective dimension. If this intersubjective dimension is most important in the case of general practitioners, and allows for the creation of an enduring relationship between patient and practitioner, it is reduced and framed in the case of hospitalization. Indeed, if the hospital is looking for total control of the patient, the drawback is the symbolic and ritual depersonalization of the patient, such as is described by Goffman [GOF 79]. In the total institution represented by the hospital, the patient is cutoff from the outside world, completely taken care of, sometimes down to the clothes he can wear, the visiting hours are strictly regulated, the contacts between medical personnel and the patient are formalized and formatted (the health sheet at the bottom of the patient's

bed); finally, the hospital has its own temporality, notably for dining hours. This rationalization of the patient goes against the current promotion for a higher quality of life, including in the medical field, which fights in favor of home hospitalization.

Within the hospital, the coordination needed to implement the processes is based on a set of exceptional operational decisions ([MIN 82], p. 76), ad hoc decisions which are not programmed, not planned. The regulation of activity is done on a case by case basis. This is possible insofar as the patient is on the premises and available. Works are currently led on such planning and scheduling (for further information, consult the reports of the annual French GISEH conferences, “management and engineering of hospital systems”).

This is not the case, however, with general practitioners. The activity of general practitioners, medical or paramedical, is governed by their patients’ health constraints (some health services must be performed at a precise hour) and personal constraints (availability). In such a context, the coordination of their activities must be flexible so as to adapt to each other’s activity, said activity being governed by many contingent and unplanned events, and mutual adjustment systems must be implemented, such as liaison mechanisms and integrating frameworks. On the other hand, the granularity of the concerned organizations, whether they be SoHo or private practices, does not allow for the implementation of other solutions of mutual adjustment such as, for example, matrix organization.

The main components of this coordination can be formulated in terms of service request (request for medical analysis, prescription of lung x-rays, request for hospitalization, prescription of ten orthoptic sessions, etc.), replies to these requests (medical analysis results, etc.), and may include temporal or periodical constraints, as well as constraints on the way to provide the required services. Lastly, these coordinating components require the communication of medical data between the agents (medical analysis results, x-ray photos of the patient’s lungs, etc.). This medical data can be stored on an analogical medium (fax of an electrocardiogram or the results of a biological analysis) or a digital medium (barcode bracelet, digital plate, RFID³ chip for the patient file, electronic message with an attached JPEG file of an x-ray photograph, etc.).

On top of these formal aspects, the informal dimension of interpersonal relationships is necessary for coordination of the medical and paramedical agents’ activities. For example, a general practitioner can also work within the internal medical department of the hospital present in his area. His knowledge on how the

3. RFID (*Radio Frequency Identification*): this type of chip is associated with an antenna and the set constitutes a radio frequency tag, a small object enabling the reception of and answer to radio requests sent by a transceiver.

hospital functions, as well as the good relationships he maintains with the doctors and the administrative staff, allow him to more easily orient patients in need of hospitalization. This accelerates and improves the coverage of patients within the hospital.

7.5. The development of information technologies and their interoperability, heart of the healthcare networks issue

We are now about to study the implementation of information technologies in the healthcare field and will then look at three major standards used to achieve interoperability of the information systems, and an example of their implementation.

7.5.1. *Information technologies in the healthcare field*

Information technologies help design and instrument the processes and activities which contribute to the production of care. But the situation of the medical field impacts on their definition, their development and their instrumentation.

While industrial processes can be clearly and strictly defined, organized and planned, within a logic of modeling work processes for automation (Chapter 12, section 12.8 Appendix A), such is not the case for most processes contributing to the healthcare system, insofar as the patient is directly concerned.

If hospitals implement information technologies ([GMS 07a, GMS 07b, GMS 07c]) in a structured manner, the situation is different for the medical and paramedical general practitioners, whose activities are poorly computerized. The website for the ENOSIS project states that in 2003 “less than 30% of general practitioners use a computerized medical file on a daily basis” [ENO 08].

In his 2006 report on the state of information systems in health facilities [GMS 07c], the GMSIH (*Groupe pour la Modernisation du Système d'Information Hospitalier* – Hospital Information Systems Modernization Group) defines the information perimeter, through the implemented process and the object of that process:

- the patient data system:
 - patient’s administrative management,
 - shared patient file,
 - medication’s circuit,
 - management of examination requests,

- management of examination results,
- management of the medico-technical units,
- emergency management;
- the economical, financial and logistical information system:
 - financial system,
 - the other functions (patient transportation, catering, sterilization, computer-aided maintenance management);
- the human resources management information system;
- the quality and risk management information system;
- the piloting information system;
- communication tools;
- secured exchanges with the outside world.

The evaluation report results, feedback and advice on the theme of transversal nature and master plan of the information system [GMS 07b] lays out results which we might synthesize with the following elements:

- there often are inconsistencies between the master plan, on the strategic level, and the implementation, on the tactical level, with poor involvement from the users on the operational level. On the other hand, the successful projects are characterized by a continuity between the strategic (implication of the management, of the CME⁴) and the tactical level, and by the users' involvement;
- the project is rarely based on an analysis of what already exists, even though such an analysis is necessary for the integration between the current situation and what will be achieved within the project;
- oftentimes, there is no mapping based on the workflows, no architecture framework featuring the operational views;
- even if the standards of interoperability are mentioned in the master plan, sometimes in the specifications, the implementation of those standards, including the IHE integration profiles (which we will study shortly), is not explicitly required. This poses an important problem for the patient file which is, by nature, transversal.

The lack of users' involvement has important negative impacts on quality and safety, such as violations and migrations ([HAR 2002], [AMA 09]), risks on patient safety, and considerable extra healthcare costs ([LER 09], [PEL 09a], [PEL 09b]).

4. CME: *Commission médicale de l'établissement* (Facility Medical Committee).

We have seen the importance of interoperability in the healthcare field, a necessary condition for the information flows to correctly circulate between the sector's various organizations. How is this interoperability managed?

7.5.2. Interoperability in the healthcare field

Since the information systems are heterogenous, interoperability lies in their ability to exchange services or data. These services or data are subject to standards, which we are about to study.

7.5.2.1. The IHE initiative

The IHE, for *Integrating the Healthcare Enterprise*, is an international organization, structured into networks, which allows the users and designers of information technologies in the healthcare field to reach a higher level of system interoperability through the precise definition of the activities within this field, the communication specifications based on the norms and standards between systems, and the testing of systems to determine their conformity level in relation to those specifications.

The IHE currently covers the following domains:

- cardiology;
- eyecare;
- IT infrastructure;
- laboratory;
- pathology;
- patient care coordination;
- patient care devices;
- quality;
- radiation oncology;
- radiology.

Integration profiles are developed for most of these domains. They are documents describing the solutions to specific integration problems, documenting the roles of the agents involved, and defining the details of the systems' implementation. They ensure that everyone is talking about the same thing, without needing to breach the many technical details which in fact contribute to the interoperability.

For example, in the field of IT infrastructure, specifications or technical frameworks deal with digital signature, the exchange of clinical documents between healthcare organizations, the administrative management of patients and the directory of the personnel.

The technical frameworks define with precision the technical details of the integration profiles.

The Connectathon is a major characteristic of the IHE. This is a conference dedicated to testing the conformity of the technical solutions, in order to evaluate their level of interoperability. These conformity tests are based on real life medical scenarios and use the IHE integration profiles. This event is organized on a worldwide scale and is held, in turn, in various towns in Europe, North America or Asia: such as Chicago in February 2009, Vienna in April 2009 and Tokyo in February 2008. The Connectathon largely contributes to the credit and the broad distribution of IHE among the healthcare field's industrialists. The organization of such an event is favored by the strong involvement of institutions, such as the INRIA in France, which provide personnel for the event. Finally, the evaluation platforms, based on web services, are easy to implement. In that context, the industrialists follow that approach and regularly perform interoperability evaluations, in-between Connectathon sessions.

To ensure the spreading of good practices and the partners' adhesion, the IHE uses a set of resources in the form of a dedicated site (<http://www.ihe.net/>), and a *wiki* (<http://wiki.ihe.net/>). The *wiki* features many pages and many documents, technical frameworks and integration profiles.

7.5.2.2. The HL7 standard

The HL7 standard (Health Level Seven) is an exchange language located on the 7th level of the OSI (application level), hence its name, designed within HL7 Inc., taken in by the ANSI (American National Standards Institute) and used in many countries. The conceptual data model HL7 v3 is now an international ISO (International Organization for Standardization) standard, filed under ISO/HL7 21731:2006, under the title "Health informatics – HL7 version 3 – Reference information model – Release 1".

The standard is designed by a set of technical committees (digital health-related record) and groups with specific interests, work-related (anatomical-pathology, pediatrics, cardiology, etc.) as well as technical (service-oriented architecture, etc.).

Version 2.5 features the following chapters and appendix:

- introduction to the HL7 standard;
- control: definition of the messages and the exchange protocols;
- patient management: admission, discharge, transfer, demographic elements;
- service order entry: clinical orders, orders of observation, medication, blood, imagery, dietetic food, suppliers;
- queries: the rules for the formulation of queries, and the answers to those queries;
- financial management: patient-related accounting;
- observation reporting: observation report messages;
- master files for health applications;
- management of the medical records and medical data;
- scheduling;
- referrals;
- patient care: problem-oriented records;
- clinical laboratory automation;
- application management;
- personnel management;
- data definition tables (appendix);
- low-level protocols (appendix);
- description of BNF (*Backus–Naur form*) messages (appendix);
- terms glossary (appendix);
- administrative aspects, such as payments, complaints and compensation requests (appendix).

Version 3 defines a set of formatted messages and includes a reference information model, a model of healthcare data, under the formats XMI, Rose, Visio, and a reference architecture of a medical file, CDA (Clinical Document Architecture) based on the XML computer language (Chapter 12).

The CDA is derived from the RIM. It allows the organization of documents with headings, highly structured so as to facilitate their indexing and sorting (document type, authentication level, confidentiality level, origin), and a body, a set of sections featuring text, images, sound, as well as multimedia content. Among other things, it treats the semantic representation of clinical events, of the patient's vital signs (temperature, pulse, saturation rate, etc.). The sections can be, for example, medication, patient allergies, and family history.

7.5.2.3. *The EHR standard*

A set of alternative standards are elaborated by the CEN (*Comité européen de normalisation*, European Standardization Committee), under the reference CEN prEN 13606, under the name "Electronic Healthcare Record Communication" (EHRcom). This standard is present on the ISO level under the reference ISO 13606-1:2008, with the title "Health informatics – Electronic health record communication – Part 1: Reference model". Its recent publication (2008), as well as the equally recent publication of other standards by ISO (ISO 17090-3:2008; "Health informatics – Public key infrastructure – Part 3: Policy management of certification authority"; ISO 11073-90101:2008; "Health informatics – Point-of-care medical device communication – Part 90101: Analytical instruments – Point-of-care test") demonstrate on the one hand the stakes of the exchange of healthcare data, and on the other hand the need for coherence between the various standards.

This standard CEN prEN 13606 is made of four volumes [EHC 03], respectively dealing with:

- the extended architecture, which describes the architectural components: the file, to keep a person's health data under a digital format, and the file's components, including:
 - compositions, dependent on the place and the moment of care (e.g.: home visit report, radio exam results, etc.),
 - directories which group the data concerning a person's health through time, or the data linked to a specific problem (a child's personal health record),
 - the data's visual synthesis (list of health problems, growth curb),
 - the links necessary to connect two file components (cataract – [aggravating factor]- diabetes; hepatic biopsy – [record of]-jaundice);
- the list of field terms provides tables of names and values which can be used to categorize the above components;
- the distribution rules define the rules for security and for the access to the personal file (who: the family doctor; why: medical certificate needed for sports, etc.);

- the messages for the exchange of information, message of file request, transfer message, notification of the exchange's status.

The standardization works are still ongoing. Thus, the technical committee ISO TC 215⁵, dedicated to computer sciences in the healthcare field (*Health informatics*) features workgroups on the following themes:

- data structure;
- data exchange;
- semantic content;
- security;
- health cards;
- pharmacy and medicines business;
- business requirements for Electronic Health Records.

Within the structure of the CEN/TC 251 Working Group IV, “Technology for Interoperability”, we find such themes as⁶:

- interoperability of healthcare multimedia report systems;
- interoperability of medical devices within acute care units;
- evaluation of physiological analysis systems;
- healthcare information system architecture.

These works standardize the exchanges and/or the mediums of health data. Thus, if the EHRcom standard offers a message perspective, the CDA (Clinical Document Architecture), on the other hand, offers a document perspective.

A document model helps elaborate a shared medical file as much as a personal card. For example, the CDA does not enforce a particular medium or a physical architecture. Moreover, for people in temporary need of mobility, such a solution allows the downloading of some of the shared medical file onto cards, which those persons can carry on them.

7.5.2.4. *The SNOMED nomenclature*

Beyond the structure of the messages and documents, the communicated data must have the same, non-ambiguous meaning for the systems which exchange or

5. Website: http://www.iso.org/iso/standards_development/technical_committees.

6. Website: <http://www.tc251wgiv.nhs.uk/pages/work.asp>.

share information. This pertains to semantics and terminology. It is the goal of the SNOMED (Systematized Nomenclature of Medicine) project, a multiaxial, hierarchical classification system, featuring 364,000 concepts⁷. It covers the greatest part of medical data, such as diseases, semiology, protocols, and pharmaceutical components.

It allows for the indexing, storage, classification and organization of the contents of medical records.

These hierarchies include [CAP 07]:

- the clinical findings represent the determined trouble, the agent responsible for it, its severity, etc.;
- the procedure explains how to treat the trouble, with a differentiation between invasive and non-invasive procedures, for example biopsy or excision, but also the technique used, etc.;
- the specimen characterizes the collected specimen (urine, etc.), as well as its morphology (cyst, abscess, etc.);
- the body structure specifies the concerned structure (thyroid gland, stomach, etc.) and positions in terms of laterality (left, right, right and left, unilateral, etc.);
- the pharmaceutical or biologic products characterize the active ingredient, the dosage, etc.;
- the situation with explicit context indicates that a medical recording is affected by the context of said recording ([CAP 07], p. 4-24): indeed, “breast cancer” can be used to describe a family history of breast cancer, a personal history of breast cancer, and finally the actual diagnosis of breast cancer;
- the events define the temporal dimension of clinical characteristics (occurrence, sequence, etc.);
- the physical force characterizes mechanisms of injury;
- the physical object characterizes man-made objects, such as a pair of rubber gloves, an artificial kidney, an implant;
- the observable entity is used to describe what can be observed, without it being a finding, e.g. the type, the age;
- the environments and geographical locations concern the places (a country, a region), the premises (a unit of intensive care);

7. Website: <http://www.nbirn.net/research/ontology/snomed.shtm> (downloaded on March, 4 2008).

- the social context concerns the social conditions which can impact health, such as family status, work, lifestyle;
- the organism characterizes a living organism, for example a virus, a bacteria, a mycosis, etc.;
- the substance helps characterize components such as food or chemical allergens, toxicity or dental porcelain material;
- the staging and scales are used to describe phenomena, such as the Stanford-Binet Intelligence scales, the Glasgow coma scale, obstetrics staging;
- the linkage concept helps define relationships, associations, such as link assertions;
- the qualifier value helps characterize an element, a phenomenon (strong, soft, bitter, etc.);
- the record artifact provides information, such as ECGs, EEGs, etc.

If the elaboration of ontology in the medical field, which would enable the semantic interoperability of information systems, is necessary, such a medical ontology could not cover the subtleties of human communication.

Indeed, such ontology used by a data treatment system, does not take into account cultural aspects, source of variations, or the pragmatic dimension typical of human communication.

To be useful to health professionals, the information systems must feature labels and data which those professionals understand, control and manipulate. Information systems often differ.

Such ontology has the quality of being a pivot language, a tool for the correspondence of various information systems.

Whatever the chosen technological solution, it will not be able to take into account and express the richness of human communication.

7.5.2.5. *An example of appropriation and implementation*

If standardization is a necessary stage, it must be completed through local, regional, national or even international actions, with medical and paramedical agents implementing standards, leaning on tools which respect these standards, and through the industrialists, who design their systems in conformity with these standards.

These are the goals and activities of the ENOSIS (*Echanges Normalisés, Organisés et Sécurisés des Informations en Santé* – Standardized, Organized and

Secured Exchanges of Health Data) project [ENO 08]. This association regroups 15 healthcare networks.

The objectives of the ENOSIS group [ENO 08] are:

- “to inform the networks’ agents, and healthcare professionals in general, the industrialists of medical computer sciences, and the institutions, on the importance of adopting communication standards in the healthcare field;
- to standardize the cards or files of the group’s networks into messages, while following the standards recommended by the standardizing structures, and develop tools that will assist standardization;
- to create common specifications, and develop a client and exchange server, standardized, organized and secure, for the health data meeting said specifications. These products’ architecture is modular, so as to help the existing tools adopt those standards.”

In this context, the ENOSIS group develops a server of standardized, secure messages, as well as client software [ENO 08], by leaning on the HERcom standards (for example CEN prEN 13606) and MMF/XMF for the compatibility with the Sentinel service of Cégétel.rss (secure medical messaging). Finally, in order to uphold the security of personal medical data, the ENOSIS project uses the professional health card (CPS, *Carte Professionnelle de Santé*, Professional Health Card) and calls on the services of the CPS GIP (*Groupeement d’Intérêts Publiques*, Grouping of Public Interests), such as the phonebook, the revocation list, the publication of public keys.

The aforementioned requirements on coordination (section 7.4) are met through a system of collaboration. Collaboration consists of a sequence of requests/replies between two or more agents in the network, a supplier and an applicant.

Moreover, several collaborations between different agents can be grouped within a single process ([ENO 08], section “The ENOSIS approach”). Collaborations may be modeled in terms of *Conversation for action* following the model of language acts ([ENO 08], section “Enosis. *Work flow* and exchange modeling” and “Collaborations and processes: the structuring concepts of the organization of communication, following the Enosis approach”).

From the standpoint of the doctor, agent within a network, the tools developed within the ENOSIS project will help him ([ENO 08], section “The ENOSIS approach”):

- choose a card model among the models designed by the network;

- write down the data within a form corresponding to that model, from a browser, offline or online;
- securely send the filled-in form to the coordinator or another network member (an Internet connection is then imperative);
- securely consult the cards which have been sent to him, or concern one of his patients (on the condition that the network is managing that particular care relationship);
- get a visual of the operating statistics, subject to rights of access;
- export all or part of the database's files, depending on the access rights, in order to run the medical data through particular statistical treatments within an adapted software (Excel, Access, SPSS, Statistica, etc.).

An experiment in actual use was led with the digital personal health records of children.

The results ([ENO 08], section “The ENOSIS approach”) show that:

- very strict protocols and a highly ergonomic interface are needed for the parents and the delegation to control confidentiality;
- the use of the CPS seems necessary to authenticate the liberal health professionals, considering the potential scope of the health record's use;
- the coexistence of both the paper and digital personal health record must be managed in a flexible, economical way, which is still to be designed;
- in daily use, the real-time data entry will require the use of some technologies which are still quite new (tablet PC, Wifi);
- the data entry must be possible offline;
- to avoid doubles, the personal software of health professionals will have to be able to interoperate with the personal health record;
- as a shared health record centered around the patient, the personal health record will undergo coherence reviews to ensure the quality of the recorded data;
- end control and use assistance functionalities will be found within the professional software;
- the personal health record must evolve easily in order to accept branches specific to the needs of the healthcare networks using it, and the structure of its data will also have to be standardized.

Moreover, this approach is also implemented within the project “ReSOP Interoperability”, aimed at creating a healthcare network for pediatric oncology. “The goal is to improve the continuity and quality of the care through the sharing of a therapeutic project and a coordinated coverage of both the children and his family, on the medical, paramedical, psychological and social levels” ([ENO 08], section “The ENOSIS approach”). “The project “ReSOP Interoperability” consists of implementing the necessary technical infrastructure for the digital transfer of the hospital-town and town-hospital liaison cards within the ReSOP healthcare network” ([ENO 08], section “the ENOSIS approach”).

7.6. Difficulties encountered

Implementing information technologies is not easy when a service must be provided from end to end, to manage and store patient-related data and keep them for the duration of the patient’s life.

The first difficulty lies in the specific characteristics of medical activities. Coordinating the activities of all the medical and paramedical agents, and relying on the coordination of their information systems, considering the characteristics of these activities which we have previously identified, can only be managed through the implementation of solutions already used in the field of process automation. It is crucial to take into account the character, both eminently dynamic and contingent, of these activities, something which is not currently possible with the concepts, methods and tools of process modeling. Consequently, studies must be led, on the one hand, to develop concepts and methods able to take these characteristics into account, and on the other hand, to develop tools adapted to these new concepts and these new methods.

The second difficulty lies in the transition between paper and digital files. For each patient, the doctors and health facilities already have a set of data within paper files. Several solutions exist. The first would consist of digitalizing the paper data in order to create an image of these pages. The results of this kind of digitizing are however hard to manipulate, classify and research. The solution which would consist to digitalize such documents using optical character recognition technology is not more conclusive. Most documents are manuscript and the character recognition performances are limited. Still, this technology would enable plain text searching in digitalized documents, hence facilitating their classification and helping easy research of pertinent documents. Managing documents both in paper and digital form is not more appealing. First, the coherence must be upheld between the two mediums, including the patient’s unique identification, and one must know beforehand on which medium to look for the required data. The search time must

take into account the fact that the research procedures depend on the medium and require various cognitive skills.

The third difficulty lies in the necessity of keeping the digital data available and readable during the patient's entire life, and beyond. Indeed, in the event of organ transplants, Knowledge of the donor's clinical history is essential in order to identify a possible contamination. The recent trial concerning the growth hormone demonstrated the risks of contamination, in that particular case contamination by a prion, and the necessity to keep a record of medical procedures. The physical medium (floppy disks of various formats, USB key, etc.) evolves, along with binary coding, the operating system on which the data are stored, the files' formats, the application which will read those files, the meaning of the data stored in the files, etc. As soon as major evolutions happen, the old data must be transferred onto the new mediums, and their completeness, integrity, privacy or confidentiality must be guaranteed. This requires organization, planning, budget, the implementation of specific resources, and the transfer to be monitored, to make sure there is no loss or corruption of data, etc.

Finally, a tool must be designed to store the data not correctly transferred. Since the probability that all data will be synchronously transferred is low, the new mediums must be compatible and able to interoperate with the old ones. But should all this information be kept during the person's entire life, or should one only keep a certain number of pertinent, meaningful data? If all data must be stored, the volume will be very large, which pleads the cause of limited storage of pertinent data. But determining the importance of such or such information is just as complicated. Data sometimes only becomes meaningful much later on, with new information, new means of investigation.

The fourth difficulty lies in the confidentiality of the personal medical data, as well as the patient's access to the featured information. These requirements on the confidentiality of and access rights to the medical data are framed by a set of legal texts. Their implementation relies on the standards and regulations on the security of information systems, introduced in Chapter 12, "Standardization in the field of systems and systems of systems engineering". Beyond those legal and technical dimensions, the questions about confidentiality and access rights have a particularly important societal dimension and, therefore, are treated in information documents, such as the thematic file "Patients' rights: access to the medical record" (*"Droits des malades: accès au dossier médical"*) and the document "the expectations of patients and people covered by health insurance about health information systems" (*"les attentes des patients et des assurés en matière de systèmes d'information de santé"*), published by the CISS⁸ [CIS 07]. This aspect also has a non-negligible financial

8. Interassociative collective on health (<http://www.leciss.org/>).

dimension, as attested by Google and Microsoft's offers about a service for the storage of medical data⁹.

In France, the GIP DMP¹⁰ is in charge of the patient file project. Lastly, providing all the medical and paramedical agents with secure systems of medical data entry, reading and communication, may have a non-negligible cost.

7.7. Conclusion

In the medical field, capability challenges led to the design of systems of systems, healthcare networks, including health facilities, private practices, but also paramedical and social agents. Providing end-to-end services, focused on the patient, requires those agents' activities to be coordinated, and patient-related data to be exchanged, with security requirements typical of personal data. The diversity of the information systems implemented to support these activities requires them to interoperate in order to achieve such coordination, an interoperability which must be largely built on the standards of the medical field.

In such a context, the main difficulties lie in the expression and modeling of the medical activities, which are always the "here and now" product of the interaction between health personnel and patients, in the transition between paper and digital mediums, the necessity of keeping these data available during the patient's entire life, and finally, in the control of the personal data's confidentiality.

7.8. Acknowledgments

I warmly and most particularly would like to thank (in alphabetical order) Karima Bourquard, Yves Constantinidis, Claude Pourcel, Jean-Claude Sarron, Michel Veret, who were large contributors to this chapter.

I would also like to thank the many people who helped me with this document and the many people whom I was able to converse with, and whom I could not give an exhaustive list of, but who will not fail to recognize themselves.

9. Advertisement "Google présente un service de stockage des données médicales", published February 29th, 2008 (<http://fr.news.yahoo.com/rtrs/20080229/ttc-google-sante-fe50bdd.html>) and "Les dossiers médicaux personnels, nouvel enjeu pour Google et Microsoft" (<http://fr.news.yahoo.com/afp/20080302/thl-usa-internet-sante-96993ab.html>).

10. GIP: Grouping of Public Interest; DMP: Personal Medical File.

7.9. Bibliography

- [ABE 05] ABE T., "What is Services Science?", *Research Report*, n° 246, The Fujitsu Research Institute, 2005.
- [AMA 09] AMALBERTI R., "Violations et migrations ordinaires dans les interactions avec les systèmes automatisés", *Journal Européen des Systèmes Automatisés*, vol 43/6, pp.647-660, 2009.
- [BRI 05] BRIQUET M., COLIN J., GOURC D., POURCEL C., "Organisation d'un système hospitalier par pôle d'activité: modélisation et identification des problèmes de pilotage", *Actes du 6^e Congrès International de Génie Industriel*, Besançon, June 2005.
- [BRO 99] BROWNING T., "Designing System Development Projects for Organizational Integration", *Systems Engineering*, vol. 2, n° 2, 1999.
- [CAP 07] COLLEGE OF AMERICAN PATHOLOGISTS, *SNOMED Clinical Terms User Guide*, 2007, available online: <http://www.ihtsdo.org/our-standards/technical-documents>.
- [CIS 06] CISS, "Les attentes des patients et des assurés en matière de systèmes d'information de santé", 2006, available online: <http://www.leciss.org/>.
- [CIS 07] CISS, "Droits des malades: accès au dossier médical", thematic sheet of the CISS n°11, available online: <http://www.leciss.org/>, 2007.
- [DAG 04] DAGNINO G.B., "Complex systems as key drivers for emergence of a resource-and-capability-based interorganizational network", *Emergence: Complexity & Organization*, vol. 6, n° 1-2, 2004, available online: http://iscepublishing.com/ECO/about_eco.aspx.
- [EHC 03] EHCR, "Introduction à ENV 13606".
- [ENO 08] ENOSIS, information taken from the site (now closed) <http://enosis.phpnet.org/>, 2008.
- [FID 06] FIDOCK J., "Organisational Structure and Information Technology (IT): Exploring the Implications of IT for Future Military Structures", *DSTO*, Edinburgh, 2006.
- [GIU 01] GIUSIANO B., "La communication dans les réseaux de soins", *XI^e Journée de Dermatologie Pédiatrique*, Arles, 2001.
- [GMS 06a] GMSIH, "Architecture et urbanisation des SIPS; Exemple d'architecture métier - Commentaires", 2006, available online: <http://www.gmsih.fr/>.
- [GMS 06b] GMSIH, "Architecture et urbanisation des SIPS; Exemple d'architecture fonctionnelle cible", 2006, available online: <http://www.gmsih.fr/>.
- [GMS 07a] GMSIH, "Etude de définition relative au système d'information pour la coordination des soins entre les réseaux de santé, la médecine de ville, le secteur médico-social et les établissements de santé", 2007, available online: <http://www.gmsih.fr/>.
- [GMS 07b] GMSIH, "Rapport de la première campagne d'évaluation BREC; transversalité et schéma directeur du système d'information", 2007, available online: <http://www.gmsih.fr/>.

- [GMS 07c] GMSIH, “Enquête sur la situation des Systèmes d’Information Hospitaliers dans les établissements de santé”, 2006, available online: <http://www.gmsih.fr/>.
- [GOF 79] GOFFMAN E., *Asiles*, Les Editions de Minuit, Paris, 1979.
- [HAR 02] HARADEN C. & AMALBERTI R., *Course: What is New in Safety Thinking and its Implications for Health Care*, IHI forum, London, UK, 2002.
- [HID 06] HIDAKA K., “Trends in Services Sciences in Japan and Abroad”, *Science and Technology Trends, Quarterly Review*, n° 19, 2006.
- [LER 09] LEROY N., CHAZARD E., BEUSCART R., BEUSCART-ZÉPHIR M.C., “Toward automatic detection and prevention of adverse drug events”, *Stud. Health Technol. Inform.*, vol. 143, p. 30-35, 2009.
- [MIN 82] MINTZBERG H., *The Structuring of Organizations*, Prentice Hall, 1979.
- [PEL 09a] PELAYO S., ANCEAUX F., ROGALSKI J., BEUSCART-ZÉPHIR M.C., Organizational vs. technical variables: impact on the collective aspects of healthcare work situations, *Stud. Health Technol. Inform.*, vol. 150, p. 307-11, 2009.
- [PEL 09b] PELAYO S., BERNONVILLE S., KOLSKI C., BEUSCART-ZÉPHIR M.C., “Applying a human factors engineering approach to healthcare IT applications: example of a medication CPOE project”, *Stud. Health. Technol Inform.*, vol. 143, p. 334-339, 2009.
- [PER 84] PERROW C., *Normal Accidents*, Doubledays, New York, 1984.
- [POU 06] POURCEL C., CLÉMENTZ C., “Approche modulaire pour l’aide à la conception de l’organisation d’un système de production de soins”, *Actes de la conférence Gestion et Ingénierie des Systèmes Hospitaliers (GISEH)*, Luxembourg, September 2006.
- [POU 08] POURCEL C., CLÉMENTZ C., BISTORIN O., “Système productif de soins de secours, d’urgence et 2HM; approche systèmes de systèmes”, documents personnels, 2008.
- [RAK 05] RAKOTONDRAINAIVO A., GRANDHAYE J.P., POURCEL C., “Identifications des décisions dans un réseau de santé”, *Actes du 6^e Congrès International de Génie Industriel*, 2005.
- [ROJ 03] ROJOT J., *Théories des organisations*, Editions Eska, Paris, 2003.
- [SPO 06] SPOHRER J., MAGLIO P., *The Emergence of Service Sciences: Toward Systematic Service Innovations to Accelerate Co-creation of Value*, IBM Almaden Research Center, San Jose, 2006.
- [THO 04] THOMPSON J., *Organizations in Action*, Transaction Publishers, New Brunswick, 2004, first edition 1967.

Chapter 8

Critical Infrastructure Protection

8.1. General context of critical infrastructure protection

8.1.1. *Challenges*

One of the main characteristics of our modern societies is their megalopolises, whose numerous infrastructures are often the hubs of the economic and social activity of the region, sometimes of the country. A non-exhaustive list of key infrastructures of the economic activity would feature: airports, train stations, subway stations, the main commercial harbors, regional centers of food supplies (for example Rungis in the Parisian suburbs).

All these infrastructures feature the following characteristics:

- they are thoroughfares through which thousands, sometimes tens of thousands, people and/or goods transit every day;
- they are open and interlinked within generally dense agglomerations;
- closing them for several days on end would have severe consequences on the human, social and economical activity of the region, sometimes of the country;
- any incident or deliberate attack leading to a severe malfunctioning of the infrastructures would instantly attract the attention of the media, or even of the politics.

Similar issues are raised by large events which bring together a high number of players or spectators: political meetings, sports games such as soccer, rugby or car races, etc.

For most of these events, security demands that the problem be studied in its entirety, the whole infrastructure taken into account, in its physical definition (the buildings and various physical components) as much as its functional definition and its procedures. Indeed, if the physical attack of an infrastructure might obviously cause a sensible alteration of, or even put a stop to, its operation, a functional attack without any physical damage might just as well lead to a shutdown.

8.1.2. *Structure of a vital infrastructure*

An infrastructure is built on three types of components: physical, functional and organizational. To illustrate these concepts as clearly as possible, we will use the station of a metropolitan network, but the ideas are generic and can just as well apply to any other type of critical infrastructure, such as stadiums, airports, regional food supply centers, etc.

The *physical components* are of course the most visible, and the first components to come to mind: they are the various buildings and works, or part of the buildings and works, which form the infrastructure. For example, the physical components of a metropolitan train station include the buildings, the corridors, the hallways, the platforms, the bulletin boards (the public ones as well as the ones used by the operators), the railway, and last but not least, the rooms and devices necessary to the train's control and power supply (transformers, control room, etc.). Evidently, any significant damage suffered by these components could, depending on its criticality, bring the activity to a momentaneous, partial or complete stop. Such would be the case, for example, in the event of an explosion, a fire, or a flood, resulting in partial destruction.

Since the main objective is to protect the infrastructure against deliberate threats, its physical components are compartmentalized, broken down following a notion of "basic physical component", characterized in relation to the people that will go through them:

- an "open" basic physical component of the infrastructure receives a flow of persons that have not been subjected to a background check, and is characterized by a large geographical perimeter, featuring unbroken segments, sometimes poorly

defended against intrusions. Such is the case, for example, with airport runways, train stations, security perimeters within the Vigipirate¹ system, school campuses;

- a “semi-open” basic physical component receives a flow of people that have not been subjected to a background check, and is characterized by a geographical perimeter through which intrusions are only possible on a small number of well-identified spots. Such is the case, for example, of an underground subway station, a public area of ministries, a school;

- a basic physical component with “controlled access” is a “semi-open” basic component which only accepts people that have been subjected to a background check. Such is the case, for example, of ministries, EDF (France’s main electricity provider company) power plants, sensitive organizations;

- a basic functional and physical component of the type “itinerary from point A to point B” is a part of the infrastructure through which people or goods transit to go from point A to point B, following a well-known itinerary. Such is the case, for example, of the highway or railway networks, a luggage screening system, waterways, power transmission channels, etc.

As we can see, these basic physical components can be easily assembled to form more complex physical structures. For example, the physical structure of an airport features the following basic physical components:

- open (runways);
- semi-open (passenger terminal);
- with controlled access (from the security screening to the plane’s door);
- itinerary (passengers’ trip from the security screening to the plane and likewise, but following a different route, for the luggage).

The *functional components* encompass all the services which the infrastructure must provide in order to fulfill its mission. For example, for the aforementioned train station, the services would be:

- bill the passengers;
- carry the passengers;
- keep the passengers informed;
- ensure the passengers’ and staff’s safety;
- etc.

1. Vigipirate: France’s national security alert system.

For example, let us pretend that the subway's barriers are locked into the open position. The physical components as previously described would not be destroyed, but a main functionality would be affected and it would in all probability lead to the momentary shutdown of the station till the barriers were fixed, or to their temporary replacement by human ticket collectors.

These main or first tier functionalities can be broken down into several functionalities of lower or secondary levels. For example, "carry the passengers" needs two functionalities in order to ensure:

- that the passengers can access the trains;
- that the traffic is regular.

Thus functionalities can be broken down into a succession of functionalities of inferior levels whose fulfillment is necessary for the global working of the infrastructure. For instance, the propagation of tear gas through the station's passageways would affect the functionality "ensure access to the trains" and would therefore paralyze the whole station, without altering the physical components in any way.

It should of course be noted that the partial or total destruction of physical components generally leads to the loss or the sensible degradation of one or more functionalities, while the reverse does not necessarily apply.

Lastly, the infrastructure features *organizational components*. The smooth running of the infrastructure generally calls for human intervention, either for its actual operation, or for maintenance and repairs, or to ensure the passengers' safety and create a feeling of trust, etc. Stopping, paralyzing or altering these men's work inevitably has immediate consequences on the functionalities, which can themselves have an impact on the physical components. An illustration is easily found in the problems arising from personnel strikes: traffic is disrupted or stopped, and therefore the functionality "carry the passengers" is altered or even completely unavailable. But one might also envision the problems arising, for example, from poor organization of the communication between the various departments, which would also lead to the disruption or actual stop of an infrastructure's operation.

Succinctly, to secure a vital infrastructure is to secure the set of its components against various hazards and threats, whether fortuitous or intentional. "Perfect" or total security will never be reached in actual practice, would it be only because of its cost. The most likely and/or most dangerous hazards and threats will systematically be assessed along with the protection focus on them.

8.1.3. *Hazards and threats*

Incidents altering the infrastructure in one or several of its components can be caused by human or material failures which are, in the case of material failures, fortuitous. However, they might also result from deliberate attacks, such as vandalism, sabotage or terrorism: so-called established threats.

To efficiently secure an infrastructure, an inventory as thorough as possible is therefore necessary, both of the hazards that might befall each of the aforementioned components, and of the most credible threats. Their likelihood and severity must also be assessed.

Most risks generally taken into account in critical infrastructures are those linked to fires and electrical and mechanical failures.

Simple hazards can lead to events with no great impact on the infrastructure's physical components, such as an electrical failure on the level of an airport's bulletin board. But they can just as well have a serious impact on the infrastructure's functional components.

Obviously, some of these hazards hold more danger for the physical components, and therefore might have consequences that will block the functional components. Moreover, these hazards can create incidents with varying consequences depending on the places and circumstances. For example, a simple fire can, if it reaches a transformer containing pyranol, turn into a chemical incident with the spreading of highly toxic smokes. This is a "domino effect", where one incident sets off a chain of other incidents, each on a larger scale than the previous one.

Likewise, threats must be described and evaluated according to the infrastructure, the eventual social tensions within the organization managing the infrastructure, the town or the country, as well as the current geopolitical situation. Nowadays, they are stigmatized by the terrorist threat, the importance of which was unfortunately attested by the catastrophes of September 11, 2001, the Madrid train bombings, the London bombings, etc.

These threats are all the more difficult to prevent since they exploit suicidal behaviors and aim at domino effects, in which the catastrophes are mainly the result of the triggering events' consequences. The September 11 attacks are a dramatic example of the desired domino effect. In that particular case, the terrorists did not board with any kind of explosive, but used the very planes as bombs, planes which then provoked massive fires, fires which made the twin towers collapse, collapse which caused many more victims than the crashing of the planes.

Finally, it should be noted that individuals with nothing to lose can use fragmented security systems as facilitators to their attacks! Indeed, in many cases, the possibility of a domino effect is not really taken into account by the system, even in relatively simple situations. For example, many critical infrastructures include restricted areas for which no solution of global security has been planned in case of fire or alarm. The fire alarm system triggers the opening of doors to allow for the prompt evacuation of personnel, and there is no system left to monitor the entries! It is therefore easy to start a simple fire in a nearby area and thus open the doors of the area one wishes to enter.

8.2. Protection requirements

The infrastructures' complexity, which stems as much from the number of their components as from mutual dependence, and the few examples we have studied which emphasize the necessity of taking the value chain into account under its physical, functional and organizational aspects, illustrate the "system of systems" dimension which must inevitably be considered if one wants to achieve true, flawless security. Protection must therefore only be conceived in its entirety, by meeting a set of requirements.

8.2.1. *Looking at the infrastructure in its entirety*

The protection of a critical infrastructure must take into account every one of its components, each belonging to one of the three categories we have mentioned. It must be conceived with the obvious purpose of avoiding major crisis situations, or even catastrophes. However, by the very essence of the aforementioned hazards and threats, there can be no absolute guarantee that such a crisis will never happen. The security system will therefore have to be of use not only before, but during the crisis, and help curb it as much as possible.

To that end, the security system must be operational:

- preemptively, to avoid a serious crisis through the instant detection of its warning signs and the activation of the required actions;
- at the outset of a crisis, to activate (or allow the activation of) the interventions of the appropriate level;
- during a crisis, to monitor its evolution if it hasn't been avoided, and assist the interventions (for example by guiding the personnel);
- at the end of a crisis, to restore a normal situation, safe and secure.

Therefore, the study and the design of a global security system must regroup, around the team of engineers in charge of that design, all the agents involved in the infrastructure on any level, namely the infrastructure's operators, the incident response teams, and the regulatory authorities.

The added value of the infrastructure's operators (or even of its end-users) lies, on the one hand, in their inventory of the existing data, tools and procedures, of the implemented organizations, and on the other hand, their inference, from that inventory, of the needs in: infrastructure modeling, available information, suitable representations, control and exchange of data with the incident response teams and the authorities.

The input of the incident response teams is essential to narrow down the demands regarding the informed mapping of the premises and infrastructures, and the teams' means of transmission and local control.

Finally, the input of regulatory authorities is necessary to narrow down the needs in rooms dedicated to the control and synthesis of the transferred data, the decision support systems, the support systems for dynamic management of the actions, the procedures and formats for the interoperability with the systems of military intervention in the event of specific major crisis.

It is indeed necessary for the system to take many aspects into account, such as:

- the high level security policies which govern the running of the town or the region in the event of a serious crisis;
- the normal, or even the corrupted, operating conditions, and the functionalities of the infrastructure which are well-known to the operators;
- the conditions in which the appropriate teams can operate, such as the time necessary to reach the scene, the access roads, the means of intervention and the damages these might cause on the security system itself (for example, using fire hoses probably implies shutting down the nearby electric circuits);
- the interfaces corresponding to the intervention services (organization, communication tools, messages formats, etc.).

The various agents must therefore participate to the studies on the definition of the global security system, to the design of scenarios that will test the systems' efficiency (during the elaboration, the testing, or the drill of the incident response teams), to the actual tests and drills, and to the establishment of conclusions.

Moreover, besides the conceptual difficulty of building a global alert and management system both adapted to the infrastructure and taking into account the

specificities of the places, the organizations and the available means, the actual achievement of such a global system is challenged by the multiplicity of the agents involved, of the procedures and various databases, and finally by the multiplicity and asynchronism of the budgets for equipments and operation.

One of the largest margins for improvement in global security lies in the coordination of that multiplicity. On the technical level, this coordination happens through the choice of tools and local procedures which are coherent from one service to the next, part of a global master plan.

The most efficient procedure is therefore to first sketch out the global system we are wishing to implement in the medium term, so as to make an informed decision on the equipments the agents should be supplied with, and thus help those agents make investments, settled on locally but part of the master plan and therefore contributing to the system's global improvement.

8.2.2. *A structured, continuous approach*

The infrastructures evolve with time, following the economical and/or demographic development of the town or of the region. An example can be found in the continuous evolution of airports, driven by the democratization of air transportation, the economical evolution and the technical advances of civil aviation. This leads to regular modifications of the functional, organizational and physical structure of the infrastructures, sometimes even to the construction of new physical structures (e.g. the arrival of new large commercial airplanes, such as the Airbus A380, entails new terminals, landing tracks, etc.).

The infrastructure's protection must inevitably evolve along with the infrastructure itself, while keeping its global dimension.

This implies an approach structured through time, consisting of:

- defining a coherent and progressive (on the scale of a few years) plan of development and evolution, so that it remains coherent with the investment plans and the evolution of needs;
- implementing an organization charged with the constant monitoring of the global security system's adaptation;
- defining corresponding budget lines, so the necessary adjustments can be made in time;
- choosing electronic and organizational systems that are by nature evolutionary, therefore minimizing the loss of previous investments. This calls for the use of

standards whose durability is probable, and the hiring of industrialists who can vouch for that durability.

These steps are still underestimated in many cases, most often due to economical constraints. For that reason, in time, the systems often become incoherent, sometimes even unsustainable, and feature “weak links”, doors wide open to hazards and threats. In some circumstances, the old savings can have expensive consequences, etc.

8.2.3. Confidentiality

The efficient protection of an infrastructure calls for the upholding of confidentiality on various levels.

First of all, the design of a security system demands that the hazards and threats be analyzed, and more specifically the ones relative to the weaknesses of systems already in use. The weaknesses of the physical, functional and organizational components must be identified and passed on to the industrial teams charged with the design of a new security system. It is clear that such information is sensible and must only be transferred under certain guarantees of confidentiality so as to avoid handing possible targets to terrorists. To this day, national legislation, in France notably, does not regulate the confidentiality of this type of information insofar as its disclosure does not threaten national security (when such is the case, classification levels such as “confidential”, “secret” and “top secret” are specified in the penal code). On the European level, this problem is slowly taken into account, notably through the possibility, introduced in 2007, of classifying certain results, or even some research programs.

Moreover, no matter how efficient, every detection system has limitations which can be exploited to circumvent it. For example, knowledge of the sensibility limit of a scanner, meant to detect objects hidden under clothes according to their volume, could lead a terrorist to carry explosives in smaller pieces.

Thus the need for confidentiality when it comes to the characteristics and limitations of equipments and detection and analysis systems which play a part in the protection of a critical infrastructure.

But this goes against the commercial promotion of the equipments sold by manufacturers, who generally mention in no unclear terms their products' limitations. Thus the importance of ensuring the confidentiality of the principles, technologies and equipments implemented within an infrastructure's global security system.

8.2.4. *Security analysis file*

The protection of a critical infrastructure calls for the establishment of a general security file which will breach every aspect, from the detailed analysis of the infrastructure, the identification of hazards and threats, to the potentially usable technologies.

Typically, a security file should include at least the following facts:

- a functional description of the infrastructure in terms of principal functions and those functions' constitutive sub-functions;
- a description of the requirements that are to be checked by these functions and sub-functions;
- the analysis of possible hazards or threats, through the identification of the affected infrastructure's component(s) (physical, functional or organizational);
- the analysis of the consequences those hazards and threats might have on the functions, in terms of gravity (through the definition of several gravity levels), likelihood (to be defined for each level of gravity), and risks (function to be defined from the gravity and likelihood, their product for example);
- a detailed description of the most likely hazards and threats against which security must be reinforced: description of the physical and organizational components enabling the implementation of the function or the sub-function (place, environment, existing security system, etc.), consequences of the hazards or threats on the physical, functional and organizational components;
- possible solutions to diminish the risk, prevent an established risk, raise the alarm as soon as a sub-function is altered;
- the technical means involved in those possible solutions: which kind of technique and technology, the feasibility and level of control, the availability of the matching equipments, the durability of the solution, in terms of the product's evolution as much as the durability of the industrial base and production industry.

8.2.5. *Decision making*

The protection of a critical infrastructure can answer constraint fields of rather diverse natures. It can be supported by:

- legal or institutional demands, such as international or regional regulations: for example, the standards laid down by the *International Civil Aviation Organisation* (ICAO) or by the regulatory authorities of the infrastructure's operator;

- the necessity to ensure the durability of the users' trust. For example, the extended unavailability of an infrastructure such as a subway line, following an incident, can lead to the development of alternative solutions by the users, and a substantial loss of customers in the long run, even once the infrastructure is back to normal;

- the prospect of a short- or medium-term return on investment. In that case, protection is a means and not an end, not necessarily part of a long-term plan.

It is easily conceived that, depending on the case, the demands will differ and so will the initiators of the protection operation. Notably, in the last case, the industry is the driving force and must not only offer its potential clients a security system, but also a profitable economical model.

8.2.6. Admissibility

The legal admissibility of a system is a constraint that must be taken into account, depending on which country the system will be implemented in. Some technologies are allowed in some countries while they are banned in others. As an example, in the United States, the use of x-ray screening systems is authorized for the control of people, this is forbidden in France, where standard controls must generally not include any intrusion upon the human body.

The public's acceptance of the security system is also an element that should be pondered on before reaching a decision, even more so since critical infrastructures often see a very great number of people through every day, people of highly varied sensibilities and cultures. The infrastructure's nature also has a direct influence on that acceptance. Let us look at two radically different cases: a regional or national public administration building, such as a French prefecture, and an airport.

In a public administration building, people must be welcomed and, as much as possible, should not feel that they are being observed, their behavior analyzed, a database filled without their consent. This feeling (the *Big Brother* syndrome) would be totally groundless, since such practices are illegal in most countries, including France. The purpose of public administration is, after all, to deliver a service to the people, and not to suspect them *a priori* or control them without their knowledge. A security system must therefore be discreet so as not to attract attention, to curb any reluctance and exaggerated protest movement.

On the other hand, in an airport, the pre-boarding screening, even though very visible and troublesome, is well-accepted, for each passenger understands the benefits to their own safety.

Moreover, depending on the events and the country's geopolitical situation, the public's acceptance of more visible, sometimes more exacting, control systems, is much more easily achieved. For example, in the weeks that followed September 11, 2001, screenings in airports were much more thorough and led to rather frequent searches, since the sensitivity of the walk-through scanners had been considerably heightened.

This shows how much acceptance depends upon the context of the security system's implementation, and the country's (and the time's) geopolitical conditions.

8.3. Security systems of the future

Many of the security systems currently in use are based on a principle of deterrence, which is to say on the fear of being found and prosecuted. Such is the case with video surveillance systems. Admittedly, this approach is successful within the current Western ethics, in which freedom and life are primordial values. It should be noted that these values are dependent on the current geopolitical conditions, and deep individual feelings. The value system of one society is not necessarily shared by other societies. For example, some terrorist organizations do not place the same value on human life at all.

We can also note that stress conditions and social troubles can lead to desperate action where life is no longer at the center of the value system. In all these situations, suicidal behaviors emerge, which completely invalidate the dissuasive approach, since the eventuality of an investigation and a sanction loses all meaning.

Finally, the ever-growing complexity of large infrastructures made up of several functional, organizational, sometimes even physical systems, leads to the study of the aforementioned domino effects, and the search for the early containment of any event that could trigger such an effect.

All these reasons lead us to consider, more and more often, proactive security systems, which would not only detect serious events, but also the warning signs of such events, and trigger the appropriate actions to keep said events from happening.

8.3.1. *Proactivity, crisis management and resilience*

The proactivity of a security system can only stem from the intelligent analysis of a set of weak signals, received elements of information which are not, in themselves, important enough to trigger an alert insofar as they only correspond to the minimum probability of a dreaded event.

On principle, the aim is to integrate several weak signals into a proactive system, in order to create a range of corroborative hypotheses whose probability reaches a sufficient level so as to correspond to the near occurrence of a dreaded event. This process is akin to the human reasoning which makes a sensible individual expect a fire when he sees a burning cigarette butt thrown to the ground next to an inflammable liquid spreading on the surface; thus the need for devices that can integrate these weak signals which are based, as suggested by the previous example, on the disjointed occurrence of elements constitutive of a dreaded event. Of course, solutions are also sought to eliminate the constitutive events themselves, which is always for the best!

As successful as proactive systems may be, it would be irresponsible not to envision that, despite those systems, a dreaded event might happen and lead, for example through a domino effect, to a serious crisis. To that effect, we should always try and anticipate possible events so as to possess, when the time comes, the appropriate means to counter them.

To this end, we need to be able to follow the evolution of the crisis, to anticipate the possible effects if the crisis is not stopped, to evaluate the means necessary to stop it and to trigger their implementation.

Thus the necessity of crisis management tools, which help monitor the crisis, anticipate the effects and evaluate which means to implement.

8.3.2. *Early reduction of risk*

The design of the security system must take into account the desired resilience following a crisis, that is to say the infrastructure's ability to repair itself so it can resume its functionalities as soon as possible, safety being one of the main functionalities to restore.

To proactively protect, we must therefore either eliminate the elements constitutive of a dreaded event or crisis, or detect them through an intelligent integration such as we studied in the previous section.

Considering the infrastructure's "system of systems" dimension, we should first try to reduce the risks linked to the apparition of an element constitutive of a dreaded event, and do so in all the functional, organizational and physical fields. To this end, the risks must be analyzed and eliminated as soon as possible, within the infrastructure's standard operation.

For example, on an organizational level, the first thing would be to check the “reliability” (trust and morals) of any personnel with access to the infrastructure’s sensitive zones, without omitting the service providers. On a functional level, replacements must be ensured in the event of the failure of some of the systems. On a physical level, for example, we should plan redundant communication tools to keep in contact with the teams or the sensors in the event of a failure of the regular tools, provide safe havens for people trapped in a fire, etc.

8.3.3. *Electronic detection systems*

The critical infrastructure security market has known an exponential growth in the last few years, in particular since the events of September 11 and the bombings in Madrid and London. The realization of some infrastructures’ fragility has directed worldwide research, leading to the emergence of new technologies. These new technologies have in-depth impacts on security measures, and they come into play on various levels within the 21st century’s security solutions by giving them a new system dimension, laying more importance on the entire infrastructure. This leads to an interest in technologies which can operate on the system’s various levels.

The security systems which have already been implemented or are being implemented are often based on video cameras, access control systems, intrusion, biometric, or x-ray sensors. For the most part, these systems use fixed or hertzian communication standards, basing themselves on predefined architectures. They are good candidates as the foundations of future surveillance and early alert systems, as long as complements are provided by various levels of intelligence (scattered and central) and by complementary sensors.

The first level is related to the addition of new sensors, or the miniaturization and industrialization of existing technologies. These new sensors, for example neutronic detection, often efficiently replace or assist humans in brand new fields. For example, in the detection of illegal goods (explosives, drugs, etc.) trafficking, the x-rays are at the base of most equipment in use in sensitive areas: luggage screening in airports, entry control, container control, etc. The objects’ shape and density are essentially the data used in the detection process. The rate of false alarm (explosives are detected in the place of harmless objects) is then rather high. The running of these systems could be improved by adding data concerning the physical composition of the object. Such is the potential and essential service that could be added by neutronic technology, for example. Another example can be found in the sensor technologies within millimetric frequency bands which allow for the passive detection of metallic or dielectric objects, carried under a person’s clothing.

Parallel to the use of new basic technologies, heightening the elementary sensors' capacities, either through intelligent networking, or through their coupling with information technologies and in particular image processing, equips a sensor or a set of clustered sensors with a local intelligence capacity. For example, scene analysis software can detect abnormal behavior, abandoned objects, etc. There lies the second major technological level.

The third technological level is related to the abundance of sensors and information, whether rough or elaborated. This overwhelming amount of data has put forward the need to correlate or merge said data, so as to avoid a single event being reported as many times as there are sensors (following the principle that too much information kills the information). Moreover, this fusion, or intelligent merging of data, leads to a much richer characterization of events, which enables the estimation of risks and even the forecasting of domino effects. For example, the simple coupling of audio-surveillance and video-surveillance leads to the correlation of events at the base of the data individually picked up by each sensor.

Lastly, for their work to be made easier, the operators need to be given the clearest and most informed vision of the situation, so they can then make the right decisions. This will only be made possible through the use of information management tools and man-machine interfaces capable of giving them the constantly updated tactical situation while drawing their attention to the most critical parameters. This presentation aspect is crucial, for closely related to the human being and his limitations. This is the fourth technological level.

The use of these technological contributions, on each level – sensors, sensor networks, data fusion, information presentation – is greatly facilitated by open software architectures of the *middleware* kind, which enable the integration of these new technologies within a coherent and interoperable information system.

8.3.3.1. *Architecture: the sensors*

Let us now detail the architecture of the system of systems formed by the critical infrastructure's security system, and in particular the information related to the sensors. This architecture ought to be as modular as possible: modularity is reached through the defining of autonomous blocks interfaced with one another, so as to control the complexity which arises as much from the number of constitutive elements as from their interconnection topology. This property partly puts the architectural problem on the level of interface management, which on the one hand reduces the complexity, and on the other facilitates the insertion *ad libitum* (*plug and play*) of new components.

Modular architecture can translate into the organization of the sensors into three levels:

- the lowest level, the level of basic sensors, such as a video camera, a microphone, or a gas detector. On that basic level, intelligence can consist, for a video camera, of pre-processing the information to determine outlines, speed, shapes, colors, so as to only transmit data relevant to the current situation. Some video cameras are also capable of changing their focus depending on certain conditions, conditions which are checked out by those very cameras;
- the intermediate level, a “cluster” of sensors of similar or differing natures, for example a set of video cameras fully or nearly collocated, observing the same scenes. In large and complex sites, the great number and heterogeneity of the sensors creates a flow of data difficult to process with a centralized architecture. The challenge is to decentralize part of the intelligence within clusters of sensors, leaning on:
 - the collaborative integration of several physical principles (audio-video, magnetic-millimetric, X-neutron, etc.),
 - local preprocessing capacities,
 - connections to data processing and communication nodes, distributed within the sensors’ network;
- the highest level, where all the data transmitted by the previous levels is integrated. This integration is done in a security control center, where the high level data, which trigger the alerts and monitor the possible evolution of events, are elaborated.

The collected data is all the richer since it comes from several kinds of sensors. The interpretation of an image transmitted by a video camera is greatly facilitated by simultaneous acoustic information: a sudden sway in the crowd and the simultaneous sound of an explosion are much more instructive than each piece of information received separately. However, the intelligent fusion of heterogeneous data is not simple and demands that the information coming from different kinds of sensors be communicated in a unified format. This integration must guarantee a level of data integrity: indeed, the fusion must compact the data into a more elaborate message, but must not under any circumstance fabricate information; moreover, it must *not* use all the available data. Intelligence, which consists of judiciously integrating the information picked up by the sensors, happens on each of the previously mentioned levels.

A security system’s optimum architecture is the one which optimizes that intelligence depending on the level, so as to lead to the best result while reducing the

global cost of the system (not necessarily by reducing the number of sensors) and securing the system's evolutionary nature.

8.3.3.2. *Communications*

The electronic security systems of a critical infrastructure almost always consist of a constellation of multisensors. Since the sensors are spread on a relatively broad geographical area, it is necessary to transmit the collected data, sometimes already pre-processed by a sensor or a cluster of sensors, toward a sublevel processing node or straight to the central control room. Whichever the case, channels of communication must be implemented between the various levels of the architecture.

Three major solutions can be used:

- dedicated communication networks;
- internal communication networks, of general use, which belong to the infrastructure's internal communication networks even though their usage is not restricted to the security system;
- open communication networks, which can have other uses besides what is required by the infrastructure: Internet, GSM, etc.

By nature, security demands that the communication network itself be safe and able to guarantee the received data's integrity. The physical or functional continuity of the transmission channel must therefore be monitored so as to guarantee the transmission, as must be the security of the transferred data, so as to prevent it being altered in any way, whether it be by accident or by design.

The transmission channel might be material (cable, optical fiber) or immaterial (radio waves, infrared). In both cases, we should monitor network intrusions, that is to say block or delete information added within the network and not coming from one of the security system's sensors. For example, interferences in wireless transmission systems. Thus the need to first authenticate information by checking its origin, then guarantee its integrity.

For example, the convergence and interconnection of application software around an IP network introduces new threats. The goal is no longer to secure one or two external connections located on the network's fringes, but to implement a strategy over the entire network, in which the security functions are scattered among the various network layers and working components. If security must always be assured on the fringes, it must also be implemented ahead of the resources (for example on the servers), on the client workstations, all the way to the network's core. Such a strategy is all the more necessary in the case of radiocommunication networks, because of their open nature.

Lastly, as can be easily guessed, upgrading a security system through the addition of new sensors asks for the design of an extensible communication network which can simplify that addition. In that particular case, the in-depth defense principle concerns the authentication of the equipment and the users, regardless of the context, and the IP communications' confidentiality, regardless of the transfer channel.

Moreover, if the network does not in itself have a sufficiently high security level (public Wi-Fi network, civil satellite access, etc.), a sufficiently efficient protocol of end-to-end IP encryption must then enable the use of multimedia applications (voice, video) while maintaining the quality of service.

8.3.4. *Plug and play*

The *plug and play* concept, well-known from the general public, consists of making the system able to automatically autoconfigure so that any physical and functional evolution is as transparent as possible to the user.

The standards and the technologies pertaining to that concept are a good illustration of the concept's growing importance as a unifying key mechanism, as much on the level of interoperability as on the modularity of ever more dynamic and mobile installations. Today, these technologies are mainly available to the public at large and identified within the global issue of *Home Networking*. Indeed, the ever-growing presence of digital equipments both intelligent and able to communicate (PC, high-rate modems, video recorders, digital television networks, surveillance cameras, printers, etc.), their increase in number and types, clearly asks for the simplification of their deployment (the user is not systematically a seasoned computer specialist!).

It should also be pointed out that the Web and the current success of service-oriented architectures (SOA) are another important source of inspiration which goes well beyond the concerns of *Home Networking*.

Two architectural approaches are possible, not necessarily opposable but rather complementary:

- the work approach, specific, generally low-level and mostly driven by equipment manufacturers;
- the generic approach, generally high level, sometimes even very high level in the case of Web services, mostly driven by software publishers (in particular operating systems designers) and the Web world.

These approaches can be used as an axis for the development of the architectures of critical infrastructure security systems, such as airports, large train stations, energy storage sites, buildings with a high media and political value. Indeed, these infrastructures' configurations might need to be upgraded for several reasons:

- either following a spatial extension of the infrastructure: for example, the Paris Roissy 2 airport is in full expansion with the opening, in the last two years, of two more terminals;
- or when the security level must be heightened:
 - new international regulations,
 - temporary increase of risk corresponding, for example, to a higher level of threat within a security system (“Vigipirate”, in France), which calls for the deployment, for a limited time, of new equipments to complement those already in use,
 - the occurrence of a particular event in a site which is not permanently identified as a critical infrastructure (social, cultural or sports events, meetings of key political figures, etc.),
 - necessity to intervene within an infrastructure during a major crisis (bombing, accident) or in an urban area after a catastrophe.

In each case, we wish to avoid modifying, or even fully replacing, the electronic security system already in use. The issue is therefore, on the one hand, to design means to detect the current configuration so as to auto-adapt in real-time the algorithms to the sensors' heterogeneity, their number and their positioning (position and coverage), parameters *a priori* unknown. On the other hand, automatic mechanisms must be designed so as to reach, in that particular situation, optimum processing of the data transferred by the sensors and the databases.

For example, the system architecture must automatically detect the introduction, on a given zone, of new sensors carried by the people entering that zone. It must auto-adapt the algorithms of data fusion and analysis so that this new influx of information is automatically taken into account without the operators' intervention. Moreover, the algorithms which monitor and manage the system must also reconfigure in real time depending on the detected configuration.

8.3.5. Crisis management tools

Recent events, whether they had natural (storms, tsunamis, earthquakes) or human origins (equipment malfunctions, organization malfunctions, human errors, terrorist attacks) demonstrate our society's frailty against unexpected events. By

definition, these events alter the working conditions of the installations and the agents, and can therefore lead to true disasters when the security systems are not able to stop the dreaded events before they can trigger deep crisis.

In such cases, the crisis must be monitored, its effects contained by keeping the populations and the infrastructures safe, and the means necessary for the restoration of the infrastructure's normal operation must be deployed. These various procedures are regrouped under the standard name "crisis management".

This management is all the more efficient when the following key-factors are mastered:

- the capacity to rapidly assess the situation and its evolution;
- the quickness of the intervention and of the appraisal of the means necessary to contain the crisis' consequences;
- the implementation of secured communication channels, out in the field and with the distant centers (control centers, hospitals, airports, etc.);
- the authorities' and field agents' handling of the information and global knowledge about the catastrophe;
- the coordination and monitoring of the incident response teams so the rescue can be more efficient;
- the use of decision support systems.

The technological challenges that must be met to master those factors are numerous. The answers can only be found through a global and mutual approach to the security and crisis management systems by the various agents, infrastructure managers, incident response teams, local authorities and industrial teams in charge of designing the security system and the crisis management system.

Indeed, the efficiency of crisis management depends on the coherence of actions such as:

- the provision of efficient teaching aids which will help mimic the effects of a crisis during exercises, trainings and drills of the various agents of crisis management;
- the detection of any element foreshadowing such abnormal events, which comes down to giving the alert as early as possible;
- crisis management, which means triggering the appropriate actions and the real-time monitoring of these actions' development as soon as possible so as to adapt them to the situation's evolution.

The aim is therefore to design a single security system featuring coherent interoperable tools of analysis, forecasting, decision support, training and management.

8.3.5.1. *Informed mapping*

A crisis management system must help organize means of rapid intervention before the crisis, and therefore limit an event's consequences, and enable the redeployment, simultaneously and in real time, of personnel and means, both on the level of the surveillance centers and the intervention field.

This can be done by implementing a set of technologies that will enable:

- the improvement of the real-time characterization of events;
- the sharing of information between all the agents, through an adapted real-time reproduction;
- the optimization of the answer to the events, through the predictive simulation of the situation.

This calls for great efficiency and reactivity, notably in the three following fields: local gathering of information; validation of that information; centralization and visual display of that information in a stationary or mobile control center.

It is therefore necessary to possess tools that will allow, on the one hand, the real-time access to an informed map, that is to say featuring information on the various micro-infrastructure within the studied geographical zone, and on the other hand the real-time forecasting of the crisis' evolution. For both those aspects, mapping tools are necessary and enable the transfer of information to an operator in a simple and adapted form.

More precisely, informed mapping includes geographical data such as:

- cartographic/geographic information in 2D or 3D;
- specific critical and/or useful characteristics of the equipments or the micro-infrastructure depending on the crisis' nature. For example, knowledge of the presence of pyranol in a transformer allows the forecasting of a particularly serious domino effect which could happen in the event of a fire: if the fire reached the transformer, the event would change nature and take on a chemical character it did not formerly possess.

The juxtaposition of information coming from several sources brings up several problems that should not be forgotten about, notably:

- the interoperability of the various databases, spread on a vast geographical zone;
- the granting of access to said databases;
- the legal matters of responsibility that come with the possible mixing of public and private information, the decision to act and the means implemented.

3D mapping leans, on the one hand, on 3D modeling of the site holding the infrastructure, and on the other hand on the visual display of that modeling for the benefit of an operator. The 3D modeling of a site consists of recreating, from data gathered out on the field with the level of detail required by the application, a 3D representation of the elements of terrain, culture, the infrastructure and the superstructures.

In the case of sites within an *open area* (sparse or non-existent houses), the entry data are as follows: digital elevation model, planimetric representation of the occupation of the grounds and infrastructures, airborne or satellite imagery (orthoimages and multi-angle views are recommended) of the textures and flat features, 3D models, either vectors or images. Based on these entry data, editing tools, for correction and alignment, and conversion tools, for the generation of the imagery (polygons, defined surfaces, etc.), are used to generate representation databases of the studied sites, first in the standard formats, then in formats specific to the processing tools (display, for example).

In the case of sites within an *urban area*, the entry data are: orthoimage airborne imagery, stereoscopic pairs (triplets, or even quadruplets), ground or elevated imagery, photos of particular facades, buildings and architectural objects, point clouds generated by a scanner (laser), drawings and maps. Based on these data, acquired on-site from airborne or terrestrial platforms, operations of alignment, and later of reconstruction (calculating the envelopes, for example) must be run.

In the case of *confined* sites, the entry data consist of ground imagery and drawings. Based on these data, edition and modeling tools reconstitute the internal volumes, as much their geometrical aspect as their visual aspect (within the visible, or possibly in other electromagnetic tapes), and furnish them with internal objects. Semantic information on the objects' nature, physical information on the material's nature, and topological information, can be associated to this geometric and visual modeling. These databases rapidly become voluminous depending on the level of detail and the width of the area, and must be organized in an optimized way and divided so as to be exploitable in segments.

The 3D modeling of an urban site is still a research field in parts. Current developments concern the search for automation of the various acquisition functions

(automation of the acquisition, geolocation, etc.) of alignment and reconstruction. It is highly likely that these techniques will be commonly used in security systems in the future.

Creating a 3D display of a site consists of creating a dynamic image (on a frequency going from a few Hz to 60 Hz depending on the application software) of the studied site following a perspective piloted by the application. Image generation takes into account, besides the rendering of the site and its layout (the objects and equipments which populate it), the conditions of observation, including the visual sensor's characteristics and the light conditions, as well as the weather conditions (fog, rain) and dynamical aspects such as smoke or other darkening cover. In the case of urban and confined environments, 3D visual display takes into account the many sources of light which will contribute to the scene's lighting.

On top of this "dead" vision of the site, the visual display must be able to take into account specific animation effects such as the triggered events, whether they concern objects, people or vehicles. The animation of these objects is realized through the use of specific components integrated within the simulation system and communicating with the 3D visual display component so as to give it the data necessary to the representation of the various objects' state and position.

Today, 3D visual displaying is commonly used in various fields, from real-time simulation to video games, as well as design and scale modeling simulation. The future systems will treat a higher number of static objects with a higher level of detail, they will represent the dynamic effects and agents more finely and with added realism.

8.3.5.2. *Dynamic tools*

The use of a field modeling common to all the agents must also enable access to crisis management data, such as:

- visual display of specific critical and/or useful characteristics of the infrastructures depending on the crisis' nature;
- visual display of the effects and their predictive propagation, and the quantification of the risks (effects on the population, industrial and economical impacts);
- adaptative and contextual visual display adapted to each agent;
- dynamic (time-dependent) determination of the security perimeters.

Moreover, crisis management support systems must allow real-time access to:

- segmentation of the actions to launch after an incident;

- creation and updating of structured databases, providing the incident response team with the appropriate information in terms of the nature, location and evolution of the information;
- knowledge on the crisis' constituents and the means of intervention;
- management and real-time browsing of the archived documents and procedures for fast access to a richer and more specific information upon the request of a team member;
- audit and replaying for legal or training purposes;
- evolution of knowledge and training.

In particular, the cartographic tools, with the added “crisis management” dimension, must allow for the exploitation of the data gathered in real-time by the various sensors (video surveillance, alert sensors in fields such as nuclear energy, radiology, bacteriology and chemistry, telemetry, satellite imagery, aerosols, gamma cameras), as well as the monitoring of the incident (prediction of domino effects, safety perimeters, etc.). They lean on hierarchized archival tools which can be located on other sites, and on decision support algorithms.

Hierarchized archival tools enable collaborative updating from heterogenous sources, the analysis and tagging of the data for the purpose of precise indexing, the real-time consultation of the archived documents and procedures so as to access richer and more specific information with increased rapidity.

The diagnosis and decision support algorithms perform:

- an evaluation of the situation according to a risk scale relative to the dreaded events that were identified during the design of the security system;
- the constitution of databases prior to the intervention;
 - definition and scale of the intervention means, planned arrival dates of the means on-site,
 - functional modifications to launch within the infrastructure (closing the doors, stopping certain services, informing the public);
- the segmentation of the actions to launch after an incident;
- the real-time optimization of the security perimeters depending on the field data and the evolution of the situation.

These dynamic decision support tools lean on the communication and interoperability tools of the geographical information systems, and the many private and/or public databases which help take into account the procedures and the

characteristics of the infrastructure's equipment and constituents (chemical contents, mechanical resistance, etc.).

Thus it is necessary to use harmonious standards and procedures that will guarantee interoperability between the various databases and tools, and have transverse organizations and processes between the various actors which will guarantee the availability, the relevance and the accessibility in real-time to the necessary data. As has been previously pointed out, and has just been confirmed, the optimization of those elements can only be reached after considering the issue in its entirety.

8.4. The human factor

In the near future, the electronic systems dedicated to infrastructure security will raise early alarm, most of the time helping people act before the real crisis can actually start. However, if these systems can automatically trigger a certain number of prevention measures (for example: intelligent stop of subway circulation, closing of doors, triggering of smoke extractors, etc.), they will most often involve operators and intervention personnel. For these men, it will be necessary to see the information they need in the most natural and comprehensible way, avoiding any possible misinterpretation.

The man-machine interfaces are therefore an important feature of the security system, necessary but not sufficient: a crisis control center requires the intervention of a diversity of experts which must be coordinated. Beyond the technical solutions which facilitate the exchanges, robust and adaptative organizational processes must be implemented so as to take into account the unpredictability inherent to crisis situations.

Moreover, in the event where domino effects were to develop, it is crucial for the operators of the impacted critical infrastructure to closely follow the situation's evolution in real-time, and also have a predictive vision of it, so as to decide which means to implement and monitor the incident response teams active on the site. Ways to monitor the situation are also necessary in order to control the data coming from assorted origins and make the right decisions by optimizing the adequacy between the needs and the available resources (incident response teams for example).

8.4.1. *Monitoring*

In order to set up human means of intervention (rescue teams, intervention personnel) and monitor their deployment as well as their results, an operational control center is necessary, preferably located away from the infrastructure for security reasons, and allowing for the grouping, in one room, of the set of information necessary for the management of a crisis, namely:

- the visual display, as detailed as possible, of the situation in real-time;
- the predictive display of the probable evolution, including the display of probable domino effects;
- the real-time situation of the rescue and intervention teams;
- the information about the infrastructure's operation.

It should be noted that these pieces of information will most often not only be about the affected critical infrastructure, but also its neighborhood (the city block, the town, or even further away). For example, in the event of a crisis in an underground subway station, it is obviously crucial to know the situation within that station, but also to follow in real-time the effects on the roadway traffic around the block and even the town, the traffic having a direct impact on the time it will take rescue teams to reach the scene, since they might be delayed by traffic jams.

Moreover, beyond that visual display, crisis management integrates the management of the entire set of resources which will be necessary to ensure the successful running of the operations relative to the intervention, regrouping the installations, technologies, equipments and human resources. The absence or failure of one of these elements can contribute to the paralysis of the intervention team. It is therefore important to know these resources' availability in advance, as well as the instructions for the equipment's use and operation.

In parallel with the assessment of the situation and the analysis of the damages, the goal is to evaluate and determine the needs of the crisis management teams, according to the objectives, and to optimize the allocation and deployment of the resources for an increased efficiency, in particular when faced with a shortage of one of these resources.

To enable the mobilization, rollout, use, monitoring and eventual demobilization of the resources, decision support systems must optimize all the plan's elements, such as time, tasks, means, resources, constraints.

Depending on the tasks corresponding to the identified actions to lead in the planned interventions (for example: "define the security perimeters", "evacuate a

zone”, etc.) and the analysis of the operational plans, sequencing constraints are defined for the various types of interventions and the corresponding resources are established in terms of the volume of means to deploy (personnel and equipments) and in terms of maximum time allotted to these interventions. Of course this monitoring relies on the modeling of intervention operations, the use of resources, as well as the command of the combinatorics resulting from the allocation of resources to tasks when the global capacity is limited or when their simultaneous use is made difficult by the acknowledgment of a certain number of operational constraints.

As is easily understood, a single operator will have a hard time monitoring the totality of this information. Hence the necessity, as is already the case in modern operational security centers, of setting up monitoring stations run by the appropriate personnel depending on the organization of the entities that are called on: means of intervention, infrastructure, authorities. From the devices controlling the display and storage of the information, the communication with the critical infrastructure and the various organizations involved in the action, to the maps of the premises, the architecture of the operating core of security is an important aspect of operational efficiency and deserves appropriate care. Indeed, we must have access to a global synthetic situation that will, with a reduced number of screens, or sometimes an entire wall of screens, help comprehend the situation and its evolution; we must also have the set of determining factors on nearby stations so as to monitor the situation, and do so depending on the many points of views relative to the various intervention trades.

The organization and architecture of the security’s operating core must obviously be specified according to the studied infrastructure. We can however mention the following things:

- the solution that is aimed at must call on the fewer number possible of equipment so as not to scatter the attention of the operator working in the control/monitoring room;
- means allowing subgroups to work in parallel on hypotheses or different evolutions must be designed with the ability to quickly change hypothesis if need be;
- multimodal interfaces such as radio or hand-held digital devices must enable interaction with the system;
- the man-machine interfaces must be, if possible, “adjustable” by personnel on-site, according to the various profiles (trades, languages, special skills), either manually (with an entry code) or automatically (through wireless communication for example). They must also be adapted to stress situations.

8.4.2. *The man-machine interface*

As has been previously mentioned, the man-machine interface is an important component of a system. It is present on the various levels of an infrastructure security system, from the control center to the personnel scattered within the infrastructure and in charge, for example, of the first controls or the first interventions.

The purpose of such an interface is to provide the operator, as precisely as possible, with the strictly necessary information at a given time, allowing him to grasp the situation and its context, and monitor:

- some of the infrastructure's functionalities in real-time (people flow, delay on specific geographical points, status of parameters characteristic of the infrastructure's operation). Of course, this first set is highly dependent on the type of infrastructure and on its functionalities. For example, the surveillance of people flow can be achieved through control screens on which any abnormal event is signaled to the operator by a visual and auditive alarm. The automatic detection of, for example, people displaying abnormal behaviors in comparison with the expected behavior on that spot of the infrastructure will enable the precise display of what is actually useful to the operator and not of other scenes coming from other video cameras and not presenting any hazardous situation;

- the development of a crisis so as to handle it as well as possible. In that case, the nature and quantity of information can vary depending on the nature of the crisis, the present time, the type of analysis required by the user, and the user's skill level.

The security systems of the future might use man-machine interfaces featuring the following characteristics:

- on the level of the control center (which might be mobile), the availability of enriched maps which are:

- 3D with layered animations reproducing what a human observer would see at the present time in a site chosen by the operator, such as is the case today in many video games which use, with reason, an animated synthetic 3D representation which is more intuitive and more natural,

- 3D and predictive, showing the results of domino effects as well as the corresponding security perimeters,

- 2D, but dynamic and large-scaled, showing the available resources in terms of means of intervention (equipments, teams), availability of the access channels for these means, security perimeters;

– on the agents’ level, display of the local enriched map corresponding to their immediate surroundings, and providing them with the necessary characteristic for them to do their work, and inform them about their own safety. This information can be displayed on laptops or PDAs.

8.4.3. Training

Finally, this presentation would be incomplete if it did not breach the subject of training. It is indeed essential to train the teams, first so they can have a proper grasp of the system, secondly so they can practice the designed procedures, so as to allow the easy interoperability of the means and organizations, and finally so they can learn through repetition the emergency motions and reflexes.

This is achieved through the use of simulation tools and field exercises in realistic configurations unknown of the tested incident response teams.

Simulation tools are of course necessary in order to place the staff in charge of managing the crisis and coordinating the agents in a situation scenario. It is indeed highly complicated and constraining for the infrastructure to create a crisis within its own walls, even if the crisis is only “pretend”, that is to say it does not require physical attacks and of course spares human lives! Thus the necessity of virtually creating incidents of all sorts, provoking domino effects and “playing” with hypothesis on the crisis’ spreading.

These simulators must integrate, besides all the aforementioned maps, the acknowledgement, within said maps, of the probable movement of population within the infrastructure or close to it (city block, or even the whole city). To this end, the simulator must implement behavioral models of the human agents involved. These models must be as realistic as possible and thus able to take into account the individuals’ individual behaviors. For, in the event of crisis or events which put the individuals’ safety at risk, panic phenomena emerge, characterized by a great variety of behavior which cannot be assumed to be average crowd behavior, and is more akin to a collection of individual behaviors, acknowledging the various levels of emotionality, stress, culture.

Moreover, in order to test the teams and the equipment in actual size, exercises must also be staged on-site, despite the complexity and difficulty this poses on the infrastructure’s normal operation. To this end, the exercises are staged while the infrastructures are closed to the public (at night, for example), so as not to put too great a strain on the infrastructure’s normal operation: of course, we must then be cautious with their conclusions, for staging the evacuation of a subway station at night is not the same thing as doing it during rush hour with traffic of 60,000 people

per hour. The aforementioned simulation helps focus the exercises on the most critical problems. Caution is necessary, during on-field exercises, not to divulge the crisis scenario that is going to be simulated, so as to effectively test the system in an unexpected situation.

8.5. Conclusion

By presenting the protection of critical infrastructures as an illustration of the concept of system of systems, we have emphasized the problem's complexity, which is present in the three dimensions, physical, functional and organizational. One cannot neglect any of these dimensions without risking the failure of a security system when the crisis happens. The partial failure of the relief efforts after hurricane Katrina and the damages suffered by New Orleans in 2005 were a good illustration of this multidimensional problem and the criticality of a systemic vision of the whole value chain.

Neither technology, nor the existence of a plan on paper, can be sufficient answers. All these components must have been thought out and designed so as to operate together in times of crisis, hence the need of a global architecture and the necessity of training sessions for the use of the system of systems.

Emphasis has been placed on the technical and functional aspects, perhaps to the detriment of the organizational aspects. They are, however, just as essential. In particular, the most important points to study are: how does an organization restructure itself in times of crisis? How can a culture of risk management be elaborated?

Chapter 9

Globalization and Systemic Impacts

“Now, more than ever, our societies need new models to address systemic, long-term challenges like the climate crisis, poverty, pandemics, water scarcity, and demographic shifts. This will involve more business and government innovation, social entrepreneurship, public-private partnerships, and more effective civil society participation.” Al Gore, 2007, foreword to “*Capitalism at the Crossroads*” by S.L. Hart.

9.1. Introduction

For a holiday get-away, and with a little money, visiting the most exotic places is the easiest thing in the world: even destinations which might seem out of reach – who would have thought, just half a century ago, that the Arctic regions or the deserts would be tourist hiking places! – are easily accessible *via* specialized travel agencies.

But exoticism can just as easily come to us: for example, to celebrate Chinese New Year or enjoy a nice Indian or Mexican dish, or a regional specialty, we need only go to the restaurant or the nearest supermarket. Of course, viruses and microbes travel just as easily via airports or commercial boats, and an Ebola epidemic in the African jungle throws the entire planet into a panic. Pandemics such as AIDS, or pandemic threats such as the swine influenza, no longer know geographic

boundaries, and nobody, wherever they may be, can consider themselves out of reach.

The flows of information, travelers, goods, capitals, are ever more numerous and facilitated by technological progress. They create ever tighter links of dependence between individuals, cultures, nations, economic and financial systems, etc. If borders still exist, they are ever easier to cross and the world's vastness decreases day by day. This phenomenon, commonly called *globalization*, will be studied in this chapter.

We will notably see what makes globalization a true “system of systems”, according to the definition given in this book, namely the result of the optimization of a set of systems' value chain. The interest of this exercise lies in its engineering point of view on globalization, which might offer new interpretations on certain phenomena.

We will then somewhat reverse the addressed problem, by studying the way globalization, and its cultural aspects in particular, should be taken into account within systems of systems engineering.

For it may have an impact on the product of the engineering process, as well as the process itself, insofar as, on the one hand, the studied object, the system of systems, is potentially aimed at an international and therefore multicultural community, and on the other hand the teams which participate in the engineering of system of systems are also potentially international and therefore multicultural.

9.2. System of systems “globalization”

9.2.1. *Globalization: a concept with many meanings*

The notion of globalization pertains foremost to economy, as the consequence of economic flows, commercial as well as financial, in terms of raw or processed materials but also virtual holdings, the latter taking on an exponential importance with the development of digital society. However, nowadays, the term is broadened to encompass notions of culture, ecology, etc., just as those notions are reaching a global scale, due to the spatial (ever faster travel to any part of the globe) and temporal contraction (quasi-instantaneousness of data transfers).

If we consult the Wikipedia encyclopedia, which has become one of the symbols of this globalization through its universality, sought for as much because of the knowledge spectrum as because of the targeted international audience, we find the following definition:

“the term globalization refers to the developing of interdependence links between men, human activities and political systems on a worldwide scale. This phenomenon touches most fields, each with its own effects and temporality. It sometimes also evokes international transfers of workforce or knowledge.”

Globalization therefore covers all the phenomena resulting from the increasing opening up of the economy to foreign goods and capitals; and as such, it describes the increase of goods, services, workforce, technology and capital flows on an international scale.

In that way, it attests to a world without boundaries, with exchanges freed from all constraints but natural ones. To quote Joseph Stiglitz [STI 02]:

“[it] has reduced the sense of isolation felt in much of the developing world and has given many people in the developing countries access to knowledge well beyond the reach of even the wealthiest in any country a century ago.”

It should be noted that French people are *a priori* the only ones using the term “mondialisation” in the place of *globalization*. However, the term “globalization” is also used in French, with a slightly different meaning: if “mondialisation” calls on the notion of flow *a priori* without restrictions on the intrinsic nature of these flows but an acknowledgment of their diversity, “globalization” focuses on the global/local opposition, putting forward a holistic approach of the various issues. A symbol of this vision is the concept of a “global village”: coined in 1971 by the Canadian sociologist H.M. McLuhan, it highlights the interdependence stemming mainly from the acceleration of media and data exchanges from one corner of the globe to the other. It sketches out a world without borders, whether they are geographic, ideological, or economic, where interpenetration speeds up and becomes an integration within a single, common model.

9.2.2. *A long story*

Globalization, as a widespread exchange between the various parts of the planet, is a thousand-year-old phenomenon: besides the progressive migration of *Homo Sapiens* who progressively domesticated the environment for their own benefit, we cannot help but think about the armies of Alexander the Great, in the 3rd century B.C., and the Roman legions, which led to the mixing of cultures and economies from the Far East to the shores of the Atlantic, from Northern Africa to the deepest corners of Scandinavia.

From that time on, cultural and monetary flows and exchanges developed: Greek and Latin were the vehicular languages for centuries; silver and gold started circulating from the birth of the first civilizations, from Persia to the occidental extremity of the Mediterranean Sea; goods were exchanged as far as Asia, as attested by the discovery of Greek vases in China, and the creation of statues representing Buddha by Greeks living in India, statues which were then sent to Japan. For centuries, the main architects of globalization were, on the one hand, the conquering armies, and on the other hand the merchants, sailors, ship-owners, naval carpenters and bankers. In the Middle Ages, around the year 1000 and for nearly eight centuries, the independent Venice was the perfect illustration of this economic dimension, as the financial and commercial hub of a globalization centered on Eurasia, reaching its climax during the 14th and 15th century: as a matter of fact, the Venetian ducat was the “world dollar” from 1284 to 1797.

Between the 15th and 19th century, the geographical space stretched from America to Africa and the whole of Asia, and its metaphorical width promptly shrunk with each new invention, in the lapse of a century and a half: steam engines on earth and on water, railways, telegraph, telephone, planes, Internet, etc. A historical consequence of these advances, colonization during the 19th century resulted in flows of raw materials from the colonies towards Europe and was behind important population flows which deeply modified the distribution of the workforce on a worldwide scale, as well as the political balance.

The reduction of traveling times, for people and for goods, and the quasi-instantaneousness with which information and commercial transactions circulate, are the vectors of a globalization inscribed within an inescapable historical perspective which spans over several centuries. It should however be noted that the rhythm of globalization has not been steady: in the 20th century, for example, the process gained speed till World War I, followed by an important slow down till World War II, after which the process sped up once more. It is estimated that, around 1970, goods exchanges once again reached the relative share of the global gross domestic product that they had achieved around 1910, and have been increasing ever since.

As a consequence of the reduction of distances and the exponential growth of exchanges, said exchanges had to be standardized in order to regulate the commercial and financial flows. Those standards take the form of trade agreements (for example, the Hanseatic League, an alliance of trading European cities, in the Middle Ages, or the free exchange treaty between France and England in 1786, which, because of the width of those two powers’ colonies at the time, concerned a majority of the globe) but also of standardization of certain goods: in this way, the universal postal Union, created in the second half of the 19th century, gave vast possibilities of exchange to international messages and a guaranteed confidentiality,

by making sure the various postal services would accept postage of letters issued by the other services of the treaty's signing states.

The internationalization of the national economies and capitals is not a recent phenomenon. The United Provinces (today's Netherlands) were the center of the world's economy for the second half of the 16th century and the 17th century. They were then relayed by France and England, those two countries controlling about a quarter of the world's trade on the eve of the French revolution. Internationalization quickly developed and was already strong at the beginning of the 20th century: in 1914, Great Britain owned almost 41% of international assets, and France owned 20%. Beyond these numbers, another proof lies in one of its negative aspects: the 1929 crisis, which stemmed from the interdependence of investments and capitals.

The negative reactions to globalization are just as old: already the Stoic philosopher Seneca lamented it in his tragedy *Medea* in 60 AC:

“Pure were the ages our fathers saw, crime being far removed.
Each person inactive, keeping to his own shore, grew old in his
ancestral fields [...] Every boundary has been removed; cities have set
their walls in new lands, and the world, now open to travel
throughout, has left nothing in its earlier seat.”

Of course, other reactions are inversely enthusiastic, from the marvelous travels of Marco Polo to Asia, to the tribulations of Ibn Battuta through Africa and the East. Some negative effects globalization has on local economies, such as are sometimes described today, are not new either: in the 12th century and for 20 years, gold lost most of its value on the shores of the Mediterranean Sea, for the emperor of Ghana had distributed the enormous quantity of that metal which he had brought with him on his pilgrimage to the Mecca.

Globalization, and this will conclude this longitudinal panorama, is a phenomenon inscribed within a historical perspective, and has always had deep repercussions on economic, financial, cultural and social dimensions.

9.2.3. *The facilitating factors of globalization*

Thomas Friedman's bestseller, *The World is Flat*, defines three major globalization stages:

- 1492-1800, the era of the world's discovery – beyond the sporadic Scandinavian expeditions in Northern America, or Chinese expeditions in Africa, which did not have any lasting consequence – reduced the spatial dimensions;

- 1800-2000, the industrial era, reduced the economic space with the development of multinational companies;
- 2000-today, the era in which the individual becomes an agent on a global scale.

Moreover, he lists 10 factors which have facilitated the development of the last stage: the collapse of Berlin's wall, Netscape and the Web, work flow software, uploading, outsourcing, offshoring, supply-chaining, insourcing, informing, and collaborative working environments. Let us briefly revise these chronological factors.

The fall of Berlin's wall gave way to new exchange grounds (geographical spaces, economic and political structures, populations) for the development of globalization, and put into serious question the unsteady geostrategic balance of the previous decades. In parallel, the information technology quickly developed, with the emergence of web search engines: beyond the physical reticular structure which had been around for years, the availability of simple tools to find information led to the development of a community of non-specialists using the Internet as a place of exchange. Then, following the standardization of protocols and software, increasingly complex software platforms were built, encapsulating *middlewares*, business software and specific software developments that were shared by the developers' community.

In turn, this helped the implementation of all kinds of data exchange platforms, through which anyone can download the information they seek, which in turn led to the emergence of new services providing specialized information, and a new so-called content business. A true industry of information search was born with the Google giant, and the goal became not only to provide information, but to provide the best information with the shortest delays. Beyond data exchange platforms, true collaborative platforms have appeared, allowing for the joint collaboration of varied skills in the digital world, far from geographical borders.

Thus, new services are provided, offering functional optimizations which were until then only performed within enterprises, and are now available worldwide, which leads to a restructuring of enterprises and a renewed acceleration of globalization.

Indeed, if geography may be partially erased from the end-to-end pooling of services via virtual collaborative platforms, value analysis suggests we go from vertical integration models, in which the enterprise seeks to control all the trades necessary to the manufacturing of its products, to horizontal integration models, in which the enterprise will focus on its core business by seeking partners who will optimize its value chain. This may naturally require other partners to be entrusted with trades for which the enterprise used to have sole responsibility: this favors

globalization even more since those partners may well be located in highly remote geographical zones.

Beyond these factors, it is interesting to see that the general attitude of users, but also of suppliers, seems to have gone from a hierarchic and relatively centralized “command and control” mode to a distributed “collaborate and connect” mode. However, if we look more closely hasn’t this movement been part of the globalization process for centuries? It only seems more evident today, maybe because spatial boundaries – at least on the scale of the earth – are closing in on us, unlike centuries when there were always new territories to explore and therefore the hope for endless growth.

9.2.4. *The necessity of a systemic standpoint*

In order to really understand some of the levers of globalization, and define some room for maneuver, we must look at it from a systemic standpoint. First, some factors must be identified: the various agents and the relationships between them, the different value chains that are established, the critical parameters within those value chains, and their qualitative influence on a local if not global scale.

Since the 1980s, part of the geographical community, among them Olivier Dollfus, interprets the world both as a geographic object that can be isolated and possesses a certain uniqueness, in short a system, and as a combination of interlinked socio-economic systems of inferior levels, that is to say a system of systems in its combinatorial aspect. This gives us two complementary interpretations:

- the “system earth”, with the planet as a global environment, which aims to develop a view of universality which goes beyond the various scales of observation which may conceal this search;
- the “system world”, built on a set of interacting components, going beyond simple juxtaposition, where the decisions taken by a government or a company may have remote repercussions: “in systemic terms, it is a set of sets (nations, human societies, cultural spaces, companies, markets, etc.) interacting one above the other, entangled, self-organizing within constant evolution” (O. Dollfus, in *l’Information Géographique*, 1990).

In reality, this systemic vision is part of the current intuitive approach, if only through the systematic use of the word “outsourcing” as soon as a task is entrusted to an agent deemed to be an “outsider” to our own system. But everything depends on where we stand: entrusting part of the activity of a company’s department to another department is not outsourcing on that company’s scale; entrusting part of a

company's activity to another company on the same regional or national territory is not outsourcing in terms of the regional or national economic development, etc. The same thing applies on the scale of the economic community (European Community, Mercosur, NAFTA, ASEAN, etc.) which is actually often described as being regional! And on the scale of the planet, which is increasingly relevant in economics, the concept of outsourcing cannot exist and must be replaced by the concept of flows, either of raw materials, of processed products, of knowledge, etc.

This systemic perspective relies on the identification of the system's components and the relationships between them, the nature of these relationships helping us understand the position of the components in relation to each other. It also highlights the global effects: political, economic, ecological, social. The simultaneous acknowledgement of these various dimensions constitutes the search for the optimization of the value chain at the basis of our definition of a system of systems.

9.2.5. *The various dimensions of the “globalization” system of systems’ value chain*

The various dimensions we must take into account are in direct association with the flows revealed in the globalization issue: migratory flows, financial flows, flows of raw material, goods and services. If the first flows mainly have an impact on the social and cultural dimensions, the second flows evidently pertain to the economic dimension, and the last flows concern the economic, cultural or even environmental dimensions.

Migratory flows are the main vector of the globalization process, regardless of the chosen dimension. From the dawn of time, driven either by survival instincts (search for food or spaces to cultivate and farm; environmental pressures caused by climatic evolutions; overpopulation in regions where the subsistence of all is not guaranteed anymore), either by the thirst for conquest, or the quest for knowledge, populations have moved from one territory to the next, one continent to the next, beyond natural obstacles, following the quirks of nature – glaciations, droughts, floods – which facilitated certain passages, or new inventions, in order to cross, ever more safely, increasing distances within (*a priori*) hostile environments. These flows were the basis of the cultural melting pots, the expansion of certain languages and scripts, of certain diseases and therefore the development of the immune inheritance of the various populations that were affected. However, for the last centuries, migratory flows have mostly had socio-economic causes: admittedly, they are today's manifestation of yesterday's survival instinct, but more than that they are the idealistic search for an environment more favorable to the thriving of the individual or his family, sometimes provoked by political persecutions, but more frequently by the different economic situations of both territories (the migratory flows from

country to city are as much a product of these flows as migrations from Europe to the United States during the 20th century). These flows translate into two major effects which can be crudely qualified as brain drain and flexible workforce.

The first is mainly the result of the flow of students who study abroad, favoring various parts of the world, and do not come back to the country they were born in, or on the contrary study in their home country, and later go to work in another part of the world. In both cases, the flow has an immediate impact: if the student is trained in another country and settles there, the first country suffers a loss of potential expertise, and we can imagine the imbalance it could result in if the movement increases. The example of the exodus of part of the intellectual elite from Germany to the United States before World War II, notably in the field of physics, is striking, even if the movement was mostly driven by the desire to escape political persecution. But an important flow of students coming back to their home country after having been trained abroad is also a potential source of imbalance, for these students will in a way have taken the place of other students in the country they studied in, students who will therefore not have followed that training. The economic effects are easily grasped, insofar as the production of goods and services differ depending on the training level of the agents of the production chains: we need only take the example of South Korea, where the proportion of PhDs is one of the highest, PhDs which were mainly acquired in the United States and helped South Korea reach the top of the competition in some high-tech sectors. For the last few years, the same trend can be observed in India in the service industry, and in particular IT services.

As for the availability of the workforce, as a consequence of certain migratory flows, it has multiple consequences on a short term basis as well as a long term basis. On a short term basis, this workforce is economically interesting, since it is generally composed of populations without high qualification, suffering from a greater lack of job security than the native population, and therefore less inclined to negotiate their work conditions. These two factors *a priori* explain why this workforce can receive smaller wages, which allows companies to cut production costs and therefore both stimulate investment on the premises – which can curb offshoring – and increase supply. But migratory flows concern the person, who can immediately step in as part of the workforce, as much as their family. The allowances and services provided for the whole family can eventually put a strain on the finances of social services. On a strictly economic level, beyond any political or ethical consideration, we are clearly facing several combined effects, and there is no easy or systematic way to measure their influence on the global value chain, in its social as much as its economic dimension. We should neither neglect the cultural dimension of these movements of population which, without any judgment of value, have impacts on many levels, whether it be linguistics (slang is found in many languages, from French and American, to the various Creole tongues), music,

culinary arts (see the “Indian” cuisine in England and South Africa, but also the so-called Southern American cuisine of Mexican inspiration, or the Cajun-Creole cuisine of Caribbean inspiration).

The financial flows concern the flows of capitals invested in speculation as much as the various international investments. The latter are a source of employment, but also offshoring, depending on the adopted standpoint as to the source and destination of the investment. An essential component of financial globalization for centuries, the financial flows, concerns the optimum allocation at a given time of worldwide resources (or at least on the scale of the currently known world, if we go back to a historical perspective).

We recognize here a familiar approach in the context of system of systems, which is not surprising insofar as the phrase “financial system”, or rather “financial systems”, is commonly used. Since the optimization happens at a given time, it is clearly possible for the reached optimum to only be local and not necessarily stable, such as what happens in some nonlinear systems: this is typically the case with speculative bubbles, and experience has shown the fragility of such optima, with the disastrous consequences that the bursting of such bubbles can have on a capitalistic level.

The analogy which has just been made with nonlinear systems can be used in the explanation of certain phenomena, and will be brought up again in later sections. As an example, let us note the rapid propagation of local effects, which is characteristic of some nonlinear systems due to diffusion mechanisms (this is what happens when heat quickly spreads within a sheet of metal exposed to a heat source): we find it in the propagation of some financial crisis, such as the Asian Financial Crisis in 1997-98, which had worldwide repercussions due to the tight imbrication of the various financial systems. Many economic experts actually underline the way globalization can provoke what can be called systemic crisis: they build on the importance taken by the financial markets, and inevitably imperfect or deficient information on the part of the financial actors. The latter is known in nonlinear systems theory as the impossibility to exhaustively know the initial terms, and leads to phenomena of so-called deterministic chaos, popularized with the image of the “butterfly effect”. The crisis can be triggered by mimetic behaviors: do we not frequently hear that the stock exchanges in one part of the world will plummet because the stock exchanges in another part of the world closed at their lowest? From the standpoint of systemics, such reactions heighten instability and behaviors such as divergent spirals, calling for strong regulation measures to curb the crisis. Which leads, in the long run, to quasi-cyclic evolutions, a succession of stages of expansion, debt, rising interest rates, speculation, bubble bursting, deflation, regulation, etc.

Let us now look at the last type of flows, which concern the raw materials, the finished goods and services. It has always been a crucial vector of globalization, from the Silk Road to the spice road, the latter being one of the factors of geographic expansion which led to the discovery of the New World. More dramatically, it has been a factor of the development of slavery on several continents and over several eras, stemming from the need for cheap workers who could work in hard conditions and meet high production demands, whether in the extraction of various rocks, or the exploitation of agricultural resources. Slavery itself brought about major social, cultural and economic changes on the long run.

Today the flows of raw materials are more important and critical than ever in the globalization process, insofar as they happen on a global scale and are factors of political instability, sources of conflict but also of financial revenue (the “black gold rent” for example) which change the political balance and the zones of regional power, with international repercussions (such as demonstrated by the crisis situation in the Middle East these last few decades). These flows of raw materials are counterbalanced by flows of finished goods, which go in opposite directions and themselves result in financial flows, with economic and social consequences: the search for the financial optimization of these various flows leads to a search for lower production costs and therefore a cheaper workforce. This being said, we should also take into account, within the value chain, the ethical and now ecological factors, as seen in Chapter 1: we no longer accept buying cheaper products when they are manufactured by children, just like we now accept to pay more for products when they are environmentally friendly. Therefore, the product must be at the same time economically, socially, and ethically acceptable.

The same thing applies to services: which is why in English-speaking call centers, many of which are located in India, and more recently in Indonesia, the employees learn to speak with a British or American accent (this is why India subcontracts in Indonesia, for the workforce there is even cheaper, and because of the events of the second half of the 20th century, the Indonesian population has had many more contacts with Americans and therefore possesses better empathy towards American callers) depending on the geographical zone they are working for. In French-speaking call centers, often located in North Africa, employees give out names which do not reveal their origins. These tricks contribute, on the one hand, to establish a relation of proximity between the call center and the caller, and on the other hand, to avoid a potential rejection towards the offshoring of the service in another part of the world.

This set of flows therefore has an impact on various dimensions, dimensions which should be taken into account if we wish to benefit from globalization rather than suffer from its downsides. Moreover, the flows are dependent and their various effects are built in a non-trivial manner. Optimizing the value chain is therefore a

difficult exercise, and, in light of the current world situation, we are allowed to think the job is unfinished. Since we consider globalization as a system of systems, this means we do not yet control it, which is not surprising considering the complexity of its architecture, as has been partially demonstrated in the previous sections. Beyond those rather negative observations, in which the systemic vision puts forward the problem's complexity and the high difficulty in apprehending it, let us now try and look at the more positive answers it can bring to certain problems.

9.2.6. The utopia of a standardizing globalization

The aforementioned analogy to nonlinear systems is a reassuring factor against the fear of a standardization imposed by globalization. For example, not only is cultural standardization not unavoidable, but it is unlikely, since cultural assimilation and adaptation is almost always followed by counter-movements, in various ways: in Chaucer's time, the English fought against excessive Frenchifying, and today the French are fighting against the reverse, but some American purists are also hunting for French expressions, and using them is all the rage in some circles; in France, the reactions against the hamburgers were rather radical, and some people complain about being invaded by American pizza chains, when in fact hamburgers are a consequence of a strong German immigration and pizzas come from the Italian Diaspora, both of whom have conquered America before conquering Europe.

The standardization of the national forms of capitalism is just as utopian, as pointed out by many economists. Indeed, in the punctual search for local optimization, a standard solution is not the best global solution, that is to say a solution stable in time: what is aimed at is a locally optimal solution which can leave room for important reaction abilities if necessary. As a slogan, we could say that resilience wins the fight against standardization in the optimization process. From the standpoint of economics, therefore, there cannot be just one solution, either total submission to a market, or economic interventionism denying all virtue to the market mechanisms. This analysis is overly simple, and excessively linearizes the system of systems; on the contrary, there are many intermediate options, whose terms vary depending on the time and place.

Even if globalization is progressing, the image of a world completely flat and devoid of borders is only utopian: as a matter of fact, the ratio between capitals invested by companies outside of their national borders and the total amount of capitals invested throughout the world is, on average, only of about 10%, and even the waves of acquisition did not raise it higher than 20%. Comparing the flows of tourists going abroad in relation to the total touristic flows, we find the same 10% average.

The implementation of a company such as Coca Cola in international markets is still 10 times less than in the American market; the various strategies for a borderless globalization, whether it be centralized management unified from one geographical zone to the next, or the standardization of products, have all failed to make the company grow. As a matter of fact, more than 400 different brands of Coca Cola exist today, compared to the handful in the 1960s. Among the companies listed in Fortune 500, Coca Cola is featured alongside a dozen companies which manage to reach at least 20% of their turnover in each of the three regions: North America, Europe, Asia-Pacific. This clearly demonstrates the effective limits of globalization today.

Even the Internet, so often presented as a borderless territory, is not really so: we need only look at the various national laws which each country's subscribers must abide to, whether it be France forbidding the sale of Nazi objects on Yahoo! in 2000, or the USA government putting a ban on online gambling in 2006. The global network is therefore more akin to a collection of networks on the scale of Nation-states than a global unified network.

The realization that the purpose of globalization, whether it be today or in the foreseeable future, is not the search for uniformity and the erasing of all borders, is the best proof that we are faced with a system of systems, and not with a system, which would be evidenced through a unity devoid of the separations which we have previously described.

9.2.7. The use of new systemic interpretations to understand the mechanisms of globalization

The systemic vision offers a grid of interpretation of globalization all the more interesting because it can be operational, if not explicative: indeed, by doing the exegesis of the system of systems' definition and exploiting the systemic analogies as much as possible, some actions for globalization become accessible – for example relating to the efficiency, or lack thereof, of its regulation under certain terms –, actions which would be offered for a system in the same conditions. The purpose of the following paragraphs is to exploit this interpretation, and in some cases offer leads on the understanding of certain situations.

9.2.7.1. Control parameters

In a systemic approach, the differences between the various dimensions must be taken into account as much as the similarities. If geography, and the distance dimension in particular, is a crucial factor in the success or failure of globalization, starting from the simple need to access a particular geographical zone (with the presence and the quality of the harbors, roads and railways), it is closely related to

the cultural, administrative, and economic aspects, which turn out to be either aids or hindrances.

The cultural aspects obviously regroup the language, but also the traditions, tastes and habits and the consumer standards. The administrative aspects concern the regulations aimed at foreign companies, the principles of national or regional preferences, the protectionist reflexes (towards raw materials or finished goods), but also the exchange facilities between the members of some communities. The economic aspects mainly concern the cost of workforce or management, as well as the possibility of cutting some expenses.

The value chain's performance can be improved by playing on the following elements, while keeping in mind the previous aspects:

- acquire larger market shares by covering as much and as well as possible the range of needs through the differentiation of products and services according to customer specificities (examples: personal preferences, constraints, geography, will to pay, etc.) and through the improvement of the perceived value (examples: perceived benefits, brand image, etc.) not by selling them cut-rate but by a tailor-made price strategy, so as to achieve the maximum amount of value and possibly increase the scale of production;
- break down the acquisition, manufacturing and use costs, and search for various optimization levers (depending on the economic aspect but also the administrative one, possibly by taking advantage of administrative disparities: taxes, etc.);
- reduce, or at least optimize, the risks, in particular financial and economic, to try and achieve profits.

To understand how these elements contribute to the value chain, we must remember that from a strictly economic standpoint, value can be defined as the product of volume and margin, with the margin being the sum of the competitive advantage (that is to say the investment in comparison with the competition's, minus the costs, still in comparison with the competition's) and the industrial margin. The previous elements have an influence on that value which is neither linear nor trivial.

9.2.7.2. *Regulation*

The systemic interpretation brings us to consider globalization as an open system of systems: indeed, even if globalization implies a worldwide scale, it has not yet reached that scale and there still exists an “outside” to the perimeter directly concerned by the globalization process, a perimeter moving with the geostrategic evolutions and some political changes.

However, flows link the outside to the inside, even if they do not have the same intensity and are not in the same category as the flows which exist within the globalization's perimeter. These flows can either be seen as exchanges between the system of systems, which represents the perimeter of globalization, and an exosystem, or as disturbances to which said system of systems would be subjected. In both cases, regulation is needed to force the system of systems into the desired working condition.

To further the analogy with dynamic systems and their control, regulating globalization is akin to defining a policy of command-and-control, first estimating the state then implementing governing laws. If we translate these concepts within the context of globalization, this means the first requirement is to monitor globalization so as to evaluate, as exhaustively as possible, the dimensioning factors and their mutual dependence, and act on them if necessary. Monitoring is actually implemented on several globalizing dimensions: for example, the World Health Organization (WHO) in the field of public health, the International Atomic Energy Agency (IAEA) which monitors the use of nuclear energy, the World Trade Organization (WTO) which supervises trade on a worldwide scale, etc. These monitoring organizations are also responsible for defining and enforcing the principles of worldwide governance: whether it be in the control of nuclear, biologic and chemical proliferation, or in the field of healthcare (see the worldwide warnings about risks of an avian flu epidemics), the field of security (by sharing and exchanging information within the fight against terrorism or more generally for the control of borders within regulated zones such as the Schengen Area), etc.

All governance principles aim at regulating the flows, whether they are physical flows of raw materials and finished goods, or immaterial flows, flows of energy or information. Regulation has a precise purpose: to avoid the creation of permanent imbalance which would set off divergent spirals such as can be observed in unstable dynamic systems. The geopolitical consequences would then threaten to go beyond limits deemed reasonable or at least tolerable, and local crisis might evade control and evolve into global crisis.

This is why even in fields which are *a priori* subjected to free trade, such as finance, and in order not to re-experience international crisis such as those leading to the dramatic historical events in the first half of the 20th century, some organizations act as lenders of last resort so as to curb a crisis's growth: central banks on the level of States and the International Monetary Fund (IMF) on an international level. Governance and regulation are actually enforced via permanent actions on the system, just as the controller in a dynamic system must draw its energy from somewhere: thus the Tobin tax regulates the flows of capitals and enables the implementation of new rules.

These few examples have no purpose other than to illustrate the validity of the analogy drawn between the various concepts, but they also attest to its benefits. Moreover, in reply to those who might fear that those governance organizations are in fact constraints limiting all action and enslaving local authorities, we only need recall that a complex dynamic system shall not be so easily controlled in any situation without a supervising law that can locally adapt to the system's particular state.

9.2.7.3. *System interfaces*

As a system of systems, globalization features some interface requirements. Those demands can vary in nature: technical, linguistic and cultural, administrative and legal, or even environmental.

Technical interface requirements are usually featured within documents of technical specification. They express the system's prerequisite ability to interoperate with others, keeping in mind that it is sometimes necessary to take into account the compatibility of standards on either side. A common example concerns electrical appliances in which the voltage and plug can differ from one country to the next.

Beyond technical requirements, taking the human factor into account in the definition of system interfaces has become crucial, since the human being is an integral part of the "globalization" system of systems. Besides the need for system interfaces to integrate linguistic and cultural specificities, the human being himself can act as an interface. Such an interface, far from being new, has always been part of the globalization phenomenon, with roles such as interpreter, emissary, ambassador, etc. These roles were formed *a posteriori* as an answer to a need for communication between countries and people. However, migratory flows have led to individuals naturally gifted with multilingual and multicultural skills, thus forming a new kind of "interface" within globalization.

Locally speaking, the administrative and legal environment could sufficiently curb the use of flows of raw materials, of workforce, as well as the financial flows, etc. Dedicated interfaces must be planned in order to conciliate and align the administrative and legal requirements of the involved parties. An example representative of this kind of interface would be an agreement between states on the taxation of their respective expatriated citizens. The articles of law featured in such an agreement would then correspond to both administrative and legal interface requirements.

As for environmental interfaces, one of the most popular examples is ecodesign, which takes into account the requirements on sustainable development in product

and service engineering¹. Going with a worldwide desire to protect the environment and natural resources, and notwithstanding its possible political, scientific, economic, ethical justifications, ecodesign enforces architectural choices as well as constraints as to the maintenance and the disposal process. The “pastille verte” in France (a green road-tax-disk indicating that a car meets certain environmental requirements), the organic certification, logos featuring the Earth, are some of its symbols.

9.2.7.4. *Control on the edge of chaos*

In a general manner, nonlinear systems do not follow the regulation principles of linear systems, where the local and global can be identified. It is therefore necessary, in relation to what has been previously demonstrated, for businesses to learn how to co-invent and co-evolve products and services, in order for them to fit the ecosystem and local culture. Local solutions are then sought in order to trigger global behaviors.

The paradigmatic example, crowned by the Nobel Prize given to Muhammad Yunus in 2006, concerns microfinance. Via this mechanism of bank loans, which cannot exceed a few hundred, or even a few dozens dollars, individuals can build small trades or businesses on a scale of one or several people, in countries belonging to what is called the bottom of the pyramid (better known by the acronym BOP), which means the billion of human beings living with less than a dollar per day, or even the few billions living with less than four dollars per day. These microcredits, which might look derisory to us because of the loaned sums, actually promote local entrepreneurship on an individual scale, and these seemingly small-scaled actions contribute to the economic development of a part of the population usually left out of such initiatives. It is actually expected that, via those local initiatives, the targeted populations will not only find a revenue source, but also the feeling of pride they do not enjoy when surviving only through international help. These initiatives would therefore help fight terrorism, insofar as the poor and neglected populations might otherwise be an easy recruiting pool for terrorism organizations.

Such initiatives demand a total rethinking of the cost structures, and in particular a drastic diminution of the investment costs, for the existing infrastructures are completely different from the infrastructures of the so-called Western World (the density of road and telephonic systems is utterly different). The institutional infrastructures are also completely different (the notion of property is radically

1. The notion of sustainable development was coined by the Brundtland Commission in 1987, following the Club of Rome's 1972 report on the dangerous effects economic growth was having on the environment, and the United Nations' 1972 conference on the human environment, which insisted on the contradictions that might arise between the development's objective and the conservation of the ecological balance.

different in some parts of the globe, depending on local cultures and religious beliefs), as are the cultures of local populations (they do not share the same individual and social aspirations). This leads to a revision of the whole value chain, mostly of its economic and sociocultural dimensions: for example, a bank granting microcredits cannot function perennially like the banks we know in the occidental countries. Indeed, “standard” opening hours, reception centers with counters and heavy infrastructures, self-service ATMs in random places, all of this comes with prohibitive costs in environments where security is not on the same level and the cost for the transfer of population is enormous compared with their daily activity. This is the way new financial services should be created, so that they can be at the disposal of the destitute client in the right place, at the right time; providers must be located in safe areas, such as police buildings or post offices. We therefore see how globalization, as an ideal granting every part of humanity, no matter how destitute under our standards of economic comparison, access to services deemed elementary under the same standards, demands an approach both creative and working on a case by case basis, far away from any standardization of practices under the pretense that they have been tried and tested for certain parts of the population.

Other examples of such “creative capitalism”, sometimes promoted by governments, exist in the healthcare field. In the speech he gave in Davos in January of 2008, Bill Gates, as the founder of the Bill and Melinda Gates Association, quoted a law passed in the United States in 2007 which guarantees priority review of a company’s product (in this case via the Food and Drug Administration) if it is also developing a new treatment for diseases such as malaria or tuberculosis, which strike the developing world. Therefore, the development of a new treatment for malaria by a pharmaceutical company can help it have one of its cholesterol drugs put on the market a year early, which represents an increase in profit equal to tens or hundreds of billions of dollars.

The previous examples seek global effects through local actions, which we could in-short describe as “micro-decisions for macro-effects”. If we carry on the analogy with nonlinear systems, this would be akin to the famous butterfly effect which has been largely mediatized on the subject of dynamic systems, illustrating the fact that a butterfly flapping its wings on one side of the planet would lead to a tempest on the other side.

Other situations sharing the analogy to chaotic phenomena are possible: this is the case with local crisis, which bring to mind some turbulent phenomena in which tornados appear, move, vanish, modifying the flow around them. To pursue this analogy, control of local crisis would mean looking to dissipate them by assuring they did not reach too big a magnitude, rather than rushing to resolve them. In some works on management, this is called control on the edge of chaos. We can

legitimately wonder whether this has an application in the issue of globalization, in particular with the regulation of some flows, and the reaction to some crisis.

We will conclude these sections on the systemic interpretations of globalization by pointing out that they bring out the importance of a thorough analysis of the performance of the globalization system of systems' value chain, and to opt for the appropriate governing, neither overly centralizing nor overly simplifying, insofar as the general principle, already quoted in Chapter 1, of the required Ashby variety can be applied: the system's complexity provokes, *de facto*, a complexity on the same scale as any regulating principle. The analogy with nonlinear systems teaches us that, with a higher freedom in regulation capacities, it becomes easier to maneuver towards local optima, and therefore gain partial command of the system of systems.

9.3. Beyond the concepts of systems

We have just grasped all the complexity of globalization by introducing the various dimensions around the major axis that is the value chain. This being said, beyond these concepts, we are taking into account a dimensioning characteristic of these systems. As a matter of fact, we are dealing with human-intensive systems. This crucial characteristic, associated with the properties of nonlinear dynamic systems, raises the question of perverse effects, and more globally of the paradoxical character of the behaviors which have appeared, behaviors which must be reported and which must be taken into account by the systems of systems' governance to achieve the utmost pertinent regulation. These are the two points we are about to discuss.

9.3.1. *Human-intensive systems*

For a long time, theories on economic models were based on a strict postulate: the rationality of human agents. Practically echoing those theories and economic models, ethical value models were based on utilitarianism. Logically, with such concepts, it was possible to conceive artificial agents displaying isomorphic behaviors. It was the golden age of artificial intelligence, and later of distributed artificial intelligence. This remains the dominant dogma of people who defend a so-called liberal ideology – far from the foundations of the original liberalism, in 18th century Europe.

Herbert Simon's Nobel Prize in 1978, and Daniel Kahneman's in 2002, both won for their works on denouncing this postulate about the rationality of agents in aid of an economy which D. Kahneman called "experimental", shows the distance which has been crossed in the past decades by researchers in the field of economics.

The economic agent is only one dimension among many of human beings, who cannot be reduced to *homo oeconomicus*.

The human being is part of social and cultural networks. Members of those networks form social groups, which delimit and regulate the behavior of and interactions between each member. The perimeter of those regulations is vast, from the monitoring of sexual partners in order to preserve and reproduce the original group, to the choice of legitimate professions, the mandatory revenues for each rank, the lifestyle, to give back, gift for gift, the proper social rituals in accordance to each given rank. All this creates a cultural heritage, both immaterial and material, which on one level is complementary and interdependent with the aforementioned chain of values. The second level of that heritage concerns the identity. The identity of the human being is linked to the group, which admits him as a member and therefore gratifies him or, on the contrary, rejects him and sanctions him. In this context, human beings seek admittance to the group, or rather groups, they belong to, and the associated gratifications. Sanctions can either be symbolic or physical. For example, a woman who, at one of her nephew's wedding, is obliged to eat alone in the kitchen, cut off from the other guests, because she has converted to a different religion from her family.

Those elements frame and structure human daily life, from the rhythm of the day or week, in pace with prayers or religious holidays, dressing habits, prohibited foods.

This gives a meaning and a direction to life. This symbolic dimension, which structures the world, or even the worlds, within which human affairs and the links between humans and material goods play out, cannot be reduced to economic relationships, and yet structures and depends on those economic relationships. What has a symbolic value here has a commercial value there, and the two are not always interchangeable.

Whether it is the sudden flow of tourists, the worker migrating to a potential "El Dorado", the pensioner retreating to a country where life is cheaper, each of them has his own values, his habits, his traditions, and his own way of interacting with other human beings. In this way, each of them is contributing, thanks to globalization, to a disruption of the current balance, which can in turn jeopardize their own projects.

9.3.2. *Perverse effects and paradoxes*

In a developing country, a tourist town is very popular among traveling Europeans. Hotels are multiplying. A nice residential area is built, welcoming sun-

seeking European pensioners. To them, the products on sale seem cheap, which encourages them to spend money. The first consequence is inflation. But in this developing country, the salaries are low. This rise in prices is immediately felt by the local population, for whom life suddenly becomes much more difficult. This situation is grounds for rancor and hostile movements, which can find the arguments leading to violence in a difficult background, be it real or imaginary.

This rhetoric can also deepen the divide between the lives of the European tourists and the lives of the local population gravitating around them, as well as between the lives of that population and the population not enjoying the benefits of tourism. Moreover, those people's traditional living makes them the target of tour operators, which basically showcase them like animals in a zoo. This situation also nourishes tensions. The town, the tourists and the clubs, are seen as decadent, devoid of morality, divesting the local population of its traditional morals, values, practices, its territory, its identity.

We could imagine that these dynamics do not apply to humanitarian actions, which would therefore be immune to those perverse effects. But this is not so, as confirmed by the recent Arche de Zoé (*Zoey's Arch*) scandal. The same desire for standardization drives the view of Occidental countries on the democratization of other countries, neglecting the fact that traditional societies already have their own modes of power and social regulation, which are, on many levels, just as democratic at their basis, even if their forms differ.

Everything which, in traditional society, would be natural, becomes problematic. Globalization disrupts the local balance of authority and power, on a financial and symbolic level. This situation generates crisis, conflicts, and nourishes the various current tensions. The mutual interdependences give a global dimension to local crisis.

Are we facing an irresolvable situation? The answer is not clear-cut. Risks can be opportunities, and vice versa. In the 1970s, the oil crisis triggered actions for the reduction of energy consumption. Paradoxically, it spurred research and development in the European motor industry. The same thing is happening today with the development of hybrid vehicles.

There is no "neutral" action, nothing devoid of consequences. There are no strictly positive or strictly negative effects, either. In a way, we are dealing with a paradoxical situation which is, therefore, indecipherable. And it is within this paradoxical logic that we must find new modes of regulation, to monitor crisis and curb unexpected effects.

9.4. Globalization's impact on systems of systems engineering

As has been pointed out, analysis of the effects of globalization depends on the chosen standpoint. Thus, from a geostrategic and political standpoint, the bipolar world of the Cold War has become multipolar in our era. On societal and social levels, the contrast between rich and poor increases; the emergence of the Internet brings with it the notion of a digital divide. As for the industrial and economic dimensions, the emerging markets and the BRIC economies (Brazil, Russia, India and China) stand next to the industrialized countries within globalization.

Following the definition we gave at the beginning of this chapter, we are trying to present our analysis from the angle of the industrial and economic dimension, which still allows us to put forward the necessity of taking into account the various factors, whether cultural, administrative, or geographic, in the designing of systems.

9.4.1. *New opportunities and new challenges*

Numerous economists agree that the overall (macroscopic) effect of globalization benefits society. The antiglobalization and alter-globalization defenders are mostly denouncing the local effects (microscopic), which leave some people stranded. The difficulty lies in the management of such local effects through a better redistribution of resources. Without joining the politico-media debates, we are convinced that globalization offers new opportunities both to the industrialized countries and to the emerging markets and developing countries.

For some, globalization offers the possibility of concentrating and acquiring further value through integration (verticals, horizontals, conglomerates) of partnerships outside the nation's borders. Despite the strengthening of the controls and the laws defining anti-competitive practices in the United States and in Europe, recent statistics show that the volume of mergers and acquisitions has reached a record high during the first quarter of 2007 (\$1,130 billion, that is to say a 14% increase over a year, according to the *Financial Times*). There are strong odds that this tendency will continue, despite the morose environment of the financial sector following the explosion of the subprimes market. It should be noted that this type of strategy is not strictly reserved to industrial countries. Some BRIC countries are also gathering benefits from it, if you look at the recent attempts at mergers and acquisitions initialized by the Indians (e.g. MITTAL's successful takeover bid on French ARCELOR) or the Chinese (e.g. CNOOC's takeover bid on American UNOCAL).

The opening of borders has helped the distribution of consumer goods in industrialized countries but also in less developed countries, where the middle and

upper classes are increasing in numbers, notably in India and China, with the impressive performances of their economy. Even the access to luxury products is becoming more democratic, as is made obvious by the evolution of the mass luxury market (*masstige* concept describing the alliance of luxury products and mass consumerism). In answer to the profusion of consumer products, businesses are tempted to offer products or services with better integration, as can already be witnessed in the telecommunication field, where the key objective is to achieve convergence of solutions and platforms while offering advanced mobility functionalities in order to gain the maximum value. Thus, the products and services are not only more sophisticated, they also have to answer the demands of cultural, geographical and environmental specificities. A good grasp of these specificities is then crucial for a system integrator seeking to optimize a value chain whose concentration is increasing, as well as heightened performance.

The economic performance is not only measured in terms of global revenue, which means that the concentration of value is not enough. Profitability also plays a part. Reduction of costs or increase of productivity can help reach that objective. A solution is to externalize the production in countries where the workforce is cheap: also known as *offshoring*. The group study [KAM 06] shows that governments shouldn't interfere with offshoring and free exchange and that, in the long run, the process would both benefit the source and target countries. This conclusion is not surprising considering that offshoring consists of a division of labor to increase productivity, as suggested by the economist Adam Smith in his book *The Wealth of Nations*. This division of labor leads to a specialization of works, with the country receiving the offshoring having a competitive advantage over the company's home country, notably enjoying a wealth of production factors associated with that advantage. In that way, the Heckscher-Ohlin model, used on the function of production, explains that the developing countries are competitive thanks to a massive and cheaper workforce, while developed countries have the advantage in terms of innovation and technologies.

From that observation, it is easy to establish that an opportunity for industrialized countries would be to focus on advanced technologies and innovations, or the integration of systems requiring qualified workforce in order to optimize the value chain. Therefore, it would be judicious to leave merging or developing countries focus on the production of parts and systems at the most, before they are in turn able to compete with the industrialized countries with products and services with a high added value. In the meantime, we must not forget that the splitting of work in the four corners of the world should go along with a stage of integration of the products and services which are being made externally. From that point of view, a certain number of factors must be considered within the analysis of the globalization's direct impact on system design. The most challenging factors are cultural and administrative. They will be studied in detail in the following sections.

Other factors must also be taken into account: for example, the geographical factor naturally translates into climatic or seismic specifications, which instantly play on the products' design and manufacturing; the economic factor influences system design through energy requirements, the latter being more or less pronounced depending on the country, etc.

9.4.2. Cultural factors

The influence of culture on design is already visible in domestic products. Let us take the example of refrigerators: Germans require a bigger space for meat than Americans; as for Italians, they prefer to have specific compartments to store vegetables. As for ovens, they are bigger in Great Britain, where people roast turkeys for Christmas, than in Germany, where poultry is cooked in a different way; likewise, Germans do not need auto-cleaning ovens as much as French people do, since they usually cook at lower temperatures.

Well-known examples in the food-processing industry demonstrate the importance of cultural differences in customer satisfaction. Following that reasoning, Mac Donald's sells different specialties in different countries: McSpaghetti in the Philippines, Teriyaki McBurger in Japan, with two rice patties instead of the traditional burger bread, etc. Likewise, music producers adapt the musical contents (rhythm, melody) to the tastes of the marketed audience, for example with boy bands in the 1990s. The same thing happens when automobile designers adapt their products to their end-users' cultural specificities: from the position of the wheel and the piloting components, which of course differs depending on traffic rules, to the country's favored data communication modes and options.

Other cultural specificities which must be taken into account in system design can be the degree with which specifications are met. Many businessmen, wanting to manufacture some components in China, have had some nasty surprises. The manufactured components did not satisfy the requirements listed in the technical specifications! This can be explained by the way Chinese culture focuses more on the oral word than the written word. Moreover, the focus is mostly put on appearances. Therefore, a product whose looks conform to the specification might turn out to be of very poor quality when it is put into actual use.

The relationship with time also varies from one culture to the next. Some cultures are very scrupulous in meeting time delays, while others are less so. The perception of time can also greatly differ. In other words, the notion of emergency depends on that perception. Those nuances can be greatly penalizing if they have not been taken into account in the system design and the project management. Typically,

you would need to control time, more precisely time zones, in a project involving teams scattered on different geographical sites.

This is all the more important since, with the growing complexity of systems of systems and the globalization triggered by the search for the proper skills, multinational teams of collaborative engineering are becoming common practice. However, those teams follow different modes of management depending on their original country: such is the case for the Boeing teams multilocalized in Seattle, United States, and in Russia, after the acquisition of their aeronautics manufacturers. The same goes with IT companies, in which work habits greatly differ between occidental Europe or the United States and India or Asia.

When Cultures Collide by R. Lewis illustrates such cultural diversity in activities as reunions and negotiations: from one culture to another, one nation to another, each person's role inside the group differs greatly, especially towards the group's leader. Relationships and hierarchy inside the group, whether they are predefined or settle in, are deeply related to the group members' cultural background. It is therefore easy to understand how these processes and the tools that implement them can sometimes be difficult to use when the teams' cultural reference is fundamentally different to the reference prevailing among the teams which designed these processes and tools.

Beyond the necessity to take the cultural differences within the engineering teams of a system of systems into account, there is therefore the matter of the implementation of the methods and tools destined for that engineering. The previous thoughts lean towards the existence of different ways of unrolling the engineering process according to the local culture, and reflect that imposing work and behavior models which are culturally foreign to some is not optimum; on the other hand, this means a higher integration level will have to be designed, through which the working models and exchange between the various teams will have to be organized. Engineering with multilocalized teams then takes on a capability dimension and the various engineering teams become the constitutive systems of a true system of systems, in its turn responsible for the engineering of the produced system of systems!

9.4.3. Administrative factors

The administrative factor mainly translates through different standards: thirteen major standards for electrical outlets, as well as different voltages and frequencies. It also translates into design requirements, for example for the ratification of aeronautical systems (let us remember the troubles the Concorde went through to be ratified in the United States, back in its time).

Poor knowledge of local regulations can also be fatal. For example, in order to restructure an acquired French company, a Chinese corporation thought it could act the same way as in China and fire employees *en masse* without taking any precaution. However, the corporation was quickly confronted to the French work code and the working syndicates. Its restructuring plan, meant to optimize the performances of its production system, was therefore not implemented.

The aspects concerning intellectual property must also be taken into account, notably for systems which feature protection against piracy in their basic functionalities. A local legal system's lack of maturity, which means it cannot protect copyrights, might impact the chosen solution for the systems' design.

9.5. Conclusion

In Western Europe, the naves of so-called Christian cross churches are oriented East-West, and their transept is oriented North-South and locate on the East side. Houses in China are oriented North-South with the main door located on the South side, and public spaces are located South whereas private spaces are located North. None of this was left to chance. The symbolic and religious dimensions (geomancy for the Chinese world, picture of the cross and localization of the celestial Jerusalem for the Christian world) were part of these buildings' architecture entry data. Whether in cities of God such as monasteries, or in plantations in Martinique, we find such an association of the symbolic, religious, economic and architectural dimensions.

Beyond the purely functional and utilitarian perspectives that guide the engineering of artificial systems, other dimensions are sometimes present, consciously or not: from the symbolic Hippodamus of Miletus's ideal city, where the spatial and social organization are the reflection of an ideal republic's political organization; Richelieu's city; Speer's architectural utopia; an architecture sometimes tinged with functional utilitarianism such as Stalinian buildings; to the historic-religious motives driving the Christian cities of the Middle Ages or the Muslim medinas, the spiritual dimension is more than underlying and demands as global a systemic perspective as possible, beyond the simple functional or logical analysis advocated by the current concepts of systems engineering.

Such is the multidimensional approach we have tried to justify in this chapter, using globalization as an excuse as much as a study case. And indeed, after studying how the standpoint of systems of systems could explain some of its characteristics, we went on to discuss its influence on systems of systems engineering. Without reaching definite conclusions, we can see in this global desire to take into account factors that are not only physical or easily standardizable, a step towards the

building of a definition of the complex systems which surround us and which we actively belong to, as designers or users, with or without our knowledge.

9.6. Appendix: a summary of the properties of nonlinear dynamic systems

The goal of the following paragraphs is to summarize – without any mathematical formula or undue rigor – some of the basic properties of nonlinear dynamic systems, so the reader can fully grasp this chapter’s analogies, in particular about the way the systemic standpoint on globalization can be put to profitable use.

Let us restate that a dynamic system is characterized by the evolution of a set of parameters through time. These parameters are *a priori* separated into input and output variables: the input variables, also called state controls, can be manipulated – via controllers – to modify the system’s behavior; the output variables, also called state observers, can be measured – via sensors – to quantitatively evaluate the system’s behavior. In order to mathematically characterize the system’s behavior, it is common practice to introduce other so-called state variables, which are designed in order to help fully and precisely reconstitute the system’s history thanks to their accumulated knowledge. Depending on the system, or rather depending on the system’s adopted modeling, these state variables can be continuous, discrete, probability, etc.

The system is called *linear* when it satisfies the principle of superposition of states, which means that the sum of two state variables of the system is also a state variable. This property actually has important consequences on the level of the system’s dynamics: because of it, the local knowledge of the dynamics within a region of the state space is enough to know the dynamics in the whole of the state space. In other words, we can instantaneously go from the local to the global.

On the contrary, a *nonlinear system* can exhibit a multitude of behaviors in which locality prevails over totality: from one point of the state space to the next, the behavior can be drastically different, for example stable (the state is nearing zero), unstable (the state takes on arbitrarily high values), erratic, etc. Moreover, there also exist nonlinear systems in which behaviors can drastically differ in places arbitrarily close to the state space, a phenomenon which can be disconcerting, but which has gotten a large amount of press coverage via the vast literature on chaos theory.

On the subject of *chaos*, the term is *a priori* associated with random behavior and a high sensitivity to initial conditions: a simple example features a rectangular billiard table, with fixed circular obstacles and a ball rolling on the table; the ball goes through elastic collisions (which implies that the trajectory will be symmetrical from the point of collision) with the table’s edges and the obstacles. Two balls

leaving in slightly different directions will eventually follow very different paths, through the amplification of the angle between those two directions. Another example of so-called deterministic chaos behavior is found in the atmospheric convection modeled by Lorenz in 1963 through three coupled ordinary differential equations.

Any general consideration on a nonlinear system's dynamics must therefore be made with the utmost prudence insofar as nothing can guarantee that knowledge in a particular spot can give any kind of information on another functioning spot of the system.

This is all the more important when the problem of the system's *regulation* arises, that is to say the right way of manipulating the input to obtain a certain behavior. Indeed, while linear systems are easily controlled through the use of relatively simple means, the situation is completely different with nonlinear systems. When getting closer to a desired behavior, we might easily enter a part of the state space in which the behavior is harmful.

This being said, let us not put too much of a damper on things: if systems with extremely pathological behaviors do exist, and we should keep them in mind before coming to hasty conclusions, they are not necessarily met in practice, etc., just as we do not always meet overly simple linear systems. Without going into particulars, the state space of some systems can be divided into regions, within which the behavior is quasi-linear and can therefore be easily regulated, the difficulty lying instead on the level of the borders between regions. Regulation then takes on the shape of a bi-leveled mechanism: the lower level is dedicated to the regulation of one particular region; the higher level supervises the whole, choosing the adequate regulation depending on the space region, and monitoring the crossing into the various regulation modes.

Finally, let us quote an important theorem in the study of nonlinear systems, the so-called stable and unstable manifolds: in substance, it declares that the state space of a nonlinear system can be broken down into two subspaces, one corresponding to stable modes, and the other to the unstable ones. The stable modes are easy to supervise, and the unstable modes are the modes giving important dynamics to the system, insofar as the trajectories undergo an exponential divergence of orbits. The skill of the supervision level previously quoted can reside in the intelligent use of those instabilities to reach more stable regions.

9.7. Bibliography

[DIA 97] DIAMOND J., *Guns, Germs and Steel: the Fates of Human Societies*, W.W. Norton, London, 1997.

- [FER 08] FERRIS T., “Ethnic culture and the systems engineering process”, *Incoze Insight*, vol. 11, n° 1, January 2008.
- [FRI 06] FRIEDMAN T.L., *The World is Flat: a Brief History of the Twenty-first Century*, First updated and expanded edition, Farrar-Strauss-Giroux, New York, 2006.
- [GHE 07] GHEMAWAR P., *Redefining Global Strategy: Crossing Borders in a World Where Differences Still Matter*, Business Harvard School Press, Boston, 2007.
- [HAR 07] HART S.L., *Capitalism at the Crossroads: Aligning Business, Earth and Humanity*, Wharton School Publishing, Upper Saddle River 2007.
- [KAM 06] KAM L. *et al.*, “Should governments restrict outsourcing?”, Executive MBA Group Case Study, London Business School, London, 2006.
- [LEW 05] LEWIS R.D., *When Cultures Collide: Leading, Teamworking and Managing Across the Globe*, 3rd edition, Nicholas Brealey Publishing Ltd., London, 2005.
- [MAT 02] MATHIEUX J., *Mondialisation: les nouveaux défis d’une histoire ancienne*, Editions du Félin, Collection Histoire et Sociétés, Paris, 2002.
- [PAS 00] PASCALE R.T., MILLEMANN M., GIOJA L., *Surfing the Edge of Chaos: the Laws of Nature and the New Laws of Business*, Crown Publishers, New York, 2000.
- [PRA 04] PRAHALAD C.K., *The Fortune at the Bottom of the Pyramid: Eradicating Poverty through Profits*, Wharton School Publishing, Upper Saddle River, 2004.
- [SIL 03] SILVERSTEIN M.J., FISKE N., “Luxury for the masses”, *Harvard Business Review*, Boston, April 2003.
- [SIM 05] SIMON H., JACQUET F., BRAULT F., *La stratégie prix*, 2nd edition, Dunod, Paris, 2005.
- [SIM 06a] SIMANIS E., HART S.L., “Expanding possibilities at the base of the pyramid”, *Innovations Journal* (available online), winter 2006.
- [SIM 06b] SIMON H., BILSTEIN F., LUBY F., *Manage For Profit Not For Market Share: A Guide to Greater Profits in Highly Contested Markets*, Harvard Business School Press, Boston, 2006.
- [SIN 02] SINGER P., *One World: The Ethics of Globalization*, Yale University Press, New Haven and London, 2002.
- [STI 02] STIGLITZ J., *Globalization and its Discontents*, Fayard, Paris, 2002.

PART 2

Systems of Systems Engineering, Methods, Standards and Tools

Chapter 10

Methods and Tools for Systems of Systems Engineering

10.1. Systems of systems engineering: from the control of complexity to the necessity of a model-driven approach

Acknowledging systems of systems leads to the collaboration of distinct systems, none of which would be able to fulfill the mission on its own, but which can do so when grouped together within a framework still to be defined. Each system, which is specified, designed, developed, implemented and maintained potentially independently from the others, corresponds to a specific project involving distinct agents (prime contractors, prescribers of needs and technical prescribers, general contracting project teams, industrial project managers, industrialists).

This raises the question of the evolution of methods, both of acquisition and of use: distinct systems, entrusted to various teams and project managers, each with their own life cycle, must now be managed in a consistent and concerted way. Moreover, these systems must be integrated within vast sets, whose architectural definition (technical as well as organizational) may change, hence a new effort to take the interfaces into account, from their specification to their withdrawal, without neglecting the possible impacts this common implementation may have on the constitutive systems (taking into account the flows coming from other systems, physical compatibility problems, etc.).

It becomes clear that, because of the importance of the resulting flows (material, energetic and informational), controlling such an increased complexity in time will be difficult without appropriate methods and tools. The basic assumption is that systems of systems engineering is not fundamentally different from systems engineering in terms of processes, as was discussed in Chapter 1, and is also widely accepted by the community (preliminary versions of the American Department of Defense guide to systems of systems engineering; works taking into account the notion of systems of systems in the ISO/IEC 15288; ongoing work of the “system of systems” AFIS workgroup, the *Association française de l'ingénierie système* – French Association of system engineering; French chapter of the INCOSE, International Council on System Engineering). Fundamentally, this assumption is based on two main observations: first, a system of systems can immediately be seen as a set of systems, the latter being seen as sets of products or services supplied to fulfill a certain purpose. Therefore, the process of system engineering can *a priori* be applied in a recursive manner.

Secondly, the system engineering processes, as standardized by the ISO/IEC 15288, have become true business processes, that is to say they organize the interrelated activities of a group of persons and resources, the responsibility of which is to acquire and contractually provide products and/or services. This translates in the way the system engineering process is broken down into processes of contractualization, business (strategy, resource management and quality management), project management and technical management of the system life cycle. These activities are sufficiently generic and completely cover the studied field, whether it is about systems or systems of systems.

If, in the absolute, the “classic” methods and tools of system engineering can be applied, it quickly becomes clear that the volumes of data resulting from engineering activities and the interrelationships between such data are of such importance that there can be no slackening in their application, and any avoidance of the professional use of those methods and tools – which might be done in the isolated context of a system, as long as the risks were controlled – then ensures failure, as much in terms of the fulfillment of the capability objective as in terms of budget management, at a given moment in time, during evolutions of the objective and/or components of the system of systems.

We are also taking as a basic principle that, to control complexity, we need more than a documentary approach focused on the traceability of the data which constitutes the various documents of the process of system engineering (traceability of requirements, traceability of projects and test results, traceability of evolutions and configuration management). It requires a model-driven approach, bringing an additional level of abstraction (and therefore of simplification) compared to the introduction of each system as a means of exchange between project teams. It should

be noted that these models, which by definition are necessarily partial representations of the system, can, from a certain standpoint, be of greatly varying natures: functional flows models, technical models, architectural models, economical models, etc. The analysis of functional architectural models can, for example, help improve the understanding of the dependences between the functions already covered by the inherited systems (hardware and organizational) and those which future systems ought to provide. This provides us with a tool to reduce redundancies, duplicates and possible critical paths. The traceability of models is a natural prerequisite and helps achieve global control of the system of systems, including the management of its positions within its life cycle in a capability approach. Moreover, these models help stakeholders (prime contractors, users and project managers) communicate and reach agreements thanks to their level of abstraction, which helps define a common level of interaction freed from the constraints of technical implementation. They also provide leads for the improvement and evolution of systems of systems and their main components.

We will therefore talk of requirement engineering, based on the use of models representing various views of the studied systems, which will help, via architecture approaches, to more efficiently control the balance between the configurations of the systems and the inter-system architectures to the specifications of need, and to envision the management of the system of systems configuration based on the analysis of the value under cost constraints (taking into account the systems in their current state of maturity, with no possibility to redevelop the constitutive systems or upgrade them to the required level) and time constraints (the capability increments must be controlled according to the evolution of the context of use on the one hand, and the availability of constitutive systems and their physical or virtual interfaces on the other).

We will also talk of test engineering, whose purpose is to verify and validate the capability of the systems to be tested on their architecture, which will lead us to bring up in detail the validation aspects, in particular the models used for the design of systems of systems architectures.

Finally, we will mention the necessary simulation tools as well as the collaborative work and concurring engineering tools which allow us to instrument the previous processes within activities of specification, definition, development, implementation and maintenance of the systems. Of course, these tools can only help if the various project teams involved can communicate in real-time and take into account the evolution of other systems, and if the teams in charge of the systems of systems are able to pronounce the necessary arbitrations in case of conflicts. The tools and the project teams then become the components of a system of systems themselves, the added value of which must be the control of complexity of the system of systems (we will not overuse this referential circularity)!

However, we will not talk about the management tools for the contractual, budgetary, financial and schedule aspects of the projects, which are considered mature and relatively independent from the increased complexity of their context of use. Indeed, even if the evolution of such issues demands new initiatives, for example in terms of innovative funding and contractual flexibility, the regulation frameworks are still *a priori* fixed and the follow-up and feedback requirements are still the same, regardless of the purpose of the contract.

10.2. Architecture

10.2.1. *Architecture: an ally of systems of systems*

Architecture comes from a desire to unroll a clarifying and simplifying process, on various levels – business, organizational, technical – in order to optimize the strategic steering and therefore the global performance of a company in terms of creation of value. If the term was first used in the field of information systems (note the use of “information” and not “computing”, contrary to a common mistake, which properly shows the strategic lining up of the approach going far beyond a simple search for technical performance), it has gone further than this strict framework and now applies to the global company.

To begin with, to ‘architecturize’ is to find a way to segment as well as find directing construction principles that will allow the information system and the informatics to evolve along the same rhythm as the strategy and organization, with a desire to anticipate. The most obvious metaphor is of course the city: when it grows beyond a certain size, a town is faced with individual and collective needs which can no longer be easily satisfied and may lead to major malfunctions: insalubrity, insecurity, congestion, damages to the architectural environment, etc. A classic example is the Parisian agglomeration of the 19th century, which suffered from those various problems in the course of its more or less anarchic growth. The Baron and prefect Haussman, commissioned by Napoleon III, started this urbanization rationale – which spread over almost thirty years – by dividing the town into *arrondissements*, districts and blocks, by defining the large common infrastructures (large communication axis, sewers, public lighting, parks, etc.), by distributing the responsibilities among the city and the *arrondissements* and by setting construction rules.

The success of this approach can be seen in the fact that over a century later, the inner city of Paris has managed to absorb a very large increase of population (on the housing, energy supply and waste disposal levels) as well as a significant increase of the daily commuting population, which greatly exceeds the population of the inner

city, all in acceptable conditions and while preserving the architectural wealth of the city, which makes it one of the most famous tourist attractions in the world.

In the same way, the first architectural phase of a company consists of dividing the main functional areas in order to set strategic and organizational maneuver margins, followed by a second phase during which the applications with identical functional perimeters must be divided by standardizing the exchange and sharing of data, and finally a third phase which transfers these previously identified principles, rules and divisions on the technical systems of the company. To begin with, this enables the distribution of the various responsibilities and the creation of economies of scale (non-redundancy and simplification of the flows and exchanges) while leaving degrees of freedom for later strategy evolutions (new functional areas and externalization of some functional areas without bringing into question the value company's creation of value); and subsequently, it means that technological choices do not need to be locked, something which would later remove any reactivity in the event of major technological innovations as well as any capability of evolution in order to take into account radically different needs without bringing into question the entire technical implementation.

The situation of large companies within their context brings into play geographical, legal, political, economical, sociological, economical, etc. entreaties, which have direct repercussions on the organization and operation of the company. Being able to adapt the latter to the evolutions of the entreaties is a survival criterion for these companies. The adaptation to systems of systems¹ is direct: it is about controlling the various key parameters of the value chain, by taking into account the clients and users' general needs (which corresponds to market research) and the product families or services available in the short/medium/long term (hence the products/markets pairs). The architecture approach can thus be applied to systems of systems with the triple requirement of taking into account the strategic, organizational and technical dimensions.

The basis of the architecture work is the search for strategic and organizational invariants, followed by the setting up of common technical authoritative accounts which guarantee common information to several businesses or activities, and the control of the technological developments. The technical data authoritative accounts are a set of standards, methods and tools which help specify the semantics of the operations, with the structuring and coherent set-up of information at stake. The goal is not about standardizing everything in a centralized manner, since that would mean

1. A system of system is a framework of systems which can potentially be acquired and/or used independently, for which the designer, buyer and/or user wishes to maximize the performance of the global value chain at a given moment in time and for a set of conceivable frameworks.

not reaching the intended flexibility; more than anything, it is about dividing the authoritative accounts according to activity areas while achieving the correct compromise. This definition of authoritative accounts is essential in order to build up partnerships and achieve external growth, which are valuable goals for any system of systems, including those thought to belong to business areas that are captive or under constraints of national autonomy, such as defense. This is what happens to partnerships which become widespread at the industrial level, within the development and maintenance phases, as well as the military users communities level, within the framework of coalition operations. The same thing applies to the search for external growth, even if it bears different names such as fair return or risk and expenses sharing within the development and maintenance phases, and interoperability or common operation in the use phase. The authoritative accounts provide the architects and the development teams with references needed to achieve the integration, and also enables the ascending compatibility of the data in circulation in the event of an upgrade and evolution of the system of systems.

The architecture materializes into an architecture plan, which defines the functional or application quarters, the data flows between those quarters, and for each quarter the technical authoritative account(s) in terms of data models, interfaces models and tools policy. Moreover, on a global level, a general technical authoritative account must also be defined with its data models and its possible tools policy in order to facilitate as much as possible the portability (capability for a technical resource to be independent from the software and hardware infrastructures which accommodate it and thus to be reusable within different infrastructures), openness (capability to accommodate new features) and modularity (capability to define and accommodate blocks likely to be added, removed, redone and reused in other functional or organizational contexts).

10.2.2. *Application examples: combat direction system within the naval aviation system of systems*

As an illustration of the approach, we hereby present some thoughts carried out within the framework of the naval aviation system of systems composed of naval and submarine platforms, as well as the various naval aviation components on board, without forgetting the sensors, weapons and communication and decision systems. In the next paragraphs, we will focus on the combat direction system, which is a critical element of the system of systems and a focal link of various issues.

Above all, the identified needs of a combat direction system are for it to be a strong, reliable and effective system, with reduced carrying costs and manufacturing lead time, while being easily portable and reusable on various platforms (submarine, surface) and able to adapt to various physical architectures or to the specific needs of

the operators. Lastly, its life can span over dozens of years, hence the necessity of being able to easily adapt to new functional needs (progressive nature) and new technologies (sustainability and obsolescence management).

Combat direction systems are notably fitted in surface vessels (aircraft carriers and frigates) and submarines, particularly for exports, when they are led to interface with systems different from those we know. They interface with detection systems (optronic sensors, infrared, radars, sonar and probes), weapon systems (guns, missiles, torpedoes, helicopters, unmanned systems and combat aircraft) as well as various information, command and communication systems (tactical data liaison terminals, etc.). Obviously, they have to take into account the constraints of the various marine platforms (surface/submarine) on which they can potentially be fitted.

On the operational level, they contribute to various warfare domains: anti-aircraft warfare, anti-submarine warfare, anti-surface warfare, anti-mine warfare and land-oriented action (the naval aviation system of systems can support coercion operations in the littoral zone and even up to a few dozens miles inland). They are not restricted to a single vessel; and within the framework of multiplatform management in the naval aviation force, they form a true combat system of systems. The main functions are: tactical data liaisons, weapon control, aircraft control (including command-control of the unmanned aerial vehicles), handling of the situation (meaning the visualization and updating of friendly and enemy forces) for each platform, but also the fusion of multiplatform plots (with a view to establish target trajectories).

Generations of combat systems succeed one another and their development has been incremental. The architecture approach was necessary to monitor the evolutions yet to come. An authoritative account of the functions and external interfaces was thus defined, which led to a first division into 451 capabilities organized into a hierarchy, able to adapt according to requirements and weakly coupled, which were then grouped into 44 coherent blocks which can be specified, verified and managed in development by an industrial project manager, or in assembling by the integrating project manager. These blocks were then organized into layers in order to structure the dependencies: indeed, in such layered architecture, a layer's component relies only on the services provided by subjacent layers. Dependency loops are thus forbidden. Seven layers were defined, following the usual philosophy inherited from the 1984 ISO/IEC 7498 standard on the Open Systems Interconnection Reference Model (OSI):

- the hardware layer concerns the various equipments, calculators, consoles, networks and physical interfaces with the agents;

- the firmware layer concerns the various operating systems and their software extensions (drivers, security mechanisms, self-tests, etc.);
- the middleware layer concerns the data distribution services, network exchanges, resource access services and brings together the shared technical components;
- the “support” layer groups the common technical services (data management, man-machine interface presentation) such as component management (assembling, roll-out, life cycle management), system data management (persistence, replication, transactions), creation and sharing of visualization elements and portability services with regards to the graphics environment;
- the “base area” layer groups the stable components from one combat decision system to another, which represent the base of the business, particularly the management of leads and tactic objects, of environment data, of emissions, of mapping and of various records;
- the “area” layer groups the components which can be found from one combat decision system to another, operated and/or configured in different ways. These are business components which deal with the tactical data liaisons, the management of monitor and weapon systems, the monitoring of external agents (sensors, data liaison, other systems and sub-systems, aircrafts) and the planning, command, training and replay functions;
- the “combat decision system” layer is specific to an occurrence of a combat decision system or a mission, and deals with the system configuration, operators’ roles and external interfaces. On this layer are found the configuration elements that meet the needs of specific clients in terms of man-machine interface or missions.

The first four layers (hardware, firmware, middleware and “support”) represent the technical base, which can evolve in terms of technological choices without questioning the functional architecture (“area” and “base area” layers) and vice versa. The technical base allows the technical architecture code to be factorized, which prevents the scattering of specialized routines more or less buried in the entire software system and thus facilitates technical architecture optimization operations. A technical policy can also be designed for this base, such as the use of components on generic frames (library, framework) and of standards guaranteeing the durability of the developments. The “base area” layer allows us to envisage a wider interoperability between the combat decision systems, particularly thanks to the common definition of the basic data of the area (format, authoritative account and semantics) and the possible centralized exploitation of the records.

The advantages of this layered organization are that each layer can evolve independently from the others, that interdependencies between modules are limited

by the layers, and that it's possible to define a simple organization of the development with a manager per layer of the technical base and a configuration management per layer. We will see later how this layered organization also makes reverse engineering possible. It should be noted that, just like in any architecture, the layered architecture is business-oriented and that for instance there is no specific layer of man-machine interface since each component can potentially be composed of a man-machine interface part, whose only requirement is to interface with the services of transport and data recording provided by the technical base. This facilitates the modularity of man-machine interfaces, the separated development and the possibility of carrying out unit tests for various components.

10.3. From architecture to detailed design: reference architectures

10.3.1. *Reference architectures*

Architecture is only the first step towards the eventual development of a solution which will answer the users' needs. It represents the prerequisite to the architecture design phase, which will provide representations of the system in terms of functionalities and of some implementation characteristics.

The importance of architectures resides in the description of the system without all the design details, in the identification of the critical interfaces, and in the understanding of the allocations between functions and components. Depending on the architecture plan, it thus enables the construction of the systems views in order to then launch the detailed design and development phases. Let us remember that the most commonly used views are respectively: logical (set of requirements and links between them), functional (division into functions and links between them) and physical (sub-systems and hardware or software components, and interfaces between them with macroscopic definition of the flows of data and information or energy).

The architecture facilitates the functional and physical division and thus produces initial diagrams which facilitate the definition of the two main views, functional and physical respectively. Besides, it also makes the standardization of the contents of these views easier, and thus enables the comparison of views between systems architectures, hence a possible change of scale in order to define the architecture of the system of systems from the architectures of the constitutive systems. If the functional blocks are thus shared or if they all have a common reference point, it is even easier to design means to assemble them; and on the physical level, the division of technical solutions facilitates the design of technical interfaces and minimizes the impact or compatibility research. We will come back to those aspects later.

Within the model-driven engineering framework, the architecture definition phase is no exception. We are thus talking about architecture reference models or reference architecture. These are abstract descriptions in terms of entities (components, for instance) and services (functionalities, for instance). The difference with the very notion of architecture may seem subtle. As a matter of fact, in practice, architecture reference models, architectures and implementations can be seen as successive steps towards the development of a solution: in an attempt to establish a clear distinction, we could say that the reference model alone doesn't allow the development of any solution, whereas the architecture allows a partial development.

The idea is not to abstract on a whim, but to aim for optimum reuse while trying to clear new levels of community (for instance between requirement sub-sets, which leads to logical architecture architectures). The significance of a system of systems perspective then becomes clear, since this abstraction provides the means to compare and agglomerate the component architectures without having to do the work all over again.

The value of architecture comes from the way it is used. It is clear that architecture reference models have a high added value since they can easily be used for mutual communication and information between design teams. Architectures facilitate the decision-making process by providing the decision-maker with the necessary information and facilitating its reuse in other ways. They are the essential bridge between the strategic-driven thoughts (concept analysis, return of experience on the usage doctrine) and the technical activities of development and production. The level of abstraction is one of the difficulties since these top-level models suit big decision-makers but not the practical users. Vice versa, a low abstraction level might overload the task with unnecessary details; hence the advantage of sharing common languages and established standards. That is why some normative authoritative accounts have been developed in recent years on the various layers' levels. Here are a few examples, as an illustration, from the lowest layers – closest to the hardware – to the highest layers which fall under the province of potential application business areas:

- CASE (computer-aided software engineering) reference model;
- OSI (open systems interconnection) reference model;
- IEEE POSIX (portable operating system interface) architecture, built on the OSI reference model, a real-time interface specification between Unix-type operating systems;
- TAFIM (technical architecture framework for information management), once imposed by the United States Department of Defense (US DoD);

- HLA (high level architecture) for the interoperability of distributed simulations, initially proposed by the US DoD and almost immediately adopted by NATO as well as various defense ministries;
- JTA (joint technical architecture), imposed by the US DoD, a supposedly universal architecture able to instantiate into other architectures according to the business areas, e.g.:
 - JAUGS (joint architecture for unmanned ground systems), for terrestrial robots, and in particular the command-control management between the robot and the ground stations, whose aim is to not be dependant on a proprietary industrial solution both for vectors and ground stations,
 - JAUS (joint architecture for unmanned systems), which generalizes JAUGS to all aerial or underwater robots,
 - CAF (C4ISR architecture framework), for the C3I systems (command, control, communication and intelligence);
- at the NATO level, all architectural works done for the NC3B (NATO Consultation, Command and Control Board) for the interoperability of the C3I systems, from the operational views to the technical views as well as the reference dictionaries, test architectures, etc.

The last example is very important. It is a NATO approach that harmonizes practices between various nations, establishing links between various levels: indeed, the goal is the intervention within coalitions whose geopolitical context is likely to have a strong influence on the constitution, sharing of responsibility and level of access and sharing of information! This approach is currently used by the French Department of Defense, where NATO interoperability standards must be applied. Let us also mention the role of France as frame-nation (of the aerial component of NATO, meaning the capability to provide a projectable command structure able to lead an operation defined by an “operation contract” which can, for instance, be expressed in terms of aerial outputs). Hence, architectural works are required on every level imaginable in order to authorize interoperability on demand, or not. Let us emphasize the fact that interoperability, as seen by NATO, covers three levels: physical interoperability (existence of a communication link, fixed or not, not necessarily provided by information or communication technologies – typically, voice can be a communication medium), procedural interoperability (a protocol and syntax must be known and used) and operational interoperability (which refers to the operation of the system in the context of other system use, via usage doctrines and conventions linked to the interpretation of information and thus the construction of meaning; this aspect comes under semantics, unlike what was evoked in procedural interoperability). Therefore, the architecture must not only focus on the technical

interface aspects but it must also be tightly paired with thoughts on organizations implemented when the system of systems is being used.

10.3.2. Two examples of architecture reference models

10.3.2.1. DoDAF (U.S. Department of Defense Architecture Framework)

This architecture reference model provides directives and rules for the representation, understanding and development of architectures within the US Department of Defense (DoD), including within cross-Service or even multinational frameworks, by providing external stakeholders with the manner in which the DoD develops its architectures. It falls within the transformation process and takes into account the technological impacts and network-focused operations concepts. It introduces the federate architecture concepts which enable the implementation of capability increments.

The architectures are described according to four views, each of which is broken down into products and data: the operational view, the systems and services view, the technical standards view and the “global” view. The latter provides the context, the area and temporal application authoritative account as well as elements of strategy, doctrine, tactic and usage procedure, operation concepts, scenario and environmental conditions. Of course, explicit connections are defined between some products of the various views, as well as between data models.

The levels of detail depend on the profile of the person operating the architecture (user, designer, developer) and can vary along with possible incremental iterations.

The entire reference model, with detailed descriptions of each step and associated UML metamodels, is recorded in a three-volume guide (*General Governance Framework; Details of the Reference Model and Products; Architecture data Management Strategy*) and guarantees, for the time being, an ascending compatibility with previous versions.

10.3.2.2. Zachman Framework Enterprise

This reference model, called enterprise architecture, generalizes the architecture offered by Zachman in 1987, initially aimed at information systems. It offers a general organization of various descriptions of a company (considered to be a set of physical and human resources and an organization, fulfilling a defined goal) by expressing the relationships between various architectural elements. It is often presented in a 6x6 matrix format, composed of thirty-six cells, with each cell corresponding to a specific view.

The columns represent the aspects: data (what), functional (how), network (where), human and organization (who), schedule (when) and strategy (why). The rows represent the views: contextual, conceptual (business), logical (system view), detailed technological (physical view) and product. Those can be seen as the points of view of various stakeholders, respectively: planner, owner, designer, builder, subcontractor and user.

Here is the row-by-row breakdown of the matrix:

- Contextual view:
 - list of key data;
 - list of key processes;
 - list of key locations;
 - list of key organizational units;
 - list of key events and cycles;
 - list of key strategies and goals.
- Conceptual view:
 - semantic business entity-relationship models;
 - business process and input/output resources models;
 - business logistics models;
 - product flow and services between organizational units models;
 - events and activity cycles schedule models;
 - strategic maps and business strategy (by goal) models.
- System view:
 - logical data models;
 - functional application architectures;
 - distributed system architectures (nodes and liaison characteristics);
 - human-provision interface architectures;
 - processes structure;
 - business rules models.
- Technological view:
 - physical data models;

- physical design of systems (functions, input, output);
 - technological architectures (hardware, software, liaison specifications);
 - data, products and services presentation interfaces;
 - control structures (temporal execution, cycles scheduling);
 - business rules design (conditions, actions).
- Detailed view:
- entity-relationship data definition;
 - processes and control blocks specification;
 - network architecture (protocols, etc.);
 - security architecture (rights and access management);
 - events and cycles scheduling specification;
 - rules specification.
- Product view:
- data;
 - functions;
 - networks;
 - human resources;
 - schedule;
 - incentives.

This general framework takes into account all the material, human and organizational aspects, from technical, strategic, schedule angles, etc. Depending on the views, it therefore groups together the various models which can be used by the sub-processes of the system engineering process, as described, for example, by ISO/IEC 15288. This is why it can be directly applied to capability engineering and systems of systems.

10.3.3. Openness: an essential criterion

A critical parameter of the architectures is their “openness”. Indeed, an architecture locked by proprietary clauses that make it impossible to reuse or modify is an obstacle to the design of systems of systems. Openness translates into the interaction between the various components in order to satisfy the requirements

established (including the interface requirements), entirely defined, public and maintained by agreement by a group of people. It heightens control over the architectures and their evolutions and reduces the total carrying cost thanks to a better obsolescence management, which facilitates evolution and increases reuse. It increases interchangeability capabilities of hardware or software components without modifying the interconnected components, interoperability capabilities, upgradeability capabilities with regards to the needs and available technologies, reusability capabilities for components as well as sub-systems and systems, and reversibility capabilities in terms of modifications done by a third party different from the initial user. Finally, it is a factor of flexibility, meaning that the global system is able to add or remove components in order to satisfy the evolutions of capability requirements.

On the technical level, service-oriented architectures (SOA) provide characteristics which facilitate this openness. They come under a paradigm of organization and use of resources which can be under the control of various proprietary areas: the key principle is to have a set of services – that is to say mechanisms that give access to resources via recommended interfaces and in accordance with access constraints and policies specified by the description of the service – that can be accessed on a network and communicate among themselves. There are three defined categories: services, service providers and service consumers. The collection of available services is managed by a service directory that has no knowledge of the service providers or consumers and is accessible to everyone via the network. Service providers have access to this service directory and can store the definition of the services they offer under as neutral a representation as possible (location of the access point which invokes the service, service parameters, quality of service, etc.), which forms what we call the description metadata of the service specification. The significance of the latter is to later be accessible through automatic search tools. Indeed, service consumers can also access this directory to find the service which corresponds to their need and then invoke the service. The directory is thus a mediator or third party and contributes to the flexibility and security of the collection, insofar as it uncouples and hides service providers and service consumers. The added value of these service-oriented architectures is to offer – for a set of functionalities non-essential to the mission – versatile, reusable and validated solutions, which is all the more useful when systems coming from various sources and possibly from different providers have to be integrated, and reduces acquisition costs.

The openness is not decreed all of a sudden during the life cycle; it is built during the entire life cycle and according to the operational and technical evolutions (technologies and standards). Especially in an incremental approach of capability need satisfaction, anticipating evolutions – for instance in the interoperability area – is essential. Modularity must be the guide in architecture design (search for blocks

with high internal aggregation and low external coupling, hence between blocks), and it must be based on the dependencies between critical hardware and software and on the existence of reusable functions and services. Critical interfaces can then be deduced. Coming back to architecture, and in particular to its technical section, *de facto* interface standards must be favored over *de jure* standards since they improve durability in terms of supply sources that are credible and potentially accessible during the entire life cycle.

The key role of the architecture phase can thus be seen. But the complexity of systems of systems also highlights the necessary use of tools to control these different phases and to interact with various design and implementation teams.

10.4. Requirement traceability and engineering tools

Requirement engineering is a key activity in the acquisition process: if neglected, the client's needs may not be understood by the provider or may only be understood after delivery, which increases development costs and delays and decreases quality, potentially causing rejection from the end users.

Let us remember the various activities associated with good requirement control, as emphasized for instance by the CMMI-type maturity processes (capability model maturity integration): to develop customer requirements (gather the stakeholders' needs, organize them into a hierarchy); to develop product requirements (establish the requirements for the product and its components, allocate the requirements between the product and the components, identify the interface requirements); to analyze and validate the requirements (establish operational concepts and scenarios, establish a division into functions, define the balance between requirements, define the explicit requirement validation methods); to manage the requirements (ensure the correct understanding of the requirements and obtain a commitment on the requirements from the client, manage the requirement changes, establish bidirectional traceability of the requirements, identify inconsistencies between the effort put into the project and the requirements).

When it comes to formulating requirements, here are the classic traps which must be avoided:

- requirements written in terms of technical solutions or including implementation means, since they are immediately put into question by any technological evolution and only the abstraction in relation to a solution valid at a given moment can provide a system with durability and evolution capability, which is all the more true when the system becomes increasingly complex;

- poorly structured requirements (from poor grouping which can potentially lead to erroneous architectural choices and the risk of non-formulated implicit requirements), ambiguous requirements (risk of diverging interpretations between the contracting manager and the project manager), inconsistent requirements (contradictions between requirements);

- requirements that cannot be validated, for which there is no acceptable existing validating procedure.

Requirement control is compulsory for good acquisition cost and time control and use of a system, whether software, software-intensive or made up of hardware components. That is what is emphasized in the engineering normative corpus (ISO/IEC 12207 Information Technology – Software life cycle processes, ISO/IEC 15288 Systems Engineering – Systems life cycle processes) and illustrated by decades' worth of experience in exceeding costs and times because of a lack of sufficient respect for this critical step.

We must insist on the fact that requirement control does not mean that those requirements are unchangeable; such a goal would be futile in the case of the type of systems we are interested in, since the evolution of environments and also requirements is an integral part of the capability acquisition process. It is not the evolution of requirements that poses a problem but the lack of anticipation and potential risk management. A lack of control of the options and choices during the forecasting thinking process would be critical and could lead to a more or less permanent incapability to face the operational situations likely to arise.

Traceability expresses the degree of relationship between two or more products of the development process, in particular products with predecessor-successor or master-subordinate relationships (see IEEE 610.12-1990). Experience has shown that the capability of tracing requirements throughout the specification, architecture, design, implementation and test phases is an important factor in order to guarantee quality. This ability to trace relationships and analyze the impact in the event of change is essential in software or critical system engineering. Of course, the same applies to systems of systems which integrate, among others, such systems and tackle the same issues, potentially increased tenfold.

Let us look at the different types of relationships which can be found: first of all, there is the change of a customer-need into a requirement, whether functional or not, linked to a product or a service. The requirements are then linked to use cases, which are in turn linked to test cases. Let us look at this traceability's first intuitive management approaches in detail.

The goal in developing any system, regardless of its importance, is to satisfy a set of needs expressed by users and/or customers (the latter being, by definition, the representative of a group of users, and it is with them that the customer-provider contractual relationship is established). Usually, the first difficulty is to transform the expression of the needs – or, often, the expectations – which are not always explicit, into a set of requirements that are formalized using a precise language. Minimum traceability goes through a “traceability matrix” which links each need to the various requirements that make up its formalization. These relationships allow us to see which specific needs should be reconsidered in the event of a requirement change during later development stages.

Just as important as the definition of requirements is the definition of use cases, also called use scenarios, which offer a perspective of the user’s point of view on the proposed system implementation and group the users’ needs, including those for other systems with which we are likely to interoperate. It should be noted that the link between requirements and use cases is done via the architecture: the latter can indeed be seen as the description in intention which organizes all the particular cases described in extension by the use cases, and organizes them into a cohesive form which generalizes them. Once again, we can see the usefulness of a matrix which matches a requirement with the various use cases that facilitate its evaluation. Its analysis will allow us to control the physical implementation knowledge, to search for, *a priori*, potential defects and increase the level of operation security. We will get back to this in a future section.

The link with reality is compulsory and is materialized in the traceability between requirements and tests, which can also be recorded by a matrix. This also facilitates the link between the use scenarios – logical linking of functions and actions allowing us to describe the expected system mission in a given state and context, which consists of imagining the development of an action to validate the delivery of a product or service implemented with a defined usage doctrine – and the test scenarios. This link is all the more complex since a use scenario can present several tests, and a test can be found in several use scenarios.

The previous steps seem simple enough: the various links are formalized by matrices, and traceability is a search for paths within the various matrices thus defined. However, the combinatorial analysis quickly becomes very important and inaccessible to non-specialized tools. The two main traceability functions are, on the one hand, the visualization of the traceability relationships with the ability of possibly zooming in on details and lower-level dependencies, and on the other hand, the automatic operation of these relationships to see the impact which a modification on one level might have on other levels and to facilitate management throughout

configurations. Such specialized tools already exist² and are used by companies such as Telelogic (data input and visualization are done via Tau, requirement management via Doors and Trek Toolbox), IBM Rational (Rose for visualization, Requisite Pro for requirement management), Borland (Together on the one hand, CaliberRM Datamart on the other), etc. Thanks to the use of such tools, it is possible to generate structured, complete and non-ambiguous specifications – or at least to greatly improve their quality – with a view to being systematic, which facilitates communication between the actors of the project (contracting teams as well as project teams) and guarantees a better balance between requirements and goals, and products and requirements. This is nothing more than what is required by standards such as IEEE Std 830-1998 “IEEE Recommended Practice for Software Requirements Specifications” and IEEE Std 1233-1998 “IEEE Guide for Developing System Requirements Specifications”, which are an integral part of the software engineering and system engineering authoritative accounts and can thus be applied to systems of systems or at least to their constitutive elements.

The equipment of the requirement engineering approach goes further than that since it doesn't stop at traceability. It requires, *a priori*, the two main functionalities, on the one hand traceability and on the other critical analysis of the traced data and the models to which they are attached. Moreover, this must be done on the three respective levels of requirements, architecture, and technical and financial characteristics. The following functions are thus required:

- on the one hand, identification and inventory of the requirements applicable to a system of system and repercussion on the constitutive systems and those of a lower level (down to the strictly necessary level); on the other, analysis of the requirements, meaning balance between need, consistency, completeness, traceability and financial relevance;
- on the one hand, definition and representation of the architecture of a system of systems according to organic points of view (constitutive systems and their main components), functional points of view (functions and flows) and dynamic points of view (use cases, usage scenarios); on the other, impact analysis of a need, requirement or architecture modification on the system and representation on the different views;
- on the one hand, technical risk management aid, operation security aid and cost estimation aid, via the recourse to appropriate models and the traceability of these models with regards to these aspects; on the other, evaluation of technical or financial characteristics according to usage scenarios.

2. In June 2007, the American company IBM launched a friendly public bid for the Swedish company Telelogic. After the opening of a study in October 2007, the European Commission finally allowed the buyout at the beginning of 2008.

On the level of traceability, an exhaustive inventory of the requirements is thus necessary, with links toward use scenarios as well as verification and validation criteria, all of this having to be performed throughout the system engineering process, with the traceability of all of the architectural choices and associated configuration management. Moreover, documentation traceability for all contract documents and the documents provided for various reviews must also be ensured. However, to allow for the evaluation and analysis stage which offers the real added value to requirement engineering, the tool must record the links between what precedes and the various associated models, whether technical, behavioral, architectural, financial, etc. This seems simple enough but actually requires special attention, as well as a view to model all the functional and non-functional components likely to play a role in the architectural and managerial framework of the system of systems. The difficulty lies in the project team's ability in using such a tool, more than in the development of the tool itself.

10.5. Reverse engineering and impact studies

Let us go back to one of the specificities of systems of systems engineering, linked to the necessity of reusing what already exists to save on costs and time.

Within the engineering process of a “simple” system, the approach is a fundamentally descending approach. The need is translated into system requirements, from which an architecture is deduced, architecture which is refined through design operations until the system is defined. Test and integration phases follow by means of successive constructions and consolidations.

It would be ideal to be able to do the same with a system of systems, but the scope of the existing systems is excessively broad. It complicates any attempt at standardizing the capability need in time *in abstracto*, that is to say without taking into account whether this need may be fulfilled by what already exists, and what can be done within compatible time limits, in terms of schedule and budget. For example, we cannot conceive a ground aviation force without tanks, helicopters and artillery. Likewise, a general cost accounting system at national level cannot be designed without existing accounting tools designed for very different rules and interfaced with various production management tools. Of course, an ascending approach is followed at least part of the way, starting from what already exists and checking how an “intelligent” organization with punctual updates of some components can enable the change from the initial situation to the desired target situation while following a series of intermediate points contributing to partial satisfaction of the capability goals.

Coupled reverse engineering and re-engineering approaches can appear interesting on this level, starting from the existing product or services, and going back to the specifications to see, on the appropriate abstraction level, how to build the trajectory in the specification space and, by model transformation operations, see how it translates on the level of necessary component evolutions.

The difficulty inherent to the development of this approach lies in the existing analysis work, for which it is necessary to have a good command of the technical concepts used, the direct characteristics standardization as well as their alignment and their reformulation. All this work requires harmonization and coherence of the terms and concepts used, which don't always represent the same thing for the various parties. Beyond the availability and validation of the models, it is essential to define the metamodeling layer, which will authorize the exchange, comparison and generation of new models, whose instantiation will then provide evolution leads.

It must be acknowledged that if this approach is starting to be mastered in the case of software systems and in applicative areas where a subjective discipline reigns in terms of development rules (since reverse engineering of software filled with tricks and loops aimed at technical performance to the detriment of functional readability is relatively ineffective, dedicating efforts whose cost then becomes comparable to the redesigning and redevelopment of the whole), it is not yet on the agenda for systems of systems such as those found for instance in the defense field. On the other hand, the approach is used on some critical sub-systems, such as on-board avionics of some fighter planes, which manages the on-board "intelligence" of the plane: management of sensors to detect threats, building of plots and association into tracks, multi-track management, weapons systems management to assist the pilot in his decision and the execution of his armed response. It was thus that Lockheed Martin applied reverse engineering to the on-board software of the F16 combat system in order to re-design it into a modular combat system compatible between platforms (since various fighter planes do not necessarily have the same hardware and electronic architectures – data buses, their number and how they interconnect, in particular – the usual approach made it compulsory *a priori* to redo the on-board avionics for each plane configuration). The results announced were very convincing. Similar studies are being carried out in France to implement these reverse engineering and re-engineering concepts in other areas.

The significance of this approach, which alternates modeling and metamodeling levels, becomes obvious in the development of impact studies which must be carried out for instance when a requirement changes, to know whether the physical components to be used or the flows between components need to be modified. Likewise, if a component becomes obsolete and needs to be replaced either on a one-to-one basis or on the level of the function it introduces, its impact on the requirement satisfaction must be known. Model-driven engineering can analyze such

questions if, on the different view levels (system, logical and technical), we possess all the models and all their coherence links within the views, as well as the allocation functions from one view to another. These impact studies are critical for the control of systems of systems, for their design (to see how to potentially upgrade constitutive systems to satisfy a capability increment) as well as their evolution. In the latter case, the management of consistency between the various views of the various systems is a key component of the system of systems configuration management control.

Let us remember that impact studies are important for the architecture and development design phase, but they are also essential during the validation and integration phases: indeed, the validation of elementary functions goes through unitary tests based, for example, on trials (we will come back to this link between validation and tests in another section), and any functional impact or impact on a physical component creates an immediate impact on the validation tests and, subsequently, on later integration levels and associated tests. Associated costs must not be forgotten either. The economic significance of controlled model-driven requirement engineering thus becomes obvious, on top of the added value for system of systems engineering.

10.6. Distributed simulation tools for model engineering

Simulation has imposed itself as a key tool for the design, development and qualification of increasingly complex systems, on account of its ability to operate models of all types (functional, technical, analytical or behavioral). However, given the scope of skills mobilized in these activities, it is no longer conceivable to use a single tool to satisfy all the needs. On the other hand, the need for simulations which are interoperable and reusable in contexts as broad as possible has imposed itself, in order to simulate a system or a system of systems throughout its life cycle and with various degrees of modeling.

Insofar as the systems of systems involve several systems (some of which already exist), which can thus be at different stages of their life cycle, it is not always possible, by force of circumstance, to bring them all together while working on the design of the system of systems architecture. Simulation sometimes is the only way to explore the solution space. Even then, it is necessary to be able to jointly use simulations which are potentially located in various geographic places, with different conditions of use and also different operating rights due to industrial property on subjacent models.

This demonstrates the significance of an available distributed simulation infrastructure, a true federate tool to implement simulated or actual means, whether

calculation codes, simulation codes, operational information system codes, hybrid bench codes, piloted simulation codes, etc. Such a simulation infrastructure is in fact a real information system which offers data and information transfer services, potentially on different networks, as well as access to a set of specialized resources which the system links with computer security levels and confidentiality guarantees. Such infrastructure projects are currently being developed by integrating system project managers as well as a certain number of acquisition services in various defense ministries and, thanks to the increasing maturity of the networks and the bandwidths available, they often have international scopes.

In France, for instance, with battle-labs (please refer to Part 1, Chapter 1), a common technical simulation infrastructure has been developed. Organized following a service-oriented architecture, such an infrastructure offers a base of common collaborative services like directories, instant messaging, forums, file transfer, multipoint communication, documentary workflow management, etc. Other services fall under system administration, archiving and data import and export. All these services run in a coherent manner according to interoperability standards:

- IPSec/IPv6, HTTPS (Secured Hypertext Transfer Protocol), SOAP (Single Object Access Protocol) for telecommunication protocols at the transport layer level;
- XML (eXtensible Markup Language) for data exchange;
- SEDRIS (Synthetic Environment Data Representation Interface Specification) for interoperability between models and simulations by standardizing data semantics and format;
- HLA (High Level Architecture) for technical interoperability of the distributed simulations as well as their reuse within simulation federations: this standard defines, among others, a service interface specification which enables exchanges between components of the distributed simulation, as well as a definition of the object model which must be defined for each distributed simulation.

For maximum openness and interoperability, the maximum amount of open formats is used and APIs are provided in order to extend the functionalities. Interfacing with the simulation component infrastructure also relies on international interoperability standards and follows an interface design method, which facilitates configuration development and management. The idea is to create an intermediate abstraction layer between the infrastructure and the component, according to the middleware principle. Thus, an interface is made up on the one side of generic attributes representing interface-type data and on the other of operations (or methods following a philosophy of programming by objects) representing services provided by the interface.

Moreover, services which take events into account are implemented, which allow actions to be started in an asynchronous manner. This is particularly useful when simulations are used, insofar as it is not always possible to systematically synchronize everything, since event management sometimes is the best way to take into account some non-determinism aspects on the level of the models.

The significance of such a federate simulation tool becomes clear in the various phases of the life cycle, when it allows models adapted to these different phases to be implemented and connected, maybe even linked. Far from being a simple technical design or technical qualification tool, it must be used to its full potential and within an exhaustive model-driven engineering view.

10.7. Global control of operational security via testability

Systems of systems architectures bring in a multiplicity of internal and external interfaces created by the integration of systems with one another. On top of it, other factors of complexity are added, such as: the heterogeneity of the logical and physical characteristics and the disparity in the lifetimes of the systems and their components (lifetime ranging from two to five years for function implementation technologies compared to 20 to 30 years for systems); the varying difficulty of having, for the needs of the verification and validation activities, sufficiently detailed documentation for specifications of the sub-system or component bought off the shelves. The operational security and information system testability aspects are thus *a priori* of a functional and structural complexity level superior to that of component system testability.

Let us remember that operational security is characterized by the following attributes: reliability, maintainability, availability, safety and security. These concepts are defined as follows:

- reliability is a product's ability to accomplish a required function, in given conditions, during a given time interval. It also represents the probability that the device will work correctly during a given time interval;
- maintainability represents a product's ability to be put back to a given operating state, within specified time limits, when work is carried out according to prescribed procedures and given conditions. It depends on testability (ability to carry out verification and validation operations on system properties, and troubleshooting) and "repairability" (ability of a system to go back to its proper operating state after swapping the broken components – for software, it is about fixing design mistakes rather than repair them);

- availability represents an entity's ability to provide a required function in given conditions, at a given time, assuming that the necessary external means are provided. It's also the probability of the device working correctly when prompted;

- as for "security", it has two different aspects: safety and security. Safety represents the ability of a system to guarantee the protection of the environment, namely goods and people, when faced with actions from the systems. It corresponds to the absence or to a low probability of events likely to have serious, or even catastrophic, consequences on the system environment, particularly on people. Finally, security is the ability of a system to resist natural, accidental or involuntary external attacks. It is usually guaranteed by protection mechanisms which limit the effects of those attacks. It involves integrity, which is the non-occurrence of modifications brought to the systems or information that are the result of the attacks.

Systems testability essentially depends on two properties: controllability and observability. Controllability is the property of a system which allows its internal and external states to be simulated or generated from the outside: simulation will be used in the case of a system representation model, and generation in the case of real implementation of the study model. Observability is the property of a system which facilitates the measure of internal and external states successively reached by the system. The goal is then the correction of possible non-tolerated disparities. This property allows us to decide, by tests and, if possible, by mathematical demonstration, whether the behavior of the system is correct or not compared to the specifications derived from the requirements expressed in the technical specifications of need.

All these properties greatly depend on the architecture of the system of systems, typically because the operational control and monitoring functions are centralized or, on the contrary, decentralized, and even distributed among the constitutive systems. Of course, they also depend on the network through which the information which characterizes the operation state is spread. They are a determining factor in obtaining required coverage ratio of internal events (detection, location and diagnosis of breakdowns) or external events (detection, location and diagnosis of evolutions of threats to the environment) within time limits imposed by constraints of cost and operational availability. This gives the system the property of being reliable and thus accomplishing its mission and even maintaining it, in the event of internal errors (such as equipment or component failure or breakdown, or attacks) or of evolutions of its environment's behavior. This property can be achieved through automatic reconfiguration and restoration of functionalities, depending on the damaged operation modes tolerated.

Reaching these capabilities requires a closed loop between the design loop and the usage feedback, in the three fields of engineering, operation and maintenance.

Concerning engineering, the important steps are the identification of needs, definition of the testability requirements, insertion of matching devices in the architectures and the components, and the verification of testability. For the operational stage, the points that must not be neglected are the monitoring of the system states, the capability of diagnosis, in real or delayed time, of the situation as well as the system and component states, the location of the errors and the reconfiguration decision. Finally, for maintenance, the critical points are the lists of measures stored in the system, the diagnosis of the errors location, the corrections or repairs or preventive swaps, and the operators' training on various event simulators in order to predict reconfigurations in the event of resource damage.

The verification and validation processes – activities which occur in all stages of the systems' life cycle: design, development, use and maintenance of operational condition (particularly preventive and progressive maintenance) – usually call for test methods. These methods essentially consist of defining and supplying a set of system input data and estimating whether the output data is in conformity with the functional and behavioral characteristics required. The input data and expected behaviors are recorded in test scenarios, elaborated according to the test goals derived from the qualification requirements. Nevertheless, given the increasing complexity of the functions to implement and the severity of the security requirements, the verification and validation processes call more and more for formal or semi-formal methods: model checking and proofs of theorems. These methods complement the classic test methods which do not allow the exhaustive demonstration, according to every scenario available, of the correction of all system properties.

Due to the complexity of the functional, organic or physical models, the cost of the test or of the formal proofs plays an increasingly significant part in the global carrying costs. This is particularly true for software, whose role is becoming increasingly preponderant in systems; the test represents about 50% of the software cost, mainly spread among the design and maintenance stages. The problem concerning the testability of systems must thus be essentially considered in both cases: systems which are still in the design/development stage and maintenance of operational condition process (offline tests and formal proofs), and systems in operation (online tests).

Offline tests and formal proofs are determining factors in generating verification and validation costs and times in design and maintenance: detection/location and diagnosis of breakdowns or design errors, and repair or correction, particularly when it comes to software.

Online tests must be considered as critical not only from the point of view of safety, but also from the point of view of operational performance. Indeed, within

the limits of tolerance for errors, systems of systems architecture entails the real-time maintenance (time which is imposed by the system environment) of the interoperability of their functional components, even in the event of breakdowns due to accidental or intentional attacks.

This point becomes significant in the analysis, if the guarantee of a minimum level of operational performance is required. This raises the problem of determining damaged operation modes of systems of systems, as well as the problem, on the one hand, of minimizing the detection/location of failure and diagnosis times or the evolutions of threats to the environment, and on the other hand, of minimizing the real-time reconfiguration times of these systems.

Considering the previous points, the optimization of the global carrying costs of systems of systems goes through the optimization of the verification and validation processes cost. As well as significantly decreasing the development delays and enabling the justified use and reuse of existing sub-systems and on-shelf products, the correct running of these processes allows a prime contractor to delay the expression of some applicative requirements, or to modify them at the end of the validated design work for a system solution, and to significantly reduce the systems' operational failure rate. This can only be guaranteed by a rigorous system engineering, management and quality assurance approach, which uniformly tackles the problems in the design of systems of systems architecture and the choice of methods of verification/validation. The sets of tests or obligations of proofs produced during the design stage, in particular, will have to be reusable during the development stage, given that the tests or formal proofs have to be used for operation or the maintenance of operational condition. Needless to say, the same thing applies for the systems, sub-systems and hardware levels.

This approach requires the possibility of representing the systems of systems architecture with models on various abstraction levels in order to control:

- the expression processes of functional, performance and operation security requirements;
- the validation processes for system solutions, and even for sub-systems at least for the most critical implemented functions offered by the industrial project managers;
- the specification, design, verification and validation processes for system, sub-system and hardware solutions.

The first two sets of processes and the corresponding models fall under the control of the general contracting teams, whereas the latter fall under the control of the industrial project managers, manufacturers and subcontractors.

During the feasibility stage, the development of conceptual models or metamodels of systems of systems architecture allows us, within an iterative process, to establish an agreement in the definition and validation of the operation security requirements. These high abstraction-level models are to be derived from the requirements and constraints expressed in the specifications, the technological and costs databases built on previous or current experience, and the technological forecasting data. On this level, it is about evaluating the various conceptual architecture orientations according to costs, duration of life cycle processes, availability and survivability criteria. To do so, it is advisable to review different hypotheses related to the operational usages, policies and organizations of the systems' logistic support. In particular, hypotheses must be made about the allocation of architectural elements on the technical levels of intervention: in the French military, the terminology used is NTI1, NTI2 and NTI3 (namely, respectively: on the field of operation, back-office on the theater of war, in factories).

From the validated technical requirements, and based on the architecture orientations previously recorded, reference architecture models of a lower abstraction level have to be generated, derived from previous metamodels destined to be used by the comparative evaluations of proposed systems of systems architecture solutions. These evaluations must be carried out by measuring, according to the requirements or constraints stated above, the impact of testability on the development and maintenance of operational condition costs and times, and on operation security. Simulation and formal proof or reasoning processes then become extremely useful.

On this level, the survivability requirement must be taken into account, which leads to setting the redundancy levels of related functions to limits below which the system of systems cannot continue its mission or risks dangerously decreasing its operational efficiency:

- failure tolerance limits, defined according to criteria or constraints of costs and reliability on the level of the system of systems and constitutive systems;
- configuration adaptability limits, defined according to the requirements of missions which have to be planned again depending on the evolution of the tactical situation, on very short notice (from less than a hour to a few hours), or of the strategic situation, on slightly longer notice (depending on the case, from less than a day to a few days).

As for the controllability and observability requirements, they depend on the choices made at the architectural level:

- definition and allocation of control, measure and observation functions for the systems of systems operational states, and observation and diagnosis functions for the systems of systems' environment behavior;
- definition of the distribution protocols, within the constitutive systems and various interfaces, of the messages characterizing the systems of systems operational states (particularly damaged states) and the environment behavior;
- definition of the protocols which establish the agreements, between consecutive systems and systems of systems, based on the results of diagnoses and measures;
- definition of the transmission protocols of potential reconfiguration orders.

As for the maintainability requirement, it is necessary to determine the optimum criteria of division of the systems into components which can be replaced online and in workshops, as well as their allocation to various intervention levels, depending on the policy and organization of logistic support in the use stage.

These models facilitate the evaluation of solutions according to testability criteria compared to the expressed requirements, with any variation on the testability level potentially liable to impact the costs, delays and performance. This involves the guarantee of traceability between requirement representation models and system architecture solution representation models. This traceability must also be able to face the evolutions of technologies or operational needs, the control of the processes of development, production or maintenance of operational condition of the systems of systems (particularly the maintenance of constitutive system interoperability which conditions the minimum maintenance of the systems of systems' operational performance). Being able to include the testability process within the process of configuration management then becomes essential.

The testability criteria feature the following:

- the coverage ratios and the delays in the detection/location of internal events or failures within the systems of systems, or of environment events or state change. These ratios must be linked with the acceptability thresholds of the risk levels (product of the severity of consequences and the probability of occurrence of the dreaded event) with regard to operational requirements. In particular, the criticality of the functions involved which might compromise the continuation of a mission, for instance after the non-detection or the faulty location of a breakdown, must be evaluated;
- the ratios of detection, location or diagnosis errors for systems of systems internal or external events, or the ratios of false alarms (detecting events when there was no breakdown or no new real threat to the environment), ratios which must also

be incorporated into the thresholds of risk acceptability (diagnosis errors with an impact on operational availability);

- the diagnosis delays linked to determining the causes of systems failures or the interpretation and identification of the operational environment events and threats;

- the nominal delays for the restoration of the proper operational state: for preventive or corrective maintenance, this falls under delays corresponding to the swapping of broken down functional components and verification/validation procedures;

- the progressive maintenance delays, namely the delays implemented to face evolutions of operational and functional needs or considerations of new technologies;

- the automatic reconfiguration delays during the development of missions, which must be compatible with the constraints on error tolerance or the operational environment constraints;

- the duration of the design and development stages (often called time-to-market), which notably depends on the duration of the verification and validation processes;

- the maximum costs which must not be exceeded for the design and application of verification and validation tests sets and formal proofs: they are essentially linked to the complexity of the architectures, of the functions and of their software implementations.

10.8. Towards a virtuous circle of simulation-tests to control the tests

10.8.1. Integrated simulation-tests approach at the service of model-driven engineering

The model-driven engineering approach brings, as seen in the previous paragraphs, many advantages during various stages of the life cycle, but *in fine* it is built on a balance between these models and reality. Beyond the intensive use of simulation models and techniques, it is thus essential to make the link with real data and behaviors observed during real usage of the systems.

The integrated tests-simulation approach must be applied as early as possible (as early as the preparation stage, before the system design). Whether it concerns the identification and representation of the threat, the general operational usage concepts or the definition of the operational need requirements, it can be very useful to have various types of simulation. The same thing applies to clarifying the functional requirements and technical specifications during the system design. On this level,

risk mitigation can easily be carried out by adequately using simulations coupled with functional tests. The first bricks of the virtual prototype then become available, built on the traceability between requirements and specifications, allowing coherence to be established between the logical, functional and physical views of the system.

This coherence, as well as its justification via the association of simulations and tests, is vital and at the heart of the impact analysis: in the event where an element of one of the views is modified (for example a requirement, function or component), it is possible (ideally easy) to evaluate the resulting implications on the level of the system, and particularly the re-validations (via tests or simulations) necessary to guarantee non-regression in terms of system performance. The tests/simulation complementarity provides an answer for this issue: if the initial test managed to correctly reset the simulation in explicit and outlined validity conditions, the non-regression test can be carried out in simulation with an estimable credibility. The economic impact and time saved are immediate and justify the interest in carrying out simulations and tests together within a system engineering process.

This joint use of simulation and tests allows us to go from the “tests, correction, test” approach to an iterative and incremental “modeling, simulation, correction, test” approach. The latter facilitates correction during each stage of the life cycle as well as the updating of the model if necessary: correction is done on the level of the model or virtual prototype, which immediately saves time and money. This model will later be used to predict and extrapolate the carrying costs, to evaluate the technical and operational performance, the system availability, etc.

This interdependency between tests and simulation facilitates reuse throughout the life of the system, with credible representations updating, which enables successive risk removals. With the control of reuse and validation processes, it then becomes possible to have one powerful tool for the evaluation of a system of systems (including to the limits), as well as the exploration of new concepts and performance predictions, before starting the manufacturing process.

A few warnings are necessary to counter generally accepted ideas:

- “the use of simulation as support for a test is common practice”: that is true, but what is discussed here goes beyond this simple observation, by emphasizing a necessary feedback and mutual contribution;
- “simulation will replace tests”: this is not the goal here and it is a completely unrealistic one. What we aim for is improve the physical tests necessary for the production of critical evaluation data;
- “simulation only allows us to reason about statistics and thus cannot be truly trusted”: but it is a different problem when we carry out a series of tests to achieve

global performance at a global level, only taking into account a few specific operation points, and we don't always know how to choose them to guarantee a change from local to global! Simulation, in its interaction with tests, does not aim for perfect results. The coupling of the two techniques must help us establish a consolidated global credibility of the system to develop.

On the one hand, the tests provide data from the real world, in given situations and environments, which are *a priori* credible, and help evaluate the achieved technical performances and level of system maturity. During these tests, security and environment protection constraints are important limitation factors which must be taken into account.

On the other hand, simulation can be costly, particularly when we must guarantee the validity of the models developed. It allows us to predict experimentation results, by exploring the field of possible solutions in terms of performance in the area accessible to experimentation, and extrapolating the performance outside the area potentially accessible to tests.

Moreover, some tests (for instance system interoperability tests) are easier to develop by simulation, and much cheaper. Indeed, it is not necessary to focus or even move all systems to a single place: this geographical distribution capability of the simulation is a clear advantage in terms of reactivity, time and costs. This is how various interoperability levels can be evaluated more easily (such as those defined on NATO's level), by mobilizing fewer resources. In the same way, complex tests (such as those required by systems of systems) are now possible, while they didn't use to be, for not all testing installations are mobile.

It is therefore wise to use tests and simulation in a cooperative way via tight coupling within virtual and digital synthetic environments, to bring together virtual prototypes and real material for a better analysis of the systems of systems throughout their life cycle.

Moreover, it is essential to simultaneously promote the reuse of validated models, because of the multiplier effect in terms of risk and cost control. The credibility of information coming from modeling and simulation activities and of the limitations thus identified is essential. It requires the definition and execution of a process called VV&A (verification, validation, and accreditation) and the collection of particular test data whose purpose will be to validate these models. Indeed, the use of uncertain data and models may introduce additional risks during the acquisition program. The subjacent objection is often put forward to compromise the use of simulation. Nevertheless, the argument can easily be turned around: a piece of data poorly referenced is a risk factor just as harmful insofar as it foresees

performance elements in an erroneous situation or environment. Mutual tests/simulation reliability is thus fundamental.

10.8.2. *VV&A and VV&C*

The definitions given in this section are taken from the *Modeling and Simulation (M&S) Master Plan*, written by the American Department of Defense in October 1995, and used again in the NATO M&S Master Plan AC/323(SGMS)D/2, published in August 1998, and widely adopted by the international simulation community.

An efficient evaluation strategy must include model and simulation verification and validation throughout the systems' maturity in order to establish credible information. *Verification* is the process which determines whether a model or simulation represents the conceptual description and development specification in a precise and faithful manner. This process also includes verification of the software development techniques used. Basically, it is about demonstrating that what has been done corresponds to what was asked and that it was well done. *Validation* is the process which determines the degree of balance between the model or simulation and the real world with regards to intended uses of the model or simulation. In other words, it is about evaluating the manner in which the initial problem was answered.

Based on an adequate evaluation strategy, the model or simulation is determined as acceptable with regard to a given use in a particular application framework; this represents the accreditation phase. This being said, the data used by the models and simulations must also be certified via a process called VV&C (verification, validation and certification). Here are a few more definitions. *Data verification*, from the data producer's point of view, consists of implementing techniques and procedures which guarantee that the data satisfies the constraints defined by the data standards and usual business rules. Data verification, from the data user's point of view, consists of implementing techniques and procedures which guarantee that data standards and usual business rules are correctly formatted. *Data validation* consists of the evaluation documented by experts in the field, and the comparison with common reference values. From the data producer's point of view, this evaluation is carried out in connection with explicit criteria and hypotheses. From the user's point of view, the evaluation is carried out in relation to the balance with the use perspective within a particular model.

The VV&C process thus verifies internal consistency and data correction, validates the fact that they represent entities from the real world in accordance with the intended use, and certifies the fact that the specified quality level of the data

corresponds to the intended use. Just as before, the process has two perspectives: that of the producer and the user.

A related problem lies in the fact that the data can come in two forms. It can be raw – from tests, literature (open or technical), information, etc. – and in that case, part of its credibility is established by the study of collection processes and intrinsic credibility of the sources used. They can also be aggregated, meaning that they are generated by applying various treatments to the raw data; the balance between this data and the models used in relation with the intended use must then be examined in detail.

What must be remembered is that the data coming from laboratory or field tests will be integrated to validate the models and simulations during the maturing of the system, which leads to a complete set of models and simulations which gains in faithfulness (the verification aspect) and in credibility (the validation aspect). It is essential to point out that any validation is done in comparison with an intended use; this has been underlined from the beginning in the various authoritative accounts of simulation but it is incidentally just as fundamental for test data or for a test. This shows the necessity of documenting all the test and decision conditions which led to choosing to control this or that degree of freedom, on the level of the scenario or the environment.

This last remark is an important one in the sense that the VV&A and VV&C processes are often described as costly and yet as a “necessary evil” that generates initial additional cost since the economic profitability only appears later through reuse. In fact, the same argument is valid *stricto sensu* for the tests and test data, which is, however, rarely recorded, all the more so because the last decades and their important budgets almost allowed all those tests to be done on demand: today we find ourselves with a plethora of test data that is completely useless.

To give an idea of the costs inferred, we can quote the *Prefeasibility Study on Simulation Based Design and Virtual Prototyping* report, published by the NATO Industrial Advisory Group in September 2000, reference NIAG-D(2000)9 AC/141(NG-6)D/25: the cost of VV&A is estimated to be in the region of 15% of the global simulation cost, and of 2 to 6% for reused simulation.

To come back to verification and validation, it is difficult to evaluate *a priori* the necessary validation level even if we can define the effort necessary to reach given credibility levels. This is due to the fact that validation is eminently done in connection with the intended use of the product (data, model or simulation) and this cannot necessarily be precisely defined before the life cycle. Of course, this difficulty must not be interpreted as a crippling obstacle but as the necessity to develop the VV&A and VV&C processes within an iterative and incremental

approach throughout the systems' life cycle. For systems of systems, we are thus faced with a double difficulty: validation on the global level relies on new usage concepts (this innovation aspect in delivering a certain effect is one of the *raisons d'être* of the systems of systems!) and potentially innovative architectures, and on validations of the constitutive systems. Furthermore, these validations have been carried out in a context that is not necessarily that envisaged after integration within the system of systems. This mutual interaction loop between the global and individual levels is a true challenge, but its command leads to the qualification of the system of systems.

10.9. Collaborative work tools

10.9.1. *New technologies, new work methods, virtual teams*

The work and business world has widely taken technological evolutions into account, in particular to develop means to create value more quickly by parallelizing tasks as much as possible. This is done through increased interactions between individuals (prescribers, members of the design and development teams, users) beyond their role, their geographical situation and their culture. From the technological point of view, the current data, voice and video transport capabilities on the network allow us to always exchange more between distant points, from one continent to another, even through instant messaging.

It then becomes possible, on the one hand, to exploit talents from people who are physically very far away, and on the other hand, to have an immediate return without having to book appointments first. Multilocalized companies thus have teams spread around various work places, which allows them to integrate particular skills throughout industrial acquisitions, but also to take into account the users' various cultures, which is an essential asset to facilitate subsequent marketing: let us mention, as fine examples of these new work modes, the automobile industry (Toyota, BMW, etc.) and civil aviation (Airbus in Europe, Boeing and its subsidiaries in the United States and Russia), but also the multimedia (artistic creations, in particular) and health sectors (telediagnosis, advice on surgical procedures and even remote intervention).

In a system of systems, each constitutive system, which is specified, designed, developed, implemented and maintained independently from the others, initially corresponds to a specific project involving distinct stakeholders. Since the systems are no longer considered individually but in relation to all the other systems with which they must collaborate, it is necessary to take into account multiple and progressive interfaces as early as the expression of need and all the way to withdrawal, both from the technical and organizational points of view.

This reasoning translates into the evolution of the acquisition methods: it is increasingly necessary to manage distinct projects, entrusted to various teams and project managers, each with their own life cycle, in a coherent and concerted manner. Moreover, these systems must be integrated within vaster sets, whose definition may change. Different software tools must allow the project teams to specify and define, and later develop, implement and maintain the systems they are in charge of while taking interactions with other systems into account. In practice, these tools can only be used if the various teams involved can communicate with their interlocutors in real-time and take into account the evolutions of other systems, and if the teams in charge of the systems of systems are able to pronounce the necessary arbitrations in case of conflicts.

The increasing number of information flows resulting from this can only be taken into account within a broad company framework, by making all of the contracting teams and project teams agents collaborate tightly during every stage of the systems' life cycle. Hence the necessity of implementing software tools which facilitate communication between the agents and the work in virtual planes, as demonstrated for instance by the battle-labs. Let us remember that a battle-lab is a federate set of hardware, software and human means permanently or semi-permanently available and dedicated, within a capability approach, to a system or a system of systems, which can be used during all its life cycle in order to study the usage and doctrines concepts as well as the technical solution performance or users' training.

10.9.2. Collaborative work environments for systems of systems engineering

In a traditional open-plan organization, as can be observed widely in the industrial world, the various agents of a project are always gathered in a single location, which facilitates group work and information sharing, and thus improves productivity as well as decision-making. This organization model finds its limits when it is no longer possible, because of the diversity of participants, to gather them in a single location, or when several distinct projects have to be coordinated. It is then necessary to build up virtual planes, the subjacent principle being that everything that can be obtained by physically gathering the teams on one single plane must be obtained without moving them and by using a collaborative work tool.

Particularly useful for systems of systems, the calling of such collaborative work tool is the instrumentation of planes which take care of them in connection with conventional project planes. Aimed at all participants of the teams involved (contracting teams, users representatives, even project teams), it must take into account all the data and information the latter are likely to manipulate. It must thus facilitate the following main functionalities:

- sharing or communication of information of any nature (project data, technical data, electronic messages, discussion forums, simulations and tests results, feedback, etc.) stored digitally, according to asynchronous (possibility of later access and delayed response) or synchronous modes (access reserved to the community of connected agents with the possibility of real-time response);
- connection of distant sites in real-time using the available infrastructures;
- virtual meetings (transmission of image and voice, whiteboard with virtual post-its, report during the meeting, instant messaging, e-mails, forums, etc.);
- work on documents in groups with validation, distribution, configuration management, traceability of modifications, search function and archiving;
- implementation on a site of specific tools (budgetary follow-up, system engineering process follow-up, definition and management of the architecture views, functional or technical simulations), and ability to see the results, on other connected, or functionally distributed, sites, of these specific tools according to the access rights and responsibilities of the members of the distributed team;
- ergonomic visualization of multidimensional data (graphics tablet, multiscreen show, virtual reality centers, etc.);
- access management via directories, according to the level of detail or sensitivity of the information and users' rights, storage security and data exchange.

The functionalities previously described are offered to the members of a work group to whom we wish to give the possibility to work on a virtual plane, and are thus accessible from the members' usual individual workstation or from dedicated workstations, for example in meeting rooms. They are built on a set of technologies which have been developing rapidly during the last few years: transport and multicast routing techniques, which consist of transmitting information to one or several recipients without sending the same packets several times on the same network link (open possibility in IPv4 and IPv6), asynchronous or instant messaging services, groupware products bringing in services of multimedia documents sharing on private virtual networks whose access is controlled, or which give the user access to data spheres and ready-to-operate applications depending on the user's profile.

The current explosion of the Web and the creation of many virtual communities based on access services and services of exploitation of multimedia data which are constantly updated are all sources of inspiration for the more or less integrated collaborative work products. Multimedia conferences via the Web with several simultaneous connection points, peer-to-peer text data and video sharing (like the world-famous YouTube), interactive sharing of applications and executions on demand, etc. All these current realities were, only five to seven years ago, the object of research projects financed by the European Community (4th and 5th Framework

Program for Research and Development, thematic Information Science Technologies). Editors of groupware nowadays provide us with integrated mail and instant messaging services as well as portals, which can be configured according to the user profile of the person who logs in, and filter access to data, applications, potentially interesting information, etc.

We even like to dream about having project teams sharable over the Web, where the entire user community participates, depending on their skills, to challenges launched on the Web, in the image of Goldcorp Inc – who, in 2000, launched a challenge on their website by uploading their geological data and rewarding people who found interesting sites for gold prospecting – or Proctor & Gamble who outsource part of their research and development through Internet surfer communities via the InnoCentive virtual network, which is only one of the e-marketplaces facilitating the exchange of innovation (we can also mention YourEncore which recruits retired scientists, NineSigma, etc.). We can of course wonder about the intellectual property problems, but here again there is no lack of innovative resources, such as the renowned yet2.com e-marketplace, as well as compensation means implemented by General Electrics or IBM. Every day, new models of creation of value based on collaboration via the world-wide network appear, and their resounding success makes us reflect on this new type of economy, dubbed Wikinomics in the wake of a recent bestseller.

In short, there have been many available technologies over these last few years, in terms of software and hardware, based on proprietary protocols, or not, which help us develop the functional requirements of collaborative work tools. The main difficulty is to define the processes appropriate for their use as well as the management of information levels between the various stakeholders (contracting managers, project managers, partners, subcontractors, even users and potential clients). Indeed, not all partners participating in a system of systems have the same prerogatives, for obvious industrial confidentiality reasons linked to the intellectual and industrial property of some system developments (without forgetting that due to the possible geographical distribution of the team members, the legal contexts protecting the information and authorizing its exploitation may vary greatly). We therefore have to use technology to share as efficiently as possible, within a given development context, protecting people's assets, and developing the partnership rationale as much as we can, for this approach is the only path to success in the design, development and maintenance of a system of systems.

10.10. Conclusion

Throughout this chapter, we have seen how model-driven engineering helps us control systems of systems engineering, as well as which tools we should use in

order to do so. In particular, we focused on the system of systems acquisition stage, since studies in the last few years have mostly been devoted to this stage in order to record the main challenges related to the complexity of the study. As far as the use stage is concerned, it seems worthwhile to step away from the usual vision of a system put into service which is maintained in operational condition via curative, preventive and progressive maintenance. Indeed, this approach is too closely linked to a “product” vision of the system of systems, to the detriment of a “service” vision which, following the example of what happens in the field of information systems, naturally falls within an incremental capability approach.

It would be advisable to develop a standardization of the ISO/IEC 20000 (which is directly inspired by the ITIL practices, widely used by computer service companies and tools editors). This comes down to adapting this standard’s five main process types: services supply process (service level management, operation status reports management, service continuity and availability management, inclusion in the budget and service posting management, capacity management, information security management), relationship management process (commercial relationship management, financial management), resolution process (incident management, or how to tackle their causes), control process (configuration management, change management) and release process.

This adaptation seems relatively easy in theory and is built on the precise definition of the services delivered and of their level (service quality, availability, etc.). Indeed, this is not very different from operational contracts defined for instance within NATO, where the commitments of the nations are not only in terms of obligation of means, but rather of capability services.

Beyond this application of the ISO/IEC 20000 to the systems of systems, or even the search for a common framework which would combine ISO/IEC 15288 and ISO/IEC 20000 in order to take advantage of the complementarities of the “product” and “service” process approaches, comes the question of the potential adaptation of the tool suites currently used to assist companies in their services management, following the ITIL practices. We would then possess a relatively complete set of methods and tools to control the systems of systems through their entire life cycle.

10.11. Acknowledgements

We wish to thank Xavier Lecinq (DGA engineer), Marc Pernet (former DGA engineer, currently working for CS), Philippe Sarazin (DGA engineer) and Jacques Montagny (DGA engineer) for their important contribution to the works discussed in this chapter, in terms of specification of need and studies follow-up.

10.12. Bibliography

- [BRI 00] BRILL J.H., CHEVALLIER J., MERCHADOU J.L., *Manuel des meilleures pratiques pour le développement de systèmes spatiaux*, Cépaduès-éditions, Toulouse, 2000.
- [DOD 07] DoD ARCHITECTURE FRAMEWORK, *Volume I: Definitions and Guidelines, Volume II: Product Descriptions, Volume III: Architecture Data Description*, v. 1.5, April 2007.
- [DRI 01] DRIVA K., MARTELLI A., VILLEMUR T., “Cooperative environments for distributed systems engineering: the distributed systems environment report”, *Lecture Notes in Computer Science 2236*, Springer Verlag, Berlin, 2001.
- [EZR 99] EZRAN M., MORISIO M., TULLY C., *Réutilisation logicielle: guide pratique et retours d'expérience*, Eyrolles, Paris, 1999.
- [HOO 04] HOOKS I., “Managing requirements for systems of systems”, *Crosstalk the journal of Defense Software Engineering*, August 2004.
- [JEA 02] JEAN G., *Urbanisation du business et des SI*, Hermès, Paris, 2002.
- [KAM 02] KAM L., LECINQ X., LUZEAUX D., CANTOT P., “ITCS: the technical M& S infrastructure for supporting the simulation-based acquisition process”, *NATO Modeling and Simulation Group Conference*, Paris, 2002.
- [LIP 00] LIPNACK J., STAMPS J., *Virtual Teams: People Working Across Boundaries with Technology*, John Wiley & Sons, New York, 2000.
- [LUZ 03] LUZEAUX D., “La complémentarité simulation-essais pour l'acquisition de systèmes complexes”, *Revue de l'Electricité et de l'Electronique*, vol. 6, June 2003.
- [MAI 02] MAIER M.W., RECHTIN E., *The Art of System Architecturing*, 2nd edition, CRC Press, Boca Raton, 2002.
- [MEI 98] MEINADIER J.P., *Ingénierie et intégration des systèmes*, Hermès, Paris, 1998.
- [MEI 02] MEINADIER J.P., *Le métier d'intégration des systèmes*, Hermès, Paris, 2002.
- [MEY 01] MEYERS B.C., OBERNDORF P., *Managing Software Acquisition: Open Systems and COTS Products*, Addison-Wesley, Upper Saddle River, 2001.
- [MEY 06] MEYERS B.C., SMITH J.D., CAPELL P., PLACE P.R.H., *Requirements Management in a System-of-Systems Context: A Workshop*, Carnegie Mellon University Software Engineering Institute, CMU/SEI-2006-TN-015, 2006.
- [PIN 02] PINET C., *Processus d'ingénierie du logiciel: méthodes et qualité*, Pearson Education France, Paris, 2002.
- [ROS 07] ROSEN E., *The Culture of Collaboration: Maximizing Time, Talent and Tools to Create Value in the Global Economy*, Red Ape Publishing, San Francisco, 2007.
- [TAP 07] TAPSCOTT D., WILLIAMS A.D., *Wikinomics: How Mass Collaboration Changes Everything*, Portfolio Penguin Group, London, 2007.

Chapter 11

Model-driven Design and Simulation

11.1. General points

Since the industrial era, our world has undergone major changes, whether it be on a geostrategic, politic, economic, social or technological level. Today's world is multipolar. A multitude of agents are interacting with one another. The dynamics of such interactions are more complex than they used to be, both because of a heightened connectivity and dependence between these agents, and the uncertainty about emergence properties relative to those dynamics. The global economy is stimulated by growing activities and exchanges between governmental and international organizations, enterprises and individuals. It is also sustained by multiple new technologies, such as nanotechnologies, biomedicine, genetics, robotics, NTICs, etc. These technologies have deep repercussions on our society.

The notion of networks is of increasing importance: more and more often, individuals, organizations and systems are organized into networks: computer networks, influence networks, old classmates networks, partnerships and alliances, etc. The digitalization of information and the systems' mobility and modular nature are only facilitating this tendency. It has even become necessary, individuals moving more easily and readily than they used to. This need emphasizes the constraints of interoperability between the components of a system or a system of systems.

Technical and technological progress allow the marketing of more and more integrated systems, which feature various and varied functions. To speed up these products' release, business enterprises must optimize the specification, design and

development delays. But they must also fight competition by lowering their production costs, among other things. Faced with such challenges, reuse seems to be able to meet the strong requirements, which are only getting stronger with the technologies' rapid evolution, the changes in the industrial scenery and the regulations.

It would be naive to think that all or part of a system or a system of systems might be reused regardless of the context. Generally speaking, an existing component is reused with a set of components – already existing or to be conceived – in order to reach a given purpose. This shows how reuse requires a certain level of interoperability between the system's constituents and between the system and its environment, but it must also take into account the need, the context and the use's purpose.

The increasing complexity of systems and systems of systems is making this double issue of interoperability and reuse all the more difficult to solve. According to Le Moigne [LEM 95], complexity can neither be attributed to the increasing number of components, nor to their level of interaction. Rather, it is linked to the unforeseeable character of emergent behaviors, knowing that a complex system is a combination of implex components (which cannot be broken down any further without loss of data) in interaction, rather than a disjointed sum of these components.

This complexity cannot be studied through an analytic, Cartesian approach. This is why Le Moigne advises the use of a systemic modeling approach, which helps achieve better comprehension of the complex systems by relying on the following basic concept: what is the system doing? The answer to this question is a necessary, but not sufficient, condition to favor such interoperability and the components', the systems', or even the systems of systems' reuse. Knowing the environment, the context of use (purpose), the structure (static aspects), and the temporal evolutions (dynamic aspects) of the system is imperative. This approach is at the base of the theory of general system [LEM 94] and the system engineering standards follow this methodological framework in detail, featuring support tools for its implementation.

The same methodological approach can be found within the software engineering community, thanks to the Object Management Group's (OMG) efforts of standardization since 2000. The OMG recommends model driven engineering (MDE), with a particular example of this methodology, namely model driven architecture (MDA). MDE and MDA offer a methodological framework and associated tools which help with the interoperability and reuse of all or part of the complex systems, for several reasons:

- model driven engineering is coherent with the ideas laid out in [LEM 95] about the modeling of complex systems;

- it is complementary to systems engineering: on the one hand, the current systems are software-intensive. On the other hand, since simulations are used with increasing frequency during the systems' life cycle, the models associated with said simulations naturally belong to systems engineering;
- its basic principle, relative to the separation between business logic and technological aspects, facilitates the *capitalization* of knowledge, necessary for reuse;
- it helps verify and validate models, thanks to the control mechanisms of the (meta-) information featured in the models (see *infra* for the details).

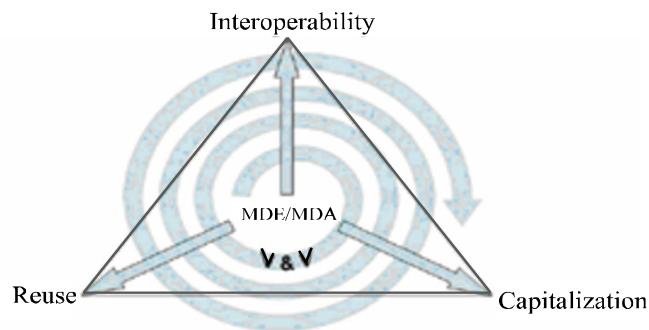


Figure 11.1. *MDE and MDA favor interoperability, reuse and capitalization*

This chapter aims at showing how MDE and MDA can favor interoperability, reuse and capitalization (see Figure 11.1), upstream of a system's life cycle, from the analysis of need to the design (the development and deployment may be automated). In order to avoid any ambiguity, we will first quote some definitions and works about modeling and metamodeling, simulation, as well as testing and validation. We will then briefly expose the state of the art on the MDE and MDA approaches. Finally, we will illustrate those concepts through examples of implementation within the research and technology programs led within the French Ministry of Defense.

11.2. A few definitions

Experience shows that it is often better to define a common vocabulary for a particular field, so as to avoid any ambiguity in the understanding and interpretation of the ideas which will be discussed between participants. This is why we deem it useful to dedicate a few paragraphs to the terminology used in the context of systems of systems engineering, driven by models and simulations.

11.2.1. Modeling

The art of modeling goes back to the paleolithic era, at the very least (around 30,000 years B.C.), during which the human being started putting his observations on a medium (rock, wood, bones, etc.). Cave paintings can thus be considered as examples of modeling of animals, anthropomorphic subjects, hunting scenes, etc. (see Figure 11.2). It is important to note that, in these examples, the result of modeling – the model – is the *representation* of a subject (object, being, phenomenon) belonging to the real world [DOD 94]. This representation might be achieved through imitation or a mental process [VOL 04], about a subject existing or not. These various ideas were well expressed by Joseph Nonga Honla [NON 00]: “the model is the artificial representation which “one constructs in his head”, and which is “drawn” on a physical medium.”



Figure 11.2. Cave painting (Lascaux) showing a man crushed by a bison

As is demonstrated by the cave painting, this artificial representation is a *simplification* of the real world. It does not transcribe every detail or the whole complexity of the observed reality, but only certain *points of view* (or aspects) which the author is trying to express, with a certain level of *abstraction*, through the use of signs or symbols, which have a meaning (signified) for the person which receives them. Several interoperability issues are apparent. Do the author and the receiver share the same knowledge and use the same signifying symbols? Do the latter properly transcribe what the author is trying to express? Do said symbols hold the same meaning for the receiver? The key to these answers lies in the link between the signified and the signifier, but also between the emitted signified and the received signified. This interface issue should be given due attention. The systemic approach undeniably follows this logic, by laying more importance on the interfaces than on the components of the studied system. We will see in the following that the MDE and MDA approach completes this systemic approach through the concepts of *metamodeling* and *transformation*.

The elaboration of a model generally serves a purpose. With cave painting, the author was probably looking to transmit a message or share his knowledge as an observer of the scene. The use of such a model is rather limited, since description is static, and people then talk of a *contemplative* model. This is a particular case of *analytic* models, which help explain a precise number of properties, and the foreseeable and deterministic behaviors which characterize the complex systems (see [LEM 95] or [MCX 05]). The downfall is that this type of model *a priori* requires exhaustive and explicit knowledge of the system, concerning the description of its components, the relationships between them and the precise links between the whole and its parts. This explicit character is very useful when it comes to controlling the description, in that we can think that what is precisely described is perfectly known. The mathematical progresses of the 20th century, and in particular the demonstration of the impossibility to display explicit solutions to certain equations (for example partial derivative equations found in fluid mechanics, which are at the base of turbulence modeling in aeronautics or meteorology), have put an end to this vague desire for control with fully analytical models.

A new family of models was therefore developed, based on another paradigm: modeling is used to formulate problems in order to achieve better comprehension, or reach a better solution, through simulation, within which these models can be executed. Modeling is then defined as [MCX 05]: “the operation through which a phenomenon’s model is established, so as to offer a representation of said phenomenon which can be interpreted, reproduced, and simulated.”

The executable models are a real improvement over contemplative models, for they help integrate every dynamic aspect of the modeled system. The expression “every dynamic aspect” is at the heart of the problem, so let us define it more precisely. From a strictly mathematical point of view, they represent the dependency of certain descriptive parameters on time, and a paradigmatic jump doesn’t seem to apply to the models that are used. But if we take the informal definition – without getting into mathematical details – which says that an analytical model is *a priori* the data of a set of form equations fixed once and for all, it is obvious there might be difficulties in taking some things into account, for example changes in structure, which may be caused by the failure of some components, the replacement or the insertion of new components. On the contrary, an executable model is constructed on a generative paradigm. Control is *a priori* replaced by an iterative search of approximate models, which eventually leads to control.

This having been said, the problem then lies in the construction of executable models able to represent the real world’s complexity. Attempts at using methods such as expert systems, hence with simple rules written along the lines of “if premises, then conclusion”, have turned out to be much too limitative for the modeling of the emergent behaviors which characterize the complex systems. This

result is not surprising, since the expert systems define a finite number of foreseeable, deterministic rules. We find the pitfalls highlighted by the analytical approach. The obtained models are therefore more adapted to complex systems, and it would be unrealistic to want to use this type of modeling to answer problems of a complex nature, even if acceptable solutions may be found and fit. The *systemic method* of modeling should be better adapted to remedying this [LEM 95].

Systemic modeling is based on phenomenology and teleology hypotheses. Which means that the modeled phenomenon or system explicits its functions and functioning, and also its purposes. The semantic cohesion (or congruency) is more important there than in the formal coherence of the modeled system [MCX 05]. We will later see how model driven engineering helps respect semantic cohesion. If systemic modeling helps achieve a better comprehension of complex systems, the complexity paradigm developed by Morin [MOR 77] offers, a conceptual modeling framework of phenomena perceived as complex by the observer-designer. This paradigm considers that complexity is organized, recursive and organizing, depending on which point of view the modeler is interested in. Following systemic modeling principles, these points of view may be modeled as quasi-breakable systems, which means they are defined by the interrelations networks between subsystems, the input-output relationships of each subsystem, as well as the relationships which link the system's input-output to the subsystems' relationships with the environment. These principles form the epistemic basis of MDE and MDA modeling.

This brief study of the modeling concepts and the associated works is enough to make the reader aware of some of the difficulties which may be encountered during the modeling of complex systems. More difficulties will arise with the reuse of those models, such as: verification and validation, as well as the models' capitalization. We will see how model driven engineering claims to bring pragmatic elements of solution to those problems.

11.2.2. *Metamodeling*

Beyond its basic meaning of “after, beyond”, the Greek prefix “meta” is often used in scientific language to express self-reference. Thus, metamathematics is the mathematical theory of the foundations of mathematics, then considered as objects of study. In computer sciences, “meta” designates a higher level of abstraction, a model (e.g. metadata is a model of data).

The two following definitions shed some light on the meaning of metamodeling in model driven engineering:

- metamodeling is the “definition of a set of concepts, properties, operations and relationships between concepts, whose purpose is to define all the necessary entities during the modeling of a specific system” (see [KAD 05], p. 9);

- “metamodeling acts as a toolbox: it captures the variety of the models’ properties, articulates models with one another, insures the data mapping between these models, etc.” (see [MET 05]).

In short, metamodeling goes beyond modeling, reaching a higher level of abstraction; we might say that metamodeling consists of model modeling (produced through modeling), following the idea of self-reference. This notion of “model of model” seems too vague and difficult to understand. To try and be more specific, the computer lingo would consider a model as an “instance” of a metamodel (produced through metamodeling). However, in order to avoid any possible confusion between the object-centric concepts – in which the notion of instance takes on an operative character – and those of model engineering, an ambiguity which is indeed not fortuitous, as will be demonstrated by the digital implementations of methodologies, usage recommends the use of notion (or property) of *conformity* to qualify the relationship of a model to its metamodel [BEZ 04]. Thus, a metamodel “encompasses” a set of conform models; in order to guarantee such conformity, metamodeling must define the implemented methods and languages, used during the modeling process.

Respect of this conformity helps favor interoperability between models, on both the syntactic and semantic levels, depending on the models’ level of abstraction. In [BEZ 01], the authors claim that the notion of metamodeling is strongly linked to the notion of ontology in knowledge engineering. Let us remind you that an ontology is defined as the specification (formal description) of a conceptualization (a way of describing on a certain level of abstraction) of a knowledge field within a specific context [GRU 93].

The result is displayed as a semantic network which links the concepts together through taxonomical relationships (for example, hierarchy of concepts), via, for example, relationships of composition and heritage, from the point of view of object-oriented languages and semantics. The existence of a description of those semantic relationships precisely helps align the signifier and the signified, and hence increase the level of interoperability.

Whether it be for metamodel or ontology, we should define a language fit to describe them. This language then becomes the meta-metamodel which, in comparison with the metamodel, must also provide a set of concepts, properties, operations and relationships between concepts, in order to define all the entities needed during metamodeling. The concepts used in metamodeling also apply. The level of abstraction in meta-metamodeling is higher than in metamodeling: a

metamodel must apply to its meta-metamodel. In fact, the latter provides a unique language of definition of the set of metamodels (see [KAD 05], p. 9). This uniqueness guarantees interoperability between interacting metamodels, since those are described with the same language on a level of abstraction where semantics are taken into account in the description.

Now, the reader is certainly wondering whether this stack of “meta” could not be followed on to further elevate the levels of abstraction. In fact, the OMG offers a four-layer architecture (see Figure 11.3): M0 (real world), M1 (model), M2 (metamodel) and M3 (meta-metamodel). The last level, M3, is reflective, or self-referent, which means it can be described with its own language. The MOF (*meta object facility*) language recommended by the OMG features this reflexive property. Other languages are available for the design of metamodels, such as ECore, KM3 or DSMDL (which belongs with the DSL Tools). One goes down the metamodeling stack through vertical transformations, also called refinements, which can be of various nature: specialization, elaboration, development, derivation, breakdown. Readers wishing to see the precise definition of such refinements can refer to [GRE 04] (Chapter 14, p. 458). In addition to these vertical transformations, there exist horizontal transformations, when these operate on models or metamodels of the same level of abstraction. It is of course possible to combine the vertical and horizontal transformations. We will study these subjects in a later section.

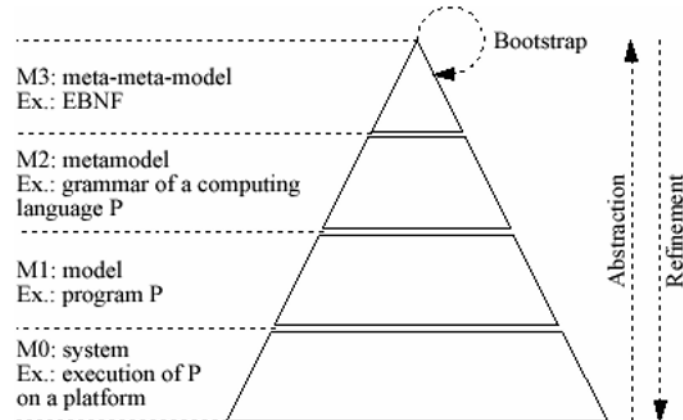


Figure 11.3. OMG multilevel metamodeling stack

11.2.3. Simulation

A rather general definition of simulation likens it to “a method for implementing a model which will evolve over time” [DOD 94]. Sometimes, several models are

necessary for the composition of a simulation. Depending on the criteria (field and purpose, techniques and means, temporal characteristics, etc.), the simulations can be classified in several categories. Table 11.1 provides an example of such classification. Of course, other classifications are possible: discrete event simulation (discrete temporal variable), distributed simulation, etc. (see [DGA 97, DOD 94] for other types of simulation). These few examples clearly show the complexity which might arise from a simulation which becomes a complex system in its own right, system which will have to be controlled.

Within this chapter's context, we focus on digital simulations, whose models are implemented as codes executable on a computer. This focus is motivated by the following reasons:

- the current systems are software-intensive;
- digital simulation offers many advantages, essentially in terms of cost and flexibility, compared to simulations which use physical models (e.g. scale models);
- digital simulation can be used all through a system's life cycle, from its feasibility study to its disposal: this is all the more important in the light of today's requirements on the ecodesign of complex systems, where we must, from the first stages, evaluate the impacts which the entire life cycle might have on the environment or in relation to a requirement on sustainable development.

Man in the loop	Real time	Hardware in the loop	Type of simulation
no	no	no	Constructive non-real-time
no	yes	no	Constructive real-time
yes	yes	no	Piloted
yes	yes	yes	Instrumented
no	yes	yes	Hybrid
yes	no	no	Interactive
yes	no	yes	Not applicable
no	no	yes	Not applicable

Table 11.1. *Simulation classification (featured in [KAM 02])*

In the field of defense, simulation appears as a mandatory tool to control performances, delays and costs in the design of large systems. The concept of an integrated and concurring use of simulation means in the various stages of a system's life cycle was first introduced by the Defense Systems Management

College in 1998, in the United States, under the title *Simulation Based Acquisition* [JOH 98]. Works have demonstrated the economical profit which may be garnered from this approach: [LUZ 03], for example, is based on the cost model COCOMO 2.0 to show that resources allotted to simulation based acquisition are instantly compensated by the cost premiums resulting from the first incident happening during the acquisition process.

It is rather intuitive to imagine that the use of simulation contributes to the early identification of mistakes and therefore reduces cost premium risks, or even plain project failure. Indeed, it helps verify specifications and validate concepts developed in the stages preceding the life cycle; reduce the risks of development and integration errors; prepare and scale qualification tests; and treat the problems of use and disposal of the system before they actually occur (such as integrated logistic support, recycling, etc.).

However, the implementation of a simulation requires us to ponder about interoperability and reuse. These two themes imply the verification and validation of data, models and simulations, but also the capitalization of knowledge. The MDA approach, and in a broader way model engineering, therefore present a certain potential in favoring the resolution of such problems.

11.2.4. Interoperability

This neologism rather explicitly expresses the notion of capability to operate together. This basic notion is found in many definitions, more or less precise and specific to a field of application (see [DGE 01, GRE 04, MIC 07, OTA 07]). Whichever meaning we choose to follow, interoperability is increasingly considered as an important aspect in the design of a system. It has a positive impact on the whole value chain, through the increase of productivity, decrease of costs and, in the end, improved client satisfaction. To achieve this, efforts must be made to standardize and open the systems [MIC 07], despite the quasi-monopoly and the misgivings some giants of informatics may have.

The field of telecommunication is a pioneer in the search of interoperability. As early as 1865, a great number of European countries had already grasped the necessity and utility of the implementation of a common coding for the transmission of telegraphic messages from one country to the other. The creation of the International Telecommunication Union (ITU) breaks another barrier by tackling the standardization of international telecommunications. However, it should be noted that the standards issued from the ITU workgroups provide technical recommendations. Thus, following these standards only ensures interoperability on a technical level, even if semantic information (network communications) may transit

within the “presentation” and “application” layers. This is not surprising, for these standards are defined in relation to the layers of the OSI (*Open Systems Interconnection*) reference model, whose level of abstraction lies on the M1 level of the multilevel metamodeling stack.

Rapidly, it becomes clear that the level of technical interoperability is not enough for the target system to ensure proper operation in its context of use. Indeed, this level of interoperability is focused on the signifier of the exchanged data and not the signified meaning. In other words, it does not take semantic aspects into account. It is commonly acknowledged that these two levels of interoperability (technical and semantic) must be sufficiently covered in order to guarantee a system’s proper operation. In fact, more is needed to reach complete interoperability, whose definition and characterization are still left to harmonize. Attempts at conceptualization of interoperability levels have resulted in reference models such as the *Levels of Information Systems Interoperability* [DOD 98], or the *NC3TA Reference Model for Interoperability* [OTA 03]. According to the [TOL 03] authors, both models ensure the coherence of data, which is a necessary but insufficient condition to ensuring the interoperability between applications. They are therefore offering an alternative model, broken down into five *Levels of Conceptual Interoperability Model (LCIM)*, which also requires the documenting of the interfaces:

- level 0 (system specific data): the data is used within each system in a proprietary way; there is no interoperability required;
- level 1 (documented data): the data is accessible through interfaces; the data and interfaces are documented using a common protocol;
- level 2 (aligned static data): the data is documented using a common reference model, based on a common ontology;
- level 3 (aligned dynamic data): the use of the data is well defined using standard software engineering methods such as ULM;
- level 4 (harmonized data): the semantic coherence of data is guaranteed by a conceptual model.

Level 1 corresponds to the technical interoperability, in which the physical connections and network layers are taken care of: data can be exchanged following standard formats and protocols. Such is the level of interoperability reached by the reference model OSI, previously talked of. Level 2 concerns the meaning of data. This is the semantic interoperability, where the description of data must be precise and non ambiguous, through the use of an ontology which structures the concepts according to a semantic graph, whose edges express the semantic relationships between these concepts. When combined, levels 1 and 2 answer the previously

raised questions about the link between signifier and signified. The answers relative to the link between the received and the emitted signified are found from level 3, which studies what should and can be done with the received data so as to heighten the level of interoperability. As for the last level, it seeks to achieve a common and complete vision of the modeled field, through a conceptual model. The latter must describe what is modeled and what is not, namely the limits and constraints. Within a systemic approach, this means properly defining the borders of the modeled system, as well as its interfaces.

This succession of levels of interoperability can be compared with the multilevel metamodel stack; in both cases, a higher level in the stack goes with a higher abstraction. The two models can even be matched: the LCIM level 1 corresponds to the level M1, levels 2 and 3 can be associated to the level M2, and level 4 can be compared to level M3. Following the hypothesis according to which a higher level of abstraction of the models favors reuse, it can be deduced from this comparison that a higher level of interoperability also favors reuse. Conversely, the positive correlation between interoperability and reuse also helps verify that a higher level of abstraction favors reuse.

It therefore seems natural to treat these issues of reuse and interoperability conjointly. The fields of application which are concerned by these issues may very well make use of the LCIM model, which remains sufficiently general despite coming from the world of simulation.

11.2.5. *Verification and validation*

As with the other concepts we have talked about so far, no common definition exists to describe the activities of verification and validation (V&V) between the communities of system engineering, software engineering and modeling and simulation. In that last field, a general consensus is still to be found, essentially because of the partial character of the models' and simulations' representation of the real world. However, concepts can be found, common to the various definitions.

To put it bluntly, verification controls that the work has been properly done, and validation ensures that the proper work has been done. To be more specific, verification is the control that a product of an activity or a process (example: specification, design, development) follows the entry requirements of said activity or process, that is to say that the resulting technical characteristics follow the requirements. As for validation, it consists of ensuring that the product of an activity or a process applies to the need for a precise purpose. The concept of "efficiency" is developed in the works of Le Moigne.

Therefore, conformity is judged against a referent, made of requirements in one case and needs in the other. Both types are linked, since the requirements stem from the need. From this observation, we can easily deduce that validation is more difficult than verification, considering the commonly acknowledged difficulties in compiling the need and transforming it into requirements. As for verification, the underlying activities seem more easily apprehensible, since, on the one hand, system and software engineering recommends expressing requirements in a verifiable, quantified way, and on the other hand, there exist numerous techniques and tools issuing from software industry, with one of the most promising approach based on formal methods. The latter however require large resources and calculation time, and are still reserved for critical systems in which test coverage must be at its maximum. In most systems, the residual errors can therefore endure after the V&V process.

If the total eradication of such errors, even when it is technically doable, is not economically viable, the favored solution would be to contain them, so as to guarantee *Fitness for Purpose*. To ensure such fitness, the results of V&V constitute tangible proofs to inform, as much as possible, on the field of validity of the target system, in terms of capacity, performance, constraints and limitations, depending on the purpose and the context of use. It seems natural to think that, the wider the V&V coverage, both in width and depth, of the detail of the possible tests and try-outs, the better the evaluation of the fitness for purpose.

This perception is not erroneous, but may induce an error if the quality of the tests and try-outs is not high enough. Perception of quality is associated to the level of trust built up by the beneficiary (user or decision maker), not only from the objective proofs collected by V&V, but also according to the experimental framework and operating mode, the skills and fame of the operating agents, as well as the processes and organizations implemented to support them. The authors of [SCO 03] have even introduced the idea of evaluating the maturity of the V&V process according to a five-tiered maturity model, akin to the CMMI (*capability maturity model integration*). The next logical step would be the implementation of a system of independent certification of the ISO kind, which would probably have a positive impact on the perception of quality. We would then have an addition pertaining to software engineering, for system engineering purposes, which would complete purely software-related standards such as ISO/IEC 9126 (software engineering: product quality), ISO/IEC 15504 (ISO/SPICE process assessment) and ISO/IEC 14598 (software engineering: product evaluation).

The works of the European project on research and technology, REVVA, taken over by the SISO (*Simulation Interoperability Standards Organization*) standardization group [SIS 07], introduce the idea of the V&V process's independence, by offering to have it designed by a third party, other than the system's contractor or its project manager. From this independence emerges the

importance of distributing responsibility between the various stakeholders. Of course, one must be aware that a better fulfillment of needs and a higher level of trust both require human and financial efforts. Since residual errors introduce uncertainties characterized by risks of use or reuse, a compromise will have to be found between the beneficiary's personal tolerance towards risks, and the efforts put into the V&V process.

From there, the works of project REVVA have helped clarify a hitherto confusing terminology around *acceptation*, and most of all *accreditation*, in the international community of modeling and defense simulation. The term "acceptation" results from the aforementioned compromise, in the decision to use or reuse the target system for a given purpose and within a particular context; this is indeed the term we are interested in. The term "accreditation" corresponds to the procedure through which an organizational authority recognizes that a moral or physical person is apt to perform some specific activities.

The presented principles and concepts must go along with a V&V methodology. It is not uncommon to find V&V processes integrated concurrently with the process of system engineering, or the process of simulation development: for example, the IEEE standard 1516.3 relative to good working practices for the development and federation process HLA (FEDEP), which follows a classic V model of system engineering, integrates a subprocess *VV&A Overlay*. This paralleling of process is adapted to currently developed systems and simulations. For existing products, integrated within new developments, the REVVA project offers a generic process of V&V methodology *post hoc*, similar to the V model (see Figure 11.4). The descending branch of this generic process starts with the formalization of the V&V need, that is to say the analysis of need and context. This need is then developed into a structured, arborescent set of acceptance criteria (*Target of Acceptance: ToA*) which are themselves developed into a set of V&V criteria (*Target of V&V: ToV*) from which the V&V director can plan all the underlying activities. The satisfaction, or lack thereof, of the V&V criteria by the items of evidence, is then analyzed and described within a V&V report. The final stage consists of evaluating the satisfaction, or lack thereof, of the need for V&V by the aggregation of the acceptance criteria. The beneficiary has the final responsibility of accepting or refusing, based on the acceptance report which was produced during that stage.

A database must be built up at the heart of this process: it helps capitalize on all the data and the knowledge issued from the V&V activities. It would be useful to link it to a business database featuring the reference business data. These databases constitute a precious source of information for any beneficiary wishing to reuse products that have been verified and validated. Moreover, thanks to these pieces of information, including the knowledge of the products' field of validity, the interoperability with other products is favored. Without this information and the

agents involved in these products' development, a much longer work of reverse engineering would have to be undertaken, and it would not necessarily produce all the necessary data. Such a situation would be comparable to the works of archeologists, who try to determine the origin of the previously shown cave painting, the true message its author was trying to transmit, its original purpose, etc.

This process is sufficiently generic to be adapted to the various V&V objectives, the nature of the investigated products, the available human and financial resources, and the risks tolerated by the beneficiary. Within the project REVVA, case studies led on simulations for acquisition and training simulations have shown the way this process can be applied to such types of application. This generic process is currently analyzed for standardization within the SISO, which has become a chapter of the IEEE dedicated to the field of modeling and simulation. On the other hand, this generic process is not necessarily adapted to the V&V of technical-operational simulations which implement performance or behavioral models, helping appreciate the operational efficacy of a future system used within a given scenario depending on its unitary performance (see [RAB 06, RAB 07], in which a process is based on the standardization of the notion of trust and credibility of simulations, which includes the levels of relative skill and trust of all stakeholders, on top of strictly technical criteria). Keeping that knowledge in mind, the reader must be conscious that as long as no international V&V standard exists, and no sufficient hindsight on these methodologies is had, we will have to choose between the various existing V&V methodologies and adapt it to our needs.

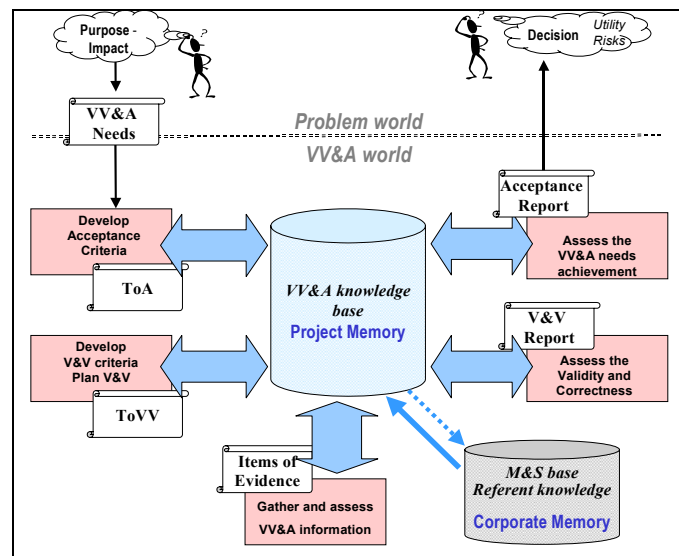


Figure 11.4. The GM V&V Generic Process (quoted from [SIS 07])

11.3. Model-driven engineering

11.3.1. *The MDA conceptual framework*

The MDA architectural model is a standard from the OMG, whose main purpose is to ensure the durability of the corporation's business knowledge against the rapid evolution of technologies. Since 2000, the OMG sponsors the MDA approach as an alternative to the failed attempts at standardizing CORBA, in the 1990s, as a lone middleware able to guarantee real interoperability of the software components issued from heterogeneous sources. The main reasons for such a failure stemmed from the over-dependency of the CORBA components on technologies.

The offered conceptual and methodological framework provides a set of directives to structure the specifications which have been given as models through UML (*unified modeling language*); then, a code is created for any operating platform (see [OFTA-4 04] for a proposed definition of a platform). Taking the specification of the system's functionalities away from any notion of implementation on a technological platform helps implement the paradigm "write once and generate everywhere". Actually, the IT press [ITE 04] admits that, with the MDA approach, models went from the contemplative to the productive mode.

The associated method of software engineering is derived from the object method and even complements it, as underlined by [BEZ 04]. On the one hand, the UML language, the MDA basis, has its roots in object-oriented methods of analysis and design, such as: OMT (*object modeling technique*) or Booch (see Figure 11.5). Moreover, in both cases, a higher level of abstraction is required (classes in object-oriented methods, metamodeling in MDA and MDE). The main difference is that in model-driven engineering, the focus is put on models, not objects, and a higher level of abstraction is possible. This helps avoid running headlong into development details (and therefore the solution) as early as the upstream stages of specification and design of the system's life cycle, as is often the case in traditional object-oriented methods, whose adaptability to evolutions is limited.

Let us remark, however, that the recent evolutions of the object-oriented technologies – design patterns and aspect weaving – are trying to compensate for this problem of adaptability in a way close to some underlying MDA principles. The idea to build design patterns as architectural components, which can be reused to solve a recurring problem, stemmed from the works of architect C. Alexander [ALE 77]. For every pattern, there is a purpose (the problem to solve), the problem's solution, and the limits of its use in a given context. In practice, the construction of a pattern goes through the analysis of the commonality (common features) and of the variability (structures liable to change) of the problem's field [SHA 02]. The commonality is obtained through abstraction, based on variations of specific

concrete cases. This abstraction must integrate as global a vision as possible on the field, in order to ensure the maximum level of interoperability (level 4 of the LCIM model shown above). Thus, the commonality provides a certain structural stability and robustness, while variability characterizes the pattern's aptitude at being reused for various problems in one unique field. This separation between commonality and variability can in fact be compared to the MDA approach, in which the first would correspond to the business process, and the latter to the technical characteristics of the technological platforms which implement these processes.

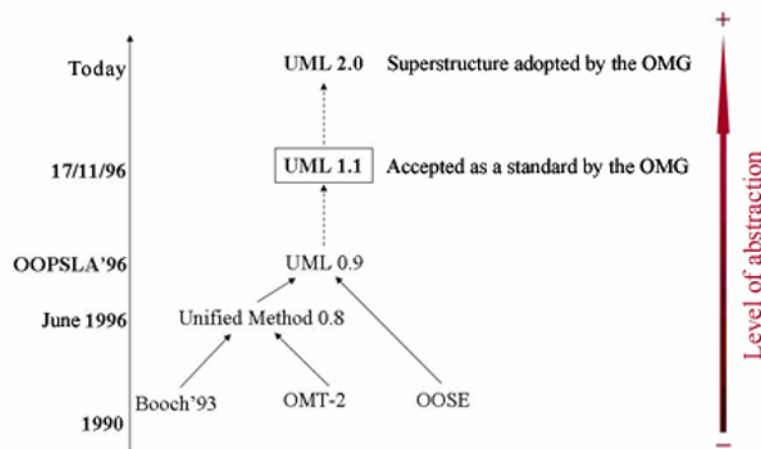


Figure 11.5. Relationship between UML and object-oriented methodologies with important temporal milestones (OOPSLA was the main meeting of the IT community around object-oriented languages)

The starting point of the MDA methodology is often presented through two separate models: PIM (*platform independent model*) and PSM (*platform specific model*). As demonstrated by their names, the former is independent from the platform, unlike the latter. The MDA principle requires modeling components to be waved between the PIM and the PSM, before applying a set of transformations in order to obtain a code which can be compiled and run on the target platform. The OMG reference documents mention the necessity of an intermediary model, a so-called PDM (*platform dependent model*), to go from PIM to PSM, but the definition of such a model is rather confused. Thus, in [OFTA-4 04], the authors offer to clarify the definition and role of a PDM. Their vision can be summed up by the right-hand scheme of figure 11.6. Putting it in parallel with the traditional Y-shaped architectural process used in system engineering, the PIM would then be akin to the functional architecture which characterizes the business aspects: the PDM would correspond to the technical architecture which defines the technical components and

their technical characteristics; the weaving stage between the PIM and the PDM would be akin to the stage of allocation of the technical components to the expected functions; and finally, the PSM and the code obtained through transformations can be compared to the physical architecture, specific to chosen technologies.

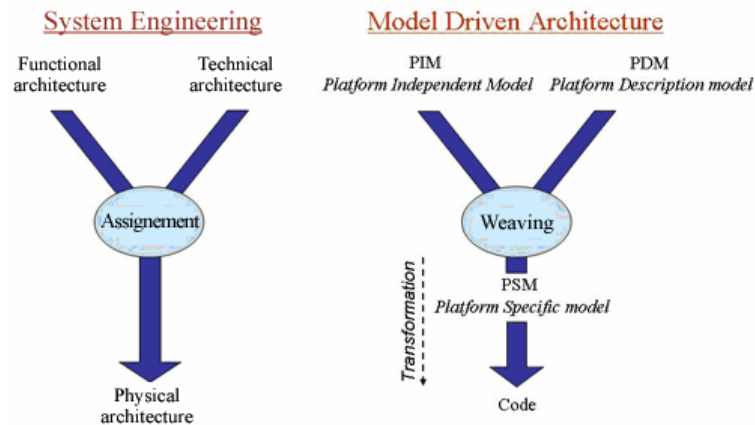


Figure 11.6. *Parallel between system engineering and MDA*

This break-up between PIM and PSM must go along with the appropriate organization of design components. The authors of [SHA 02] recommend encapsulating the objects' variations and composition, rather than classify them via class heritage relationships, as is done in object-oriented languages. Composition helps achieve a more modular architecture, and encapsulation helps isolate complexity-generating variations through the successive addition and modification of the design details. According to Brooks [BRO 87], this so-called accidental complexity can be reduced, in opposition with so-called essential complexity, which is intrinsic to the real world problem. Nevertheless, the authors of the introductory chapter [OFTA-1 04] of the collective work [OFTA 04] rightly underline that modeling helps reduce the intrinsic complexity of the real world by ignoring the details which do not pertain to the expected purpose. By controlling the refinement process within the developing process, accidental complexity is gradually controlled. The MDA approach is actually trying to achieve such control by offering to automate refinement through vertical transformations leading to the production of a compilable and executable code. These transformations should also be modeled as models. This is not enough, however. It is most important to verify and validate them, so as to enable reuse with a good knowledge of the associated quality, and therefore of the level of trust which a beneficiary will have when reusing it for his own benefit.

11.3.2. MDA methodological framework

MDA's main methodologic innovation lies on the level of mapping. In system engineering, the components of the technical architecture are manually assigned to functions, while in MDA methodology, the models' weaving and transformations can be automated if the transformations and the use of MDA tools have been previously modeled (see the OMG's Internet site <http://www.omg.org>, which supplied the list of support tools, or [OFTA-8 04], which offers a synthesis of said tools' evaluation). This is the main plus of this software engineering approach in which "everything is a model". Hence, the code becomes disposable, since it can be automatically regenerated, and becomes similar to a commodity without any real added value. The transformations, acting as interfaces between models, take on new importance, such as is the case with the interfaces between components in systemic logic. Better still, the conformity link between two consecutive levels of abstraction up to the highest level (meta-metamodel or conceptual model of reference compatible with the MOF) guarantees global coherence and favors interoperability and reuse once all the (meta-) models and the rules of transformations have been verified and validated (see [TOL 04]).

We have previously brushed on the two categories of transformation: vertical and horizontal. They are both used in the Y cycle of Figure 11.6: the transition from PSM to coding relies on vertical transformations (for example $TV_{1,1 \rightarrow 0}$ and $TV_{2,1 \rightarrow 0}$ in Figure 11.7), which add details of implementation within models whose level of abstraction is lowered. On the other hand, the weaving between PIM and PDM belongs with horizontal transformations, in which various specifications or conceptions are integrated within one unique specification/conception, without changing the level of abstraction. Vertical transformations may be necessary to obtain the corresponding PSM. In the case of technological evolutions, instead of adapting or directly transferring the existing source code towards a new source code specific to another platform (for example $TH_{0,1 \rightarrow 2}$ in Figure 11.7), we may use alternative solutions, as long as effort is put on the abstraction of PSM and PDM platforms, and a composition of transformations (the symbol \circ denotes the composition operator): $TH_{0,1 \rightarrow 2} = TH_{2,1 \rightarrow 0} \circ TH_{1,1 \rightarrow 2} \circ TH^{-1}_{1,1 \rightarrow 0}$ and $TH_{1,1 \rightarrow 2} = Weaving_2 \circ TH^*_{2,1 \rightarrow 2} \circ Weaving_1^{-1}$.

If we refer to Figure 11.7 in order to understand both these formulas in a natural language, they are thus expressed:

- the transformation between the C1 and the C2 codes happens when returning to the PSM1 model, transformed into PSM2, and the C2 code is generated;
- the transformation of the PSM1 model into PSM2 happens when going from PSM1 back to PDM1 (therefore, through the "inversion" of the weaving process),

then transforms the latter in PDM2, and generates the PSM2 model via the weaving process.

Despite the apparent complexity of these formulas, these transformations are all the easier to achieve for they are done on models with an increasingly high level of abstraction (transformation between models on the level M0 happens via the transformation of models on the level M1, which in its turn happens via the transformation of metamodels on the level M2), and therefore with minimum focus on implementation details which are *a priori* complicating factors and must not take the advantage over structuring aspects, lest the portability be substantially diminished.

This course of action is preferable to facilitate the modeling of horizontal transformations and the maintenance of their evolutions, and therefore increase the levels of interoperability and reuse thanks to a higher level of abstraction. Indeed, the amount of design and implementation details diminishes with the elevation of the level of abstraction, which facilitates the control of the modeled system's complexity. Moreover, the break between business and technological logics will be better controlled on a higher level of abstraction. Should the opposite occur, efforts to perform the break would be vain, for transformations would have to integrate both aspects. For example, on the M1 level, the transformation $TH_{1,1 \rightarrow 2}$ would have to integrate elements from PIM and PDM, to go from one PSM to another (see Figure 11.7).

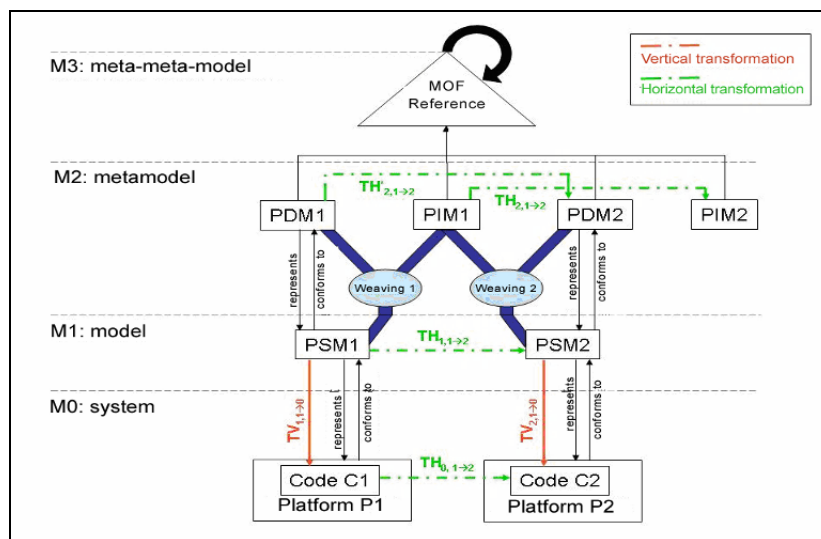


Figure 11.7. MDA transformations on the different levels of the metamodeling stack

Thirdly, even if horizontal transformations may be necessary for the evolution of business aspects (example: $TH_{2,1 \rightarrow 2}$, see Figure 11.7), an automation of transformations performed as early as possible before the MDA cycle undoubtedly improves the system's flexibility towards technological evolutions and reduces production costs and delays.

11.3.3. *Another instance of MDE: the DSL tools*

Even though the MDA approach is the most advanced specific instance of the general MDE approach, its acknowledgement by the software community and its industrial development are still impending, due to numerous critics. First of all, the MDA approach focuses on the separation between the specific or independent aspects of the technological platform, whereas the MDE approach aims at being more general: for example, it includes development methods such as aspect-oriented programming, in which the functional and non-functional aspects (performance, quality of service, reliability, security, etc.) are separated in a modular way, then weaved within an object-oriented application.

It is true that the UML language, an MDA medium, has the merit of being generic and features assets to become a universal language unifying both system and software engineering practitioners. In that way, it helps specify, build, visualize and document the components, the static and the dynamics of complex systems through a thorough representation of these systems into models. However, this generic nature leads to imprecise descriptions and hampers the development of the MDA approach. Indeed, use shows that UML is rather adapted to the informal graphic documentation of design [GRE 04]. A more thorough use of UML must go through specialization by business field, which requires confirmed and rare skills, both in UML and business knowledge. Without this double competence, the recurring problem of dialogue and comprehension between technicians and functionalists (for example an interface problem) is a major obstacle to the adoption of MDA.

Some specialists are even more severe towards the UML language, calling into question the imprecision within the UML language metamodeling, and notably the way semantics are subject to various possible interpretations, but also the limits of UML, notably in the modeling of Java or C# interfaces, and the reuse of UML parts. Even the arrival of UML 2.0 cannot make up for its weaknesses, since this new version does not cover the needs of some aspects of software development, such as the modeling of data and user interfaces (see [GRE 04], Appendix B for more details).

Another difficulty in applying MDA and using the UML language lies in the way no method is provided to guide the user in his modeling work, and notably in the choice of the proper level of abstraction, relative to the metamodeling stack, to meet his needs.

From the standpoint of model transformation technologies, [KAD 05] underlines the weaknesses of the MDA standards. For example, respecting the standard MOF often leads to overly complex interfaces, by imposing the use of IDL CORBA when the standard XMI is often judged too complex for the programming of model exchange. Besides the necessary simplification of the use of MDA tools, it advocates the use of a Framework (software development infrastructure) of model transformation which will *a minima* allow:

- the application of specific design patterns on the models;
- the fusion of a model's various views;
- the generation of code specific to certain platforms;
- the execution of the models' validation tools.

For others [ITE 04], the future of MDA methodology seems linked to the development of a market of transformation components, where a component bought from a supplier might be run on another supplier's transformation engine. At this time, such a commercial and industrial development can hardly be imagined, since an MDA tool editor cannot profit from the marketing of tools which might interoperate with the competition's. At least, such a thing will not happen as long as the field's industry has not reached a higher degree of maturity.

The current trend rather consists of building on DSL (*domain specific language*), materialized through small specialized metamodels, expressed in a language (textual or graphic notation) close to the end-user's business language, in order to separately take into account the systems' various aspects. This approach thus solves the problems stemming from the generic nature and the UML semantic imprecision that we have previously mentioned. For example, through the use of the XML *Schema* (*extended markup language*) format, DSL becomes portable and independent from the general-purpose programming languages designed for IT specialists. DSL implementation must be performed with tools which follow the aforementioned [KAD 05] requirements. In that way, the efforts put in the design and development of business-specific software and systems may be capitalized on, through the factoring of repetitive tasks and the reuse of design patterns encapsulated in verified and validated components. Nowadays, two competing editors offer DSL tools to implement MDE with DSLs: Microsoft's "Visual Studio" or IBM's "Eclipse".

11.4. Feedback

11.4.1. Issue faced by the DGA

Within the French Ministry of Defense, the *Délégation générale pour l'armement* (DGA, General Armament Delegation) prepares the future defense capabilities and conducts armament programs for the French military. It works in close relation with the General Staff, from the identification of future needs to the monitoring of the users' satisfaction. Like in civilian industries, achieving the best product while respecting budget and delay requirements is one of the main, constant preoccupations of project leaders. The way to achieve this is to take advantage of what exists without having to redevelop it for a particular system, and also to make functionalities and performances evolve at the cheapest cost possible, all the while guaranteeing a continuity of service to satisfy the needs of the client or the end-user.

Nowadays, defense capacities are no longer defined by following the simple logic of weapon systems (example: fight plane, tank, frigate, etc.) but rather force systems which offer capability effects (for example: engagement and fight, dissuasion, etc.) and simultaneously involving different armies (land, air, sea), whether national or within coalitions. In this way, the number of interacting components increases inexorably and leads to the increasing complexity of defense systems, which have not necessarily been designed to operate together; their life expectancy becomes heterogenous, their respective life cycles are not synchronized, and their architectures and interconnections vary. Moreover, the predominance of software in those complex systems only heightens the difficulty in controlling the variability of their components in time and space.

To define future systems and be able to do the proper compromises according to the geopolitical context, the analysis of threats, and the documentation use policy, one must have the proper simulation tools. Using simulations in the various stages of a defense system's life cycle is an old practice. However, simulation tools were used in a decentralized, fragmented and non coordinated way. The evolution towards systems of systems goes along with organizational changes: pooling of resources, creation of multidisciplinary teams and coordination of all actions. In this framework, interoperability, reuse and capitalization are the main focus and constitute privileged research themes. The DGA has launched several research and technological projects on those themes, applied to the modeling and simulation of defense systems through the use of model-driven engineering methodologies and tools. The three following parts each describe one of these projects, and their feedback in terms of model-driven engineering.

11.4.2. *Feasibility study of the MDA approach*

The various DGA technical centers and the industry in the defense field own an important catalogue of simulation models and keep developing more. We have noticed that, on the one hand, these models' definition, design, development and operation depend on a great variety of modeling and programming languages (UML, XML, ADA, C, C++, JAVA, etc.); and on the other hand, these models have been developed in order to be integrated in specific and/or proprietary simulation platforms, which operate with specific equipment configurations (machines, physical communication networks, etc.) and software (operating system, communication protocols, middleware such as CORBA or a *run time infrastructure* compatible with the *high level architecture* interoperability standard, resulting from works led by the American Department of Defense).

These platforms can offer integrated work environments, featuring services such as: edition and coupling of models, definition and execution of simulations, definition and implementation of communications, visualization and analysis of results, capitalization, information configuration management, etc.

In 2003, therefore, a project was launched in order to define a level of modeling sufficiently abstract and independent from simulation platforms, to help with the design of simulation models. The expected result is a design chain of models adapted to these platforms and which will enable, from a UML modeling, an automated generation of executable code on said platforms. The attained level of abstraction must allow the models' interoperability and reuse in various contexts of simulation, and therefore a durability of investments. Convinced that the business part is the enterprise's true capital, and that the infrastructure part will be linked to the evolution of IT technologies, the DGA saw a possible solution in the MDA approach.

Instead of taking on the M1 level of the metamodeling stack through the direct definition of a PIM and some model transformations, to obtain a PSM which will then generate executable applicative code specific to one of the target platforms, this study follows another approach: it elevates abstraction to the level M2 and defines so-called MOF transformations (see Figure 11.8) which operate on metamodels with which the PIM and PSM are respectively compliant.

If this approach favors the models' interoperability and reuse, as defined by the objectives, there are however non negligible differences between this solution and the transformations described in Figure 11.7, and which deserve some explanations. The PIM in Figure 11.7 is located on the level M2, whereas it is located on the level M1 in Figure 11.8. This gap in the levels of abstraction is in direct link with the blurry specifications of the OMG on the MDA, and to the difficulty of positioning

oneself on the abstraction scale since the four levels of the metamodeling stack are expressed in a relative way.

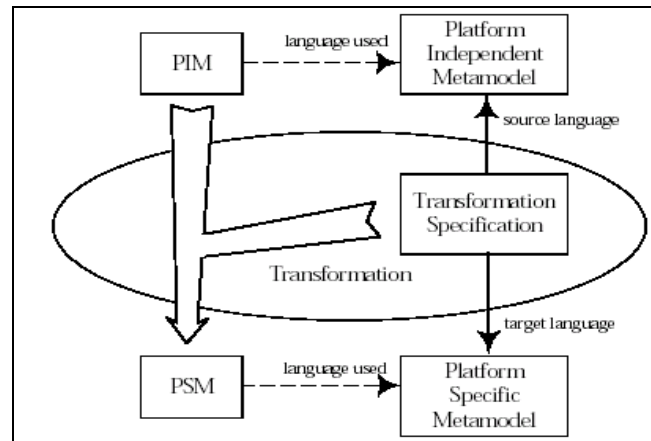


Figure 11.8. Transformations of MOF models (featured in [MIL 03])

Metamodeling goes through the elaboration of a metamodel of the simulation's field, in order to abstract the business concepts and the search for the concepts' community, among the specific variations of simulation platforms. The expression of the metamodels relies on the UML v1.5 language, by using the Rational Rose modeling tool (version 2.0 of the UML was not available at the time of the study).

The exercise shows, however, that because of the aforementioned semantic ambiguity in UML, the resulting metamodeling is too generic and elementary, from the business and technological standpoint. For example, the representation of time cannot be taken into account in a satisfying way. Moreover, the study also shows the necessity of having an in-depth knowledge of the platforms in order to master the scope of concepts in the implemented field of simulation.

As for the definition of the models' transformation, an analysis of the currently available tools recommended the use of the MIA-Transformation tool for the chosen type of transformation (MOF). Expressing transformations with the MIA-TL language, 280 rules of transformation have had to be defined for two of the platforms. Such work required an important effort of design and development.

For one of the modeled platforms, the study did not go past the metamodeling stage, because of the impossibility of modeling its simulation engine's time and dynamics. For other platforms, the automatic generation of the executable code was realized with the language *Rose Basic Script* of *Rational Rose*.

The feasibility of the MDA approach has therefore only been partially proven in that study, which underlined a gap between theory and practice. From that experience, it is clear that the main difficulties that are met during the implementation of the MDA approach are caused by:

- the need to reuse basic classes in every platform;
- the differences of service levels in the various platforms;
- the differences in the models' granularity;
- the incompleteness of the tools which instrument the approach;
- the lack of functional and technical skills.

11.4.3. Feasibility study of the MDE approach

On top of the previous project, in which the models obtained through MDA must adapt to the existing technological platforms, another project was launched by the DGA in 2005 to study the feasibility of a joint evolution of several simulation platforms towards a single platform based on more modern software technologies.

This platform must provide the analysts and developers with an environment for the development and exploitation of models of systems (arms and forces), to transform them into executable code, and exploit and shape the results.

From a methodological standpoint, the target platform must enable the implementation of an MDE/MDA approach, applied to the particular field of technical-operational simulations (which are used during the analysis of need, where the operational need is formalized and the functional requirements, or even some technical requirements on the level of the system, are defined), as the informal description of the simulation are needed for the generation of executable code, with a clear separation between business logic and technical and technological considerations (see Figure 11.9):

- the informal description of the problem is expressed in a free format (text and diagrams), providing precisions on the service functions of the system to be developed. In the case of technical-operational simulations, it describes the scenario which represents the operational context we wish to simulate, and its desired use, that is to say the purpose of the simulation's exploitation;
- the analysis model defines the way the functional needs will be covered, while staying within business considerations. The lower levels components may be modeled, as well as their behavior, and the main operations, both provided and required;

- the untargeted design pattern is obtained through the application of architectural patterns, untainted by the features of the target language, on stereotypes (example: extension of ULM modeling components) which define a component's semantics and the way it should be used in a model;
- the targeted design pattern takes into account the technical considerations (in particular the implementation language) through the application of architectural patterns specific to the target platform. It expresses a more precise modeling (used types, operations signatures, etc.);
- the code itself, following the compiling and link, provides the application for simulation. It is represented by text files, which also include free commentaries as well as complementary structured information (structured commentary, attributes, etc.) which will enable the automatic generation of documentation, or help reverse engineering by keeping data about the design pattern.

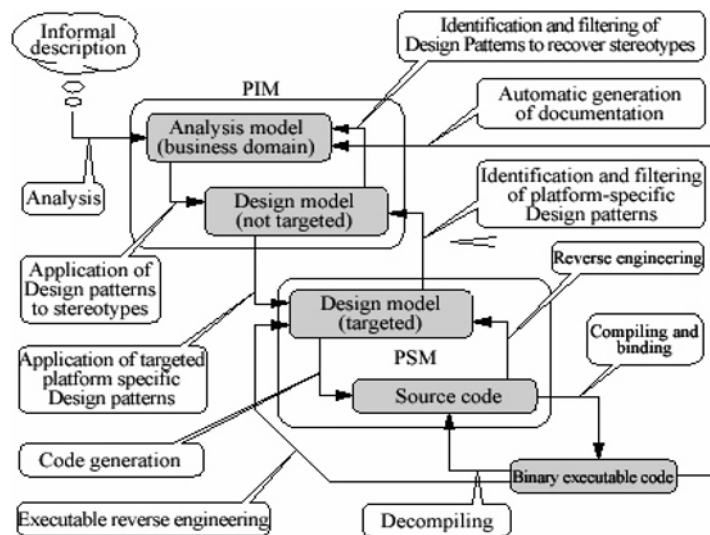


Figure 11.9. Transformations of models within the reference platform

The first stage of the project consists of a definition study, during which the available technologies and tools are evaluated and compared. For the specific field of technical-operational simulations, it turns out the use of DSL Tools is better adapted than an approach purely based on MDA and UML. Indeed, unlike general-purpose languages such as UML for graphic modeling, Java for programming or XML for the sharing of structured data, a DSL is, by its mere principle, a language adapted to particular corporations or specific needs, such as with technical-

operational simulations. Moreover, a DSL is closer to a business language (thanks to the XML *Schema* format), and much more simple to use than ULM for business experts who do not necessarily master every concept of a language of a high level of abstraction. Moreover, a DSL formalism helps capitalize on efforts made in software development by favoring repetitive tasks while providing application developers with an appropriate space of creativity.

Parallel to this first stage, the state technical experts have defined a standard for the capture and analysis of needs within technical-operational simulations, baptized XMS. This new DGA XMS standard relies, for its syntactic part, on the W3C XML *Schema* for its syntax, and, for its graphic semantics, on a form akin to UML while respecting the standard factual format of *Rational Rose* (.mdl). This approach implies that the code is generated from a graphic model (direct top-down engineering) and the metamodel's reverse engineering done from the code (bottom-up engineering). The goal is to use the XMS standard as a DSL graphic, within an MDE approach, leading to the best possible reuse of the model components managed in a Framework library dedicated to technical-operational simulations (see the project described in the following section). Since this framework takes care of the average design components in its field of activity, it enables the developer to concentrate on the functional added value which should be added to its development, and not on its technical resolution. The Framework answers the requirements defined by [KAD 05]. It implements concepts with a higher level of abstraction, which appear in the model. Thus, in the event of a change in the programming technology, one must and only need reconstitute the Framework's basic components. The effort of evolution is reduced to the Framework's components, but the essential of the business expertise is capitalized in the DSL.

The XMS standard corresponds to an M2 metamodel. It describes the usual concepts used in technical-operational simulations, and defines these concepts' rules of use. It constitutes a kind of grammar, guaranteeing the syntactic and semantic coherence of the model during its construction, for example by forbidding the association of graphic components when they cannot be linked together. Since the generated code can be manually modified by the developer, in order to add business expertise, there is a non-negligible risk that the XMS metadata initially featured in the code will eventually be incomplete.

To control this risk, the reverse engineering component verifies the XMS metadata, to see that all elements own the associated XMS metadata, and signal everything that might hamper the reverse engineering process. Building on the bootstrap mechanism of .net and the exploitation of the XML file which contains the commentaries of the source code, reverse engineering can find the components (classes, methods, attributes, etc.) as well as the associated metadata. A function of

validation of the meta-data is also provided, to verify that the editing operations of the source code have upheld the coherence of the XMS meta-data.

11.4.4. *Feasibility study of the models' capitalization and reuse*

The two previous projects focus on the interoperability of models and simulations, as well as their reuse, thanks to the implementation of a home structure and a methodological framework which complies with the requirements of object-driven engineering. In terms of reuse, capitalizing the models is crucial to fully profit from that approach. Thus, a third project has been launched, to create a library of models associated to the single reference platform, models which must be:

- *generic*: the models do not reference any particular hardware, system or equipment, but rather a class of tools, representing various devices. The entry data will define which particular device will be represented by an instance. The models will therefore be adaptable and reusable in a broad variety of contexts (e.g. technical-operational simulation, training war game) and constitute a basis for the build-up of finer or more complex models if necessary;
- *available*: these models are capitalized within a structure (digital and organizational), so as to facilitate access and reuse by the broadest public, within the ministry of defense or by the agents of the defense industry;
- *durable and evolutionary*: the models must be able to evolve according to the needs (integration of new modeled systems, upgrade of the models, etc.) and be able to adapt to a modification of their operating environment (home structure, operating system, etc.);
- *validated*: the models are validated through a standardized process. Precise elements will be needed to evaluate these models' applicability for each context of use;
- *coherent with the current needs*: the models will enable the modeling of new forms of operation (control of violence, humanitarian aid, counter-terrorism, etc.).

The models which will be capitalized within that library will eventually have to include three levels of modeling (e.g. granularity):

- *elementary*: the simplest level to describe a physical agent (e.g. punctual point to describe a plane);
- *intermediary*: an elementary level refined on certain parameters that are useful to modeling (e.g. taking into account the 3D attitude of a plane);
- *evolved*: technical-functional level which models the detail of the modeled system's operation.

Each of these models will feature description documents of function, technical, design, the source code, the associated technological data, and finally of validation. This set of knowledge, both business and technical, will be capitalized on to facilitate the models' reuse. Thus, the architect of a simulation will be able to choose, among the models featured in the model library's database, the models with the proper level of granularity, to assemble them (easily, thanks to a plug and play connection) and create the appropriate simulation.

We have previously pointed out that the UML language is rather adapted to documentation needs. It therefore looks like the perfect candidate for the constitution of documents about the functional, technical and design descriptions which go hand in hand with the models. However, taking into account the difficulty met by business experts in apprehending UML, it became obvious the models should be expressed in XML and follow the XMS metamodel. This is why a XMS-to-UML translating tool has been developed, to help business experts produce the necessary UML documents for the capitalization within the model library.

Both the approach and the tools have been experimented on existing models, with the aim of putting them through reverse engineering and capitalizing them according to the MDE methodology. The business experts were involved in the project from the start, which eased the change of their work habits, notably in terms of the documenting and traceability of their work. On the other hand, multilevel modeling, which allowed plug and play connections, did not provide any sufficiently convincing results for this type of reuse to be considered at the moment.

Once the set of models has been developed and validated, their organization is imperative to facilitate their exploitation and reuse. A taxonomy was therefore defined to structure the information of the library's generic and reusable models. To this taxonomy, an ontology based on OWL (*Ontology Web Language*) is added, specified by W3C, the international organization of standardization of web technologies. Such an ontology corresponds to a metamodel dedicated to the engineering of knowledge, by standardizing the semantic relationships existing between the capitalized information (for example, level 2 of the LCIM model, see section 11.2.5). It must be coherent with the XMS metamodel (for example level 4 of the LCIM model) in order to guarantee the best reuse of the models. When this chapter was written, further work was ongoing to deepen the taxonomic and ontological aspects for the project's advancement.

11.5. Conclusion and perspectives

The complexity of systems of systems must be controlled even before the beginning of their life cycle. Modeling and model-driven engineering, from analysis

to design, help reduce said complexity without losing sight of the fitness for purpose. The use of models and simulations helps reduce developing costs and delays, while providing early warnings against errors of design. This affirmation is all the truer since the models and simulations are interoperable and can be reused, as long as efforts are put in their verification, validation, and capitalization.

In software engineering or in modeling and simulation, experts agree that heightening the level of abstraction, such as is recommended by the MDA/MDE approach, definitely favors interoperability and reuse, by defining a model of reference of the field (level 2 metamodel) which guarantees the semantic cohesion between models. Since the business and technological aspects are separated, their relationship must be modeled through the transformation of existing models which will play the role of interface, a crucial point for systems engineering. To ensure the durability of investments, capitalization of verified and validated data constitutes the third complementary panel to achieve interoperability and reuse within the control of complexity. This third pillar requires an important effort of documentation and meta-information, which analysts, designers and developers must provide and update according to mastered engineering processes which reflect a certain maturity in their methodological approach.

Feedback attests to this methodology's feasibility, and to the expected profit. The MDA approach is tried-and-true in the case of general database applications, in which the implemented technologies, and the software and physical architectures are relatively standard. On the other hand, in the case of nested applications such as technical-operational simulations, an MDE approach with DSL Tools is more appropriate, considering the field's specificities and the associated simulation technologies. Difficulties are still left to overcome, notably in choosing the models' and metamodels' level of abstraction, the reduction of the system of systems' intrinsic complexity, but also the availability of mature MDA/MDE tools. Here are some leads on possible improvements so as to minimize such difficulties:

- the use of functional analysis might help the designer in specifying his needs in terms of modeling and metamodeling;
- the use of appropriate interview techniques, featuring open questions and following a candid attitude, may result in implicit and unexpected information (*unknown unknowns*, see [MUL 06]) thus reducing the uncertainty about the modeled complex system.

From an economical standpoint, model-driven engineering modifies the chain of value. Business enterprises ought not to worry about the production stages, since code generation can be automated. They ought to focus on their core business, in which they have a competitive advantage and through which they can bring their clients true added value. If the approach is industrialized, economies of scale may be

realized, or at least economies of scope for particular fields (see [GRE 04]). To achieve such profit, people must be trained to acquire the proper skills, and both their culture and state of mind must evolve. For example, the added value must be precisely analyzed, in terms of know-how between the physical or phenomenological equations subjacent to the models and technological performance data which help calculate precise numerical values with the help of equations. This added value is directly linked to possible ownership or industrial rights, which the producers of these technological data may well claim. The scope within which this claim of ownership applies should be very precisely defined, so as not to put unneeded restraints on the distribution of components through excessive and groundless protection. Model-driven engineering may bring practical solutions to this question insofar as the designer will take these aspects into account upstream of the life cycle: on the level of exchanged and manipulated data, one ought to distinguish, depending on the levels of generalization, what may be freely shared and what may not be without a specific agreement. To caricature, in an infrared camera, the general optical model must be generic, and it would be counter-productive to see it as a black box model which could not be freely accessed; on the other hand, data specific to the equipment, such as sensibility, may pertain to industrial protection. The interest of dissociating between generic and specific data is that it is always possible to exploit and use these models with the standard data of the open art, even if this means reaching *in fine* a digital performance less precise than it would have been through the use of exact data.

11.6. Bibliography

- [ALE 77] ALEXANDER C., ISHIKAWA S., SILVERSTEIN M., *A Pattern Language*, Oxford University Press, New York, 1977.
- [BEZ 01] BÉZIVIN J., GERBÉ O., “Towards a Precise Definition of the OMG/MDA Framework”, *Proceedings of Automated Software Engineering*, San Diego, United States, November 2001.
- [BEZ 04] BÉZIVIN J., “Sur les principes de base de l’ingénierie des modèles”, *RSTI – L’objet. Où en sont les objets?*, p. 145-156, 2004.
- [BRO 87] BROOKS F., “No silver bullet: essence and accidents of software engineering”, *Computer Magazine*, 1987.
- [DGA 97] DÉLÉGATION GÉNÉRALE POUR L’ARMEMENT, *Guide dictionnaire de la simulation de défense*, April 1997.
- [DGE 01] DIRECTION GÉNÉRALE DES ENTREPRISES, définition de la mission à la stratégie de normalisation pour la société de l’information (<http://www.telecom.gouv.fr/rubriques-menu/organisation-du-secteur/normalisation/interopabilite-349.html>).

- [DOD 94] DoD DIRECTIVE 5000.59-M, DoD M&S glossary, January 1994.
- [DOD 98] DoD C4ISR ARCHITECTURES WORKING GROUP, Levels of information systems interoperability, March 30, 1998 (<http://www.c3i.osd.mil/org/cio/i3/>).
- [GRE 04] GREENFILED J., SHORT K., *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*, Wiley, New York, 2004.
- [GRU 93] GRUBER T.R., *Towards Principles for the Design of Ontologies Used for Knowledge Sharing in Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic Publishers, New York, 1993.
- [ITE 04] *IT Expert*, issue n°51, September/October 2004.
- [JOH 98] JOHNSON M.V.R., MD KEON M.F., SXANTO T.R., “Simulation based acquisition: a new approach”, *Report of the 1997-1998 DSMC Military Research Fellows*, Defense Systems Management College Press, Virginia, United States, December 1998.
- [KAD 05] KADIMA H., *MDA – Conception orientée objet guidée par les modèles*, Dunod, Paris, 2005.
- [KAM 02] KAM L., Terminologie M&S dédiée aux tableaux de synthèse des réponses au questionnaire DCE-CAD, Document technique n° CTA/00350109/NTECH/001 du Centre Technique d’Arcueil, 20/03/2002.
- [LEM 94] LE MOIGNE J.L., *La théorie du système général*, PUF, Paris, 1994 (4th edition).
- [LEM 95] LE MOIGNE J.L., *Modélisation des systèmes complexes*, Dunod, Paris, 1995.
- [LUZ 03] LUZEAUX D., “Cost-efficiency of simulation-based acquisition”, *Proceedings of SPIE Aerosense’03, Conference on Simulation*, Orlando, United States, April 2003.
- [MCX 05] Glossary of the European program MCX, “Modélisation de la CompleXité”, Définition usuelle de la complexité, 2005 (<http://www.mcxapc.org/>).
- [MEN 02] MEINADIER J.P., *Le métier d’intégration de systèmes*, Hermès, Paris, 2002.
- [MET 05] Workshop A08 FROM THE EUROPEAN PROGRAM MCX, “Modélisation de la CompleXité”, Méta modélisation : de la modélisation conceptuelle aux formalismes de modélisations, 2005 (<http://www.mcxapc.org/>).
- [MIC 07] MICROSOFT CORPORATION, Livre blanc sur l’interopérabilité, April 2007, (further information at <http://www.microsoft.com/interop/>).
- [MIL 03] MILLER J., MUKERJI J., *MDA Guide version 1.0.1 of OMG*, June 12, 2003, available at: <http://www.omg.org/mda>.
- [MOR 77] MORIN E., *La méthode T.I.*, Le Seuil, Paris, 1977.
- [MUL 06] MULLINS J.W., *Discovering Unk-Unks: How to Learn What You Don’t Know You Don’t Know*, London Business School, London, 2006.

- [NON 00] NONGA HONLA J., Les fiches de lecture de la chaire D.S.O. du CNAM, 1999-2000 (<http://www.cnam.fr/lipsor/dso/articles/fiche/lemoine.html>).
- [OFTA 04] OBSERVATOIRE FRANÇAIS DES TECHNIQUES AVANCÉES., *Ingénierie des modèles : logiciels et systèmes*, series ARAGO 30, Tec & Doc, Paris, May 2004.
- [OFTA-1 04] JÉZÉQUEL J.M., BELAUNDE M., BÉZIVIN J., GÉRARD S., MULLER P.A., “Contexte et problématique, chapitre 1”, in *Ingénierie des modèles : logiciels et systèmes*, Observatoire Français des Techniques Avancées, series ARAGO 30, Tec & Doc, Paris, May 2004.
- [OFTA-4 04] BELAUNDE M., BÉZIVIN J., MARVIE R., “Transformations et modèles de plates-formes, chapitre 4”, in *Ingénierie des modèles : logiciels et systèmes*, Observatoire Français des Techniques Avancées, series ARAGO 30, Tec & Doc, Paris, May 2004.
- [OFTA-8 04] KAM L., L’HOSTIS B., LUZEAUX D., “Application aux simulations Défense, chapitre 8”, in *Ingénierie des modèles : logiciels et systèmes*, Observatoire Français des Techniques Avancées, series ARAGO 30, Tec & Doc, Paris, May 2004.
- [OTA 03] NATO ALLIED DATA PUBLICATION 34 (ADatP-34), NATO C3 Technical Architecture (NC3TA), Version 4.0, March 2003 (<http://www.nato.int/docu/standard.htm>).
- [OTA 07] NATO Glossary of standardization terms and definitions, AAP-6, OTAN, 2007.
- [PEL 98] PELISSIER C., *Unix. Utilisation, administration, réseau Internet*, Hermès, Paris, 1998 (3rd edition).
- [RAB 06] RABEAU R., Proposition d’une démarche de validation des simulations technico-opérationnelles utilisées comme une aide à la décision, Thèse de doctorat de l’université de Versailles Saint-Quentin – spécialité informatique, July 7, 2006.
- [RAB 07] RABEAU R., “Credibility of defense analysis simulations”, *20th International Conference on Software & Systems Engineering and their Applications, ICSSEA07*, Paris, December 2007.
- [SCO 03] SCOTT H., YOUNGBLOOD S., “A proposed model for simulation validation process maturity”, in *Proceedings of 2003 Spring Simulation Interoperability Workshop*, Orlando, April 2003.
- [SHA 02] SHALLOWAY A., TROTT J.R., *Design patterns par la pratique*, Eyrolles, Paris, 2002.
- [SIS 07] SISO GM V&V PRODUCT DEVELOPMENT GROUP, *Guide for Generic Methodology (GM) for Verification and Validation (V&V) and Acceptance of Models, Simulations, and Data – Reference Manual*, Version 1.0, 2007.
- [TOL 03] TOLK A., MUGUIRA J.A., “The levels of conceptual interoperability model”, in *Proceedings of 2003 Fall Simulation Interoperability Workshop*, Orlando, September 2003.

- [TOL 04] TOLK A., “Composable mission spaces and m&s repositories – applicability of open standard”, in *Proceedings of 2004 Spring Simulation Interoperability Workshop*, Washington DC, April 2004.
- [VOL 04] VOLLE M., A propos de la modélisation, Michel Volle’s website: <http://www.volle.com/travaux/modelisation2.htm>, March 5, 2004.

Chapter 12

Standardization in the Field of Systems and Systems of Systems Engineering

12.1. Introduction

This chapter will focus on standardization, in the field of systems and systems of systems engineering; it will study the standards relative to the engineering processes, the products and services, data exchanges, as well as the standards relative to the modeling of business processes.

This brings it within the scope of Chapter 10, “Methods and Tools for Systems of Systems Engineering”. Indeed, the methods and tools in system engineering may be subject to standards. Such is the case, for example, with UML and SysML for the description of systems, or ISO 15288 for the life cycle processes. Moreover, some methods and tools, such as system engineering tool infrastructures, may be compliant or compatible with a set of standards.

This chapter will try to provide as broad a list as possible, while keeping in mind that the standards are so diverse and numerous that such a list could not be exhaustive. The standards are, indeed, in constant evolution. Neighboring standards are harmonized, standards are created from new practices. Some standards, no longer coherent with the state of the art, or replaced by new standards, are declared obsolete.

Currently, such efforts at harmonization are very important within, and between, the various standardizing organizations. This makes it impossible to present a systematic, durable panorama of the standards pertaining to our subject. Moreover, while we are trying to present the standards as faithfully as possible, it is only a synthesized, simplified presentation, which does not try to substitute itself for the standards it references.

In such a context, this chapter aims to offer a representation, as simple as possible, which will underline the purpose of the various standardizing documents, the links between them and their relevance within the context of systems of systems.

We will start by highlighting the relevance of standardization in such a field, its critical aspect for the interoperability of systems and systems of systems. We will then give an outline of the various standards relative to operational processes, to engineering processes, methods and tools, to products and services. We will then look at the way those standards can be implemented and adapted in the context of systems of systems. We will then offer some rules for the design of a standard reference base of systems of systems, and to keep up-to-date with the evolution of standards in that field. Finally, we will offer an approach for the control of general contracting and project management during the implementation of those various standards.

12.2. Example of the importance of standards in the interoperability of systems and systems of systems

Chapter 4 presents Maier's definition of a system of systems, and the example of a system of systems, namely the service which provides automobilists with real-time information on traffic, providing alternative paths in case of traffic jams. We will use this definition and this example as a guiding thread to prove the relevance of standards for this system of systems' interoperability.

The service relies on data about the state of traffic, provided by each highway network's operators, integrated in a collecting center and finally transferred to the drivers' navigation systems.

Each network operator has designed his own supervision system. From an operational standpoint, each system is independent. Which means that the network operator, who owns the system, has designed it to fulfill a certain number of missions, and the system features all functionalities to do this, without calling on other systems. Each network operator is free to manage his system as he pleases, depending on his technical and financial priorities, the contractual links with his suppliers, upholding the managerial independence of his supervision system. This

supervision system informs him on the state of his network, whether it be accidents, particular weather conditions such as black ice, or the state of traffic. This knowledge helps manage resources and infrastructures, for example by opening more tollbooths in one way rather than the other during the principle vacation periods. This also helps keeping the users, clients of the highway operator, informed, so they can adapt their driving to the situation; for example, slow down in case of traffic jams or bad weather conditions.

The collecting center receives and integrates the information provided by the various systems, so as to create an integrated representation of the traffic situation. For such a thing to be possible, these pieces of information must have the same meaning, share compatible formats which the collecting center may translate, and the systems must implement communication protocols that are compatible with the collecting center's.

The information relative to the state of traffic must be transferred through the telecommunication systems to the drivers' navigation systems. Once again, a definite meaning, protocol and format are necessary. The navigation systems must translate this information and compile it with the geolocalization data, so as to exploit them in calculating the itinerary. The driver may, or may not, subscribe to regular updates of the navigation systems, taking into account the evolution of road and highway networks, such as the creation of crossroads, bypasses to avoid the center of towns, or changes in the roadways, such as one-way streets. If the driver hasn't subscribed to those updates, the service cannot be efficient.

Moreover, for drivers who travel outside national borders, and to achieve coherence on a European level, the diversity of the systems of regulations, signaling, etc., of the various countries, must be taken into account. This transborder aspect also has impacts on the interoperability of the telecommunication systems, as well as the meaning of the transferred data, taking the various measure units into account.

Taking into account the diversity of these various technical systems, how can the service be provided to the driver?

Standards help with the coherence and compatibility of the meaning and format of data, communication protocols and use of available common services. Those standards are relative to products and services. If many of these standards pertain to information technologies, others pertain to telecommunications, wired as well as Hertzian. But in another context, standards from other fields can be used.

In order to specify, design, qualify and maintain those systems, the partners, acquirers, suppliers, and subcontractors must exchange technical information, and use processes and activities which are both coherent and compatible with one

another. Enterprises which, depending on their activity, purchase, specify, design, realize, exploit, maintain, take apart and replace these products and services, must coordinate with one another in those various activities. Indeed, if an enterprise wants to make a system evolve, it must evaluate the impacts this might have on the systems with which the system is interfaced, and launch the evolution of these related systems as much as needed. The more coherent the processes and activities of the various enterprises, the easier the coordination will be. This coherence is ensured by a set of standards pertaining to engineering processes. These standards formalize the state of the art to lead the life cycle processes of the systems and the related processes. Those are engineering process standards.

The documents which describe the systems, and which the enterprises exchange within such coordination, must also have a clear, non-ambiguous meaning, for the benefit of each enterprise which will use them. A set of standards deals with technical data exchange formats and their meaning.

Moreover, Chapter 4 offers another example of a system, that of ticket booking, and more broadly that of a transportation company. Thus, when this company works with another to offer meals during train journeys, the technical systems are not the only ones concerned.

Both companies must coordinate their activity for the meals prepared by one company are served to the clients of the other company. Each partner's processes and activities, just like the roles and responsibilities of their colleagues, are concerned. Thus, the meals must be delivered before the train or the plane leave. Besides synchronizing the activities of the various partners to obtain an end-to-end service, those partners must also exchange data. Such as the number and types of meals booked by the client, and said client's requirements in terms of diet. These pieces of information are attached to other client-related data, such as the date and the hour of the trip, the train number, the car number and the booked seats, to make sure that the meals are delivered to the client who booked them. If the transportation company is dealing with several partners, depending on the towns, the rules of coordination and the exchange of data must be common to all partners, so the service can remain coherent from one end to the other.

These aspects depend on the organization and the adopted business processes. Standards have been elaborated to taking into account and organize processes, since they are automated by each partner's information systems. Those standards pertain to the business process. They do not intend to take into account all activities, or the operating of the organizations and enterprises beyond what pertains to the automation of business processes. These organizational aspects are studied in Chapter 4.

The next section will look at the available standards.

12.3. Standards used in the field of systems and systems of systems

Application	Relevant standard	Reference
Business process modeling notation	Notation for the modeling of business processes	BPMN
	Web services business process execution language	WS-BPEL
Exchange of business data	Electronic data interchange for administration, commerce and transport	EDIFACT
	Universal Business Language	UBL
	Ontology Definition Metamodel Specification	OWL
Descriptions and characteristics of systems, products and services	Architecture Framework and Enterprises Architecture	AF and EA
	UML Profile for DoDAF/MODAF	UPDM
	Software-intensive System Architecture	IEEE P1471
	Group of standards relative to Service Oriented Architecture in the IT field	SOA
	Unified Modeling Language	UML
	System Modeling Language	SysML
	Best practices in the management of IT services (Information Technology Infrastructure Library)	ITIL
	Software engineering - product evaluation	ISO/IEC 14598
	Software engineering - product quality	ISO/IEC 9126
	Information Security Management System	ISO 27001
	Common Criteria (CC) for Information Technology Security Evaluation	ISO 15408
Engineering process	System life cycle process	ISO/IEC 15288
	System engineering process	EIA 632
	Standard for system engineering implementation and management	IEEE 1220
	Software life cycle processes	ISO/IEC 12207
	Software measurement process	ISO/IEC 15939
	Human centered design processes for interactive systems	NF EN ISO 13407
Exchange Of Technical Data	System Engineering Data Exchange	ISO 10303-AP233
	Product Life Cycle Support Data Exchange	ISO 10303-AP239

Table 12.1. *Standards and their applications*

In this section, we will introduce five standard applications:

- business process modeling notation;
- exchange of business data;
- descriptions and characteristics of systems, products and services;
- engineering processes;
- exchange of technical data.

Those are synthetic descriptions. The following table presents the main standards for each of the five aforementioned types. It also features the most significant standards, and/or the ones who integrate or synthesize other standards.

These five applications structure the five following sections.

12.3.1. *Business process modeling notation*

Coordinating and synchronizing automated business processes is essential for organizations to collaborate. An example is found in the necessary coordination between the transportation company and the company which supplies the meals. These standards pertain to business processes and do not treat the technical means implemented to realize, support, these business processes. We will start this section with those standards.

12.3.1.1. Notation for the modeling of business processes

The business process modeling notation (BPMN) specification ([OMG 06a]) is a graphic notation standard used to describe a process's realization within a workflow. The adopted representation is the one of workflow diagrams.

This standard was developed by the BPMI (Business Process Management Initiative) and is maintained by the OMG (Object Management Group), following the merger of works relative to both organizations' business processes management in June of 2005. Other standards have been elaborated, or are being elaborated, on this theme, such as Semantics of Business Vocabulary and Business Rules, or Business Rules Markup Language.

Moreover, this standard is linked to the standard on Web services business process execution language (WS-BPEL) which we will study shortly. The latter specifies an executable format used by a machine to perform an automated business process implementing Web services technology. Therefore, the models elaborated with the BPMN standard and the WS-BPEL executable models may share

similarities. Both their semantics must tally. If the WS-BPEL standard specifies an executable format, the BPMN standard, on the other hand, specifies a format which can be used by humans (analysts, specialists, etc.) to model, specify and manage the business process models.

This standard relative to the modeling of business processes describes said processes with diagrams. It helps model these processes' structuring elements, such as:

- the events (the booking of a meal by a client is a trigger event);
- the activities, from the booking to the client's actual meal;
- the flows, which manage exceptions, such as a meal booked for a client with diabetes;
- the business data, when they impact the flows; the client's dietary restrictions must be recorded by the transport operator during the booking, and transferred to the meal supplier so he can prepare a meal which respects the client's requirements and dietary restrictions.

12.3.1.2. *Standard relative to the Web services business process execution language*

This standard (WS-BPEL) adds to the one we have just studied by offering a more dynamic perspective within a business process rationale, namely the processes implemented by different organizations, clients and suppliers, between one another, during their business relationships.

This standard was elaborated by the OASIS consortium (Organization for the Advancement of Structured Information Standards).

It is closely linked to the standard relative to WSDL (Web services description language), a technique for the implementation of service-oriented architectures, which we will study later on. It enriches it, insofar as "The interaction model that is directly supported by WSDL is essentially a stateless model of request-response or uncorrelated one-way interactions" ([OAS 07], Introduction).

This standards provides the necessary concepts for the description of business processes, including data-related behaviors, exceptional conditions, recovery sequences, temporal and logical links between activities (sequence, iteration, parallelism, etc.), and finally the context within which these processes take place.

This standard may be implemented in the field of e-commerce, for example in a digital marketplace, such as the one available on Amazon. This marketplace presents the offers from Amazon's partners which correspond to the product selected by the customer.

12.3.2. Business data exchange

Following the study of standards relative to business processes, namely what is done and how it is done, this section will study the data exchanged between partners, for example order forms, factors. The coherence of the exchanged data is essential for the organizations to share the same information and, *in fine*, collaborate. We will therefore study the standards relative to the exchange of business data.

12.3.2.1. Standard relative to the Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT)

This standard concerns the exchange of data between various enterprises and helps define a common meaning to use in such exchanges. This standard is old, dating back to 1994, and was elaborated within ISO, under the reference ISO 9735, “Electronic Data Interchange for Administration, Commerce and Transport” (EDIFACT). Among others, this standard was adopted by the United Nations, with the purpose of defining a common semantics for digital exchanges, under the title UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business).

This standard describes the messages exchanged between partners, through the interchange structures which feature segments, some of which are mandatory, some conditional. These segments go in pairs, with header and trailer segments.

A set of rules (exclusion, omission, truncation, repetition, message overlapping, etc.) are formulated.

This standard is old and new technologies based on XML (*extensible markup language*) make it obsolete. But it shows that the challenges of e-business are old and that the solutions implemented do evolve, bringing *de facto* evolutions in the related standards.

12.3.2.2. Universal Business Language

In the context of systems of systems, when the systems exchange data, beyond the fact that these data share the same code and the same format, they must also have an unambiguous meaning for the various systems which exchange them. The messages which we have treated in EDIFACT express this meaning in a rigid way. Technologies based on XML replace EDIFACT and offer increased flexibility.

Just like the WS-BPEL standard, which we have just studied, the universal business language is elaborated and published by the OASIS consortium. It defines the contents and meaning of e-commerce documents, such as order slips and invoices.

In the case of meals during train or plane trips, these would be the forms exchanged between the transportation company and its partners, forms which feature the description of the booked meals, the requests and diet requirements formulated by the client, the references of the train or the plane (train number, wagon number, seat number) on which the client will travel. This set of information, shared by the various partners, must have the same unambiguous meaning for each of them.

12.3.2.3. *Ontology definition metamodel specification*

Beyond the documents' structure and meaning, the definition of ontology can report meanings in a given field. This thematic pertains to semantics, and features concepts from philosophy and linguistics adapted to the field of information systems, such as ontology concepts, taxonomy, knowledge, etc.

The standard Ontology Definition Metamodel Specification was elaborated by the OMG ([OMG 06b]).

In the field of information systems, “an ontology defines the common terms and concepts (meaning) used to describe and represent an area of knowledge. An ontology can range in expressivity from a Taxonomy (knowledge with minimal hierarchy or a parent/child structure), to a Thesaurus (words and synonyms), to a Conceptual Model (with more complex knowledge), to a Logical Theory (with very right, complex, consistent and meaningful knowledge)” ([OMG 06b], p. 31). “The ODM is applicable to knowledge representation, conceptual modeling, formal taxonomy development and ontology definition” ([OMG 06b], p. 1).

We might envision the creation of an ontology dedicated to in-flight meals. What is an in-flight meal? What does it include? An in-flight meal ontology may report synonyms (“java” as a synonym for “coffee”) as well as business rules, for example food incompatibilities specific to religious beliefs.

12.3.3. *Descriptions and characteristics of systems, products and services*

While the standards we have just studied were about business processes and business data exchange, the standards we are about to see concern the systems, products and services. The latter concern the technical systems and are crucial for their interoperability, thus helping provide an end-to-end system. They help describe those systems', products' and services' internal layouts, and their position in relation to one another, namely their architecture. They help describe their structure and behavior, the ones of the data models, as well as their performances.

We will start this section by introducing the architecture frameworks and business architectures. Some sections will be about organizations and operational

processes, all of which might have been studied in the previous section. But these subjects will be treated from the standpoint of the architecture of products and services sustaining business processes, via the automation of said processes by the information systems. We will therefore study these subjects in the present section. When documents study operational processes, they quote standards relative to the description of business processes, which we have previously studied.

We will then give a brief presentation of a set of standards which concern business-oriented architectures, then a standard concerning the architecture of software-intensive systems.

We will continue with system and software structural and behavioral description models, and data models.

Later, we will introduce standards which help manage and control the operating systems', products' and services' performances.

We will then study software characteristics of quality.

And we will end with a set of standards which concern the security of information systems.

12.3.3.1. *Introduction to architecture frameworks and business architectures*

Architecture frameworks aim at expressing the various points of view held by an information system's stakeholders (user, acquirer, supplier, etc.), namely the operational processes, the services supplied by the system's components, their operation and the technical components.

These points of view help demonstrate, describe, an information system's architecture, such as the internal links between components, the links with the operational environment within which the system operates. This allows people to compare systems of differing origins, belonging to the same information system or system of systems, or compare different information systems so as to achieve better management, create links so they can exchange services and data and, *in fine*, adequately implement the enterprises' processes. This is the architecture of information systems, namely a coherent, harmonious design and integration of these systems' applications. The most recent versions of architecture frameworks also describe how those systems are acquired, to achieve a coherent acquisition of both the components and the global architecture.

These architecture frameworks, born in the field of information systems, have had their scope broadened to include every software-intensive system. As such, they have become the keystone of systems of systems' architectural models.

The first architecture framework was developed in 1987 by an IBM engineer, John Zachman. This architecture framework, before the arrival of Service Oriented Architecture (SOA) concepts, which we will study later on, was considered as a reference by all the other architecture frameworks, based on its global approach of architectural description.

The Zachman Framework is represented in a table (Figure 12.1), whose columns help raise the following questions about the information system, in order to facilitate its management:

- what (the data): lists important themes that are relevant to the organization, such as objects and business data;
- how (the functions): lists the processes run on those data by the organization's information system;
- where (the network): lists the sites on which the information system's processes run;
- who (the people): lists which of the organization's services are concerned by these processes;
- when (the time): lists the significant events which direct these processes;
- why (the motivation): lists the organization's goals and strategies.

The rows describe the following perspectives:

- the scope (contextual): describes the context of use, the strategic dimensions which have an impact on the operation of the information system;
- the enterprise model (conceptual): describes the prescribed¹ and specified business processes;
- the system model (logical): concerns the conceptual descriptions and representations, independent from the implementation, of the software system, which include, among other things, the conceptual data models;
- the technology model (physical): concerns the technical representations, linked to the implementation, for example physical database models;
- the detailed representations (out of context): treat the data which concerns the subcontractors.

1. In opposition with the actual activity performed by interface experts.

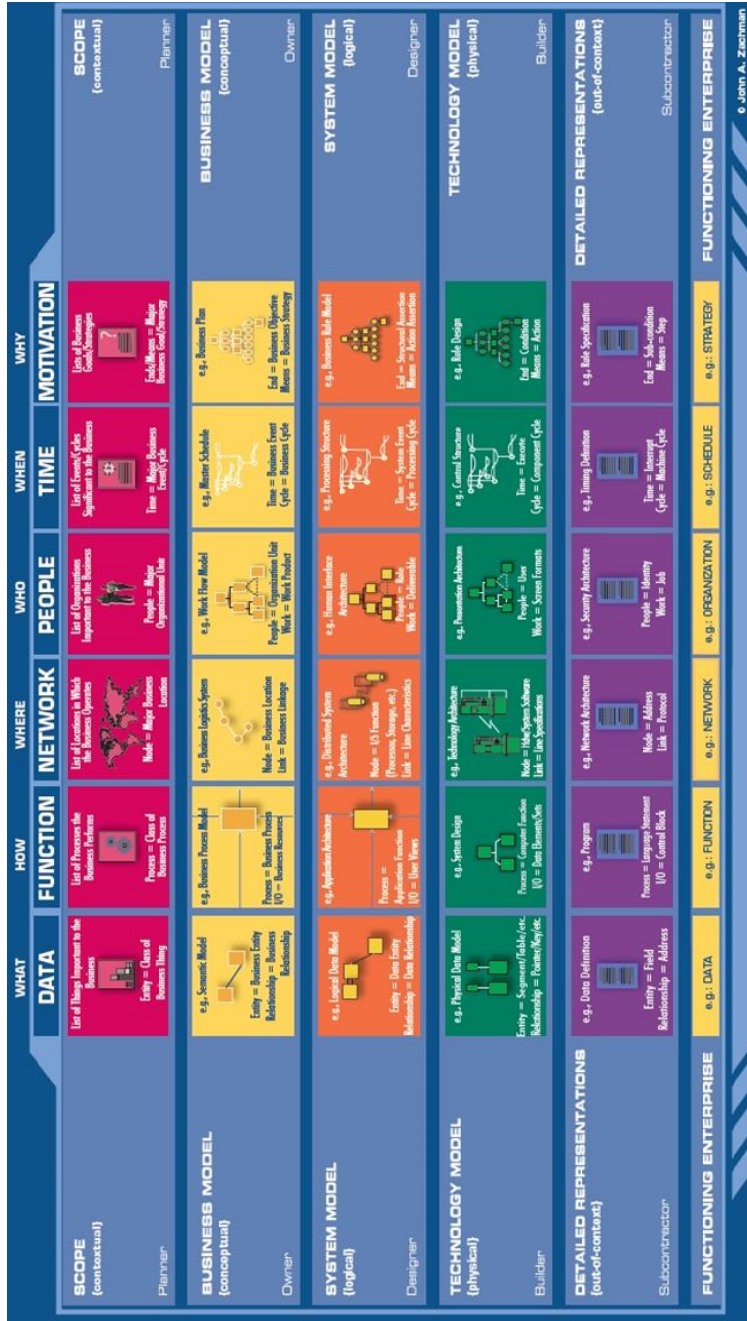


Figure 12.1. Structure of the Zachman Framework (source [ZIF 07])

Nowadays, many architecture frameworks exist. We may give the following, incomplete, list, which features the main frameworks:

- *Federal Enterprise Architecture Framework* (United States government);
- *Department of Defense Architecture Framework*, known under the acronym DoD AF (United States Department of Defense);
- Minister of Defense Architecture Framework, known under the acronym MoD AF, (United Kingdom Ministry of Defense);
- NATO Architecture Framework, known under the acronym NAF;
- The Open Group Architecture Framework, known under the acronym TOGAF.

These architecture frameworks are *de facto* standards. The American government demands that the description of information systems respect its architecture framework, respectively, FEAF for civilian activities, and DODAF for the army.

These architecture frameworks describe the information systems' architectures under various points of view, to which *views* are attached. The views feature a certain number of diagrams.

The MoD AF, for example, features the following views and diagrams:

- the all view viewpoint includes an overview and an integrated dictionary;
- the strategic viewpoint (StV), includes, among other things, an enterprise vision and the programs' phasing components;
- the operational viewpoint (OV) includes operational descriptions such as activity models, operational rules, operational concepts, information models;
- the system viewpoint (SV) includes descriptions such as functionality descriptions, resource constraints specifications and resource interaction specifications;
- the technical standards viewpoint (TV) includes technical standards such as operating systems or communication protocols (e.g. IP v6);
- the acquisition viewpoint (AcV) includes acquisition clusters and program timelines.

The architecture frameworks are enriched with each new version. Thus, OTAN's architecture framework's latest version (NAF v3) brings, in addition to the MoDAF views, a service viewpoint, to describe service-oriented architectures. Works are

currently led to add views adapted to human factors within architecture frameworks [HAN 07].

Moreover, software engineering tool editors have developed functionalities specific to their tools, for them to describe these architecture frameworks.

Business architecture is developed with the same purpose. The goal is to provide decision-making support tools for enterprise management. Its objective is, on the one hand, to optimize the organizational structure, the business processes and the enterprise's information system, in order to increase performance and productivity; and on the other hand, to align the information technologies on the strategic objectives of the enterprise, by reducing the gaps between the information system's current and expected states.

This consists of developing architecture frameworks to describe the various states of the reference architectures: current, intermediary, and target state, as well as the plan to migrate from one state to the other. In that way, business architecture helps achieve better traceability between the enterprise's strategy and the technologies needed to implement it².

This translates into documented descriptions of the four following categories, corresponding to the various states of the reference architectures:

- the business, which includes, among other things, the enterprise's policy, its goals, its organizational models, its business processes;
- the applications, which include the list of IT applications, their interfaces, the components interacting with other organizations, etc.;
- the information, which presents the data and metadata models (the metamodels describe the data used by the applications);
- the technical describes the technical platforms, the local or long range networks, the operating systems, the software infrastructures such as application servers.

The implementation of business architectures leans on a value-based approach, seeking quick wins, which brings all the more value to the organization.

A set of methods and tools is available in stores to implement those business architectures.

2. This is a structuring perspective on organizations. Part 1, Chapter 4 presents a broader range of dimensions which closely interact within a systemic logic.

These architecture frameworks, initially designed to describe information systems, are used to describe and model the enterprises' processes. Rigid formalization, based on finite state automaton models, might not be able to support the necessary adaptations to answer the organizations' contingencies, which modify their processes. Chapter 4 looked at organizations from another point of view. Moreover, business process modeling does not take into account all the scenarios, all the alternatives and all the contingencies. In such a case, deployed information systems do not comply with operational needs and constrain users to violate procedures. Recent Amalberti's article presents an example of such a violation in a case of automation medicine drug distribution [AMA 09].

Chapter 4 uses an example in the field of transport and ticket booking. Looking at this example in a MoDAF context, the views would be as follows:

- the all view viewpoint includes an overview and an integrated dictionary;
- the strategic viewpoint features, for example, the scheduling and organization of the ticket booking system's evolutions, taking into account new input/output devices and evolutions towards a service-oriented architecture which will replace a client/server architecture, whose deployment will be prepared through phasing components;
- the operational viewpoint features the price calculation rules, the price lists, the description of accounting procedures;
- the system viewpoint includes the booking system's functional descriptions, the descriptions of the interactions between the various sale systems, the technical management systems for these systems and the accounting center;
- the technical standard viewpoint describes the sale systems' operating systems, those of the technical management systems, the communication tools, including the ones taking care of the transfer of data between the technical supervision systems and the sale systems, etc.

12.3.3.2. *UML profile for DoDAF/MoDAF (USA Department of Defense and UK Ministry of Defense Architecture Framework)*

The OMG (Object Management Group) has designed a UML profile for DoD/MoD Architecture Frameworks. A UML profile is an adaptation of UML (Unified Modeling Language) specific to those architecture frameworks. We will later study another UML profile, and specify the meaning of UML. You may refer to that part to understand UML if you are not already familiar with the concept.

This standardizing document originated in two places: on the one hand, the USA DoD and UK MoD architecture frameworks, and on the other hand, the UML modeling language. This modeling language was designed and implemented in the

field of information systems. This twofold paternity translates into that document's conformity requirements. Thus, this UML profile must answer the needs of the stakeholders who develop information system architectures and systems of systems for the USA DoD and the UK MoD.

“UPDM will support the capability to ([OMG 07a], p. xxi):

- model architectures for a broad range of complex systems³, which may include hardware, software, data, personnel, and facility elements;
- model consistent architectures for system-of-systems down to lower levels of design and implementation;
- model service oriented architectures⁴;
- support the analysis, specification, design, and verification of complex systems; and improve the ability to exchange architecture information amongst related tools that are UML based and tools that are based on other standards.

The profile provides the modeling of operational capabilities, services, system activities, nodes [editor's note: nodes within networks], system functions, ports⁵, protocols, interfaces, performance, and physical properties and units of measure. In addition, the profile enables the modeling of related architecture concepts such as DoD's doctrine, organization, training, materiel, leadership & education, personnel, and facilities (DOTMLPF) and the equivalent UK Ministry of Defence lines of development (DLOD) elements” ([OMG 07a], p. 7).

UPDM concerns the following DoDAF and MOFDAF views:

- acquisition viewpoint;
- strategic viewpoint;
- operational viewpoint;
- systems viewpoint;
- technical viewpoint;
- all views.

3. Those systems are designed by human beings, and must not be mixed up with the notion of complex systems such as is treated by Edgar Morin.

4. References service oriented architectures in the field of software (SOA), used in particular for the Internet.

5. References the notion of port defined in modeling languages and methods, such as SysML.

UPDM is designed to extend to other views, including:

- services views;
- custom views;
- logistics views.

The UPDM profile was designed so as to be used within the paradigm of Model Driven Architecture. Transformations from and to the UPDM profile can be performed. The design of the UPDM profile can also extend to include:

- the interoperability and reuse of the UPDM profile;
- service-oriented architecture;
- additional views;
- artifacts, such as reports or graphs;
- model-driven architecture;
- the executable UPDM profile.

12.3.3.3. *Standard relative to the architecture of software-intensive systems*

This standard specifies the “recommended practice for the architectural description of software-intensive systems. A software-intensive system is any system where software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole. [...] The purpose of this recommended practice is to facilitate the expression and communication of architectures” ([IEE 00], p. 1). Those are methodological recommendations, independent from implemented architectural solutions.

This standard recommends taking into account the environment’s influence on the system, the stakeholders, the architectural activities in the systems’ life cycle, the uses of architectural descriptions, such as the communication between the acquirer and the supplier as part of contract negotiations. It also studies the necessary documentation, the coherence between the various descriptions, the architectural justifications.

12.3.3.4. *Standards relative to service-oriented architectures*

While the previous standard concerns the methodological recommendations, this set of standards relative to service-oriented architectures offers a framework for a given architectural solution. This set of standards was elaborated in the field of information systems, to enable the interoperability of computer applications within an open context, such as the Internet.

A set of recommendations has been elaborated on the subject, mainly by the W3C (*World Wide Web Consortium*; <http://www.w3.org/>), the OASIS (*Organization for the Advancement of Structured Information Standards*; <http://www.oasis-open.org/>), the IETF (*The Internet Engineering Task Force*; <http://www.ietf.org/>) and the WS-I (*Web Services Interoperability*; <http://www.ws-i.org/>). These organizations offer recommendations, but do not have the status of organizations for standardization, unlike the ISO on the international level, the CEN on the European level, and BSI, DIN, AFNOR, ANSI national standards bodies. They often are industrial consortiums. In some cases, these consortiums and organizations for standardization have joined together to design protocols so that the documents produced by these consortiums may be standardized. Such is the case between the OASIS consortium and ISO.

The scope of these recommendations, on that particular subject, is wide, from technical services and protocols and transport and communication services, to business services. This chapter will only succinctly study the highest level standards.

The notion of service is the keystone of this type of architecture. The use of these services must be described and published so the partners may refer to it.

Recommendations thus concern:

- the Web services description language, recommended by the W3C, describes, in XML, a public interface of access to a Web service, defining the way to communicate in order to use said service, the messages, the type of data needed for the service to operate, and the product or state provided by the service;
- the messages exchanged through the electronic business using extensible markup language (ebXML);
- the message exchange protocol, initially Simple Object Access Protocol, in XML, elaborated by the W3C, describes the Remote Procedure Call, and the message exchange protocol;
- A universal description discovery and integration registry, based on XML and elaborated by the OASIS, enables the storage of technical information and information on the service's supplier, such as his address or the name of the concerned business unit;
- the service mediator which enables a loose coupling between the producers and users;

- the security components include: transport layer security, which used to be called secure socket layer, XML signature and XML encryption, security assertion markup language, and XML key management specification⁶;
- the coordination of services, in order to provide business processes, within an enterprise, or business processes between several enterprises, for example in the case of e-commerce⁷;
- a transactional service management, to ensure the controlled updating of several databases.

The following table offers a synthetic list of the various standards elaborated by the OASIS and the W3C.

OASIS	W3C
DocBook	HTML
ebXML	XHTML
SAML	CSS
UDDI	DOM
LegalXML	XML
PKI	MathML
Relax NG	SVG
XDI	PNG
XRI	IETF
OpenDocument	O.P.I
WS-BPEL	

Table 12.2. *W3C and OASIS standards*

We describe the standards BPMN, ebXML and their common criteria in the following sections of this chapter.

6. Functionally, these components are detailed in the section on “common criteria for the evaluation of IT security.”

7. “Orchestration” and “choreography” are also used. We will look at this theme in further detail in the section dedicated to the modeling of business processes.

12.3.3.5. *Unified modeling language*

The Unified modeling language (UML) is created from the clustering of several object-oriented modeling languages, in the field of IT, dating back to the 1980s and the beginning of the 1990s. The first version of UML was standardized in 1997. The current version is version 2.1.1. This is the version we will briefly study.

The unified modeling language is a visual language used to specify, design and document software systems artifacts. This modeling language can be used with most methods and object components (object-oriented analysis and object-oriented design), in many fields of application and on many implementation platforms.

The unified modeling language is based on a common core, upon which the MOF (Meta Object Facility) and the profiles used to develop specific UML adaptations for specific uses also depend. An example is found in the aforementioned UPDM. We will not define the links between these various components, as it would deviate from the purpose of this chapter.

UML includes a certain number of diagrams, building on key-concepts, to represent the following software dimensions:

- the structural and static dimension, which describes the system objects, the relationships between them (“type-of” relationship, composition relationship, etc.), and the software components;
- the behavioral and dynamic dimension, which describes the behavior of objects called upon by external events, the interactions between objects, among other things, as finite state automata.

Moreover, UML can be extended and configured. It also features a metamodeling (MOF), a constraint language (OCL), and capabilities to design stereotypes and extend UML to design the aforementioned profiles.

12.3.3.6. *Systems modeling language*

The systems modeling language (SysML) was derived from UML and has recently been standardized by the OMG, in close collaboration with the INCOSE. Some elements of UML have been taken away, judged as less adapted to the systems, and other elements have been added.

The two dimensions, both structural and dynamic, are adapted to the context of systems in the following manner:

- the structural and static dimension, evolves to integrate structuring diagrams, including the diagram of block definition, of internal blocks, and the parametric diagram which introduces the dynamics’ continuity ([OMG 06c], p. 21);

– the behavioral and dynamic dimension is only superficially modified, with add-ons and evolutions affecting its concepts ([OMG 06c], p. 79).

The following table displays the similarities and differences between UML and SysML.

UML diagrams	SysML diagrams
Class diagram	Borrowed and adapted with the concept of blocks within the Block Definition diagram
Object diagram	N/A
Package diagram	Borrowed
Component diagram	N/A
Composite Structure diagram	Borrowed and adapted with the concept of blocks within the Internal Block diagram
Deployment diagram	N/A
Use case diagram	Borrowed as-is
Activity diagram	Borrowed and extended
Interaction diagram	N/A
Communication diagram	N/A
Interaction Overview diagram	N/A
Sequence diagram	Borrowed and adapted
Timing diagram	N/A
State machine diagram	Borrowed and adapted
N/A	Requirement diagram
N/A	Parametric diagram

Table 12.3. *Comparison between UML and SysML, similarities and differences*

To these two dimensions, SysML adds cross-cutting constructs, which both apply to the structures and the behaviors. These cross-cutting constructs concern notions of allocation, requirements and profiles and models libraries. The notion of allocation defines “a basic allocation relationship that can be used to allocate a set of model elements to another, such as allocating behavior to structure or allocating logical to physical components.” The notion of requirements “specifies constructs for system requirements and their relationships.” Finally, the notion of profiles and

models libraries defines “the approach to further customize and extend SysML for specific applications.” ([OMG 06c], p. 121).

12.3.3.7. *Good practices of IT service management*

The ITIL (Information Technology Infrastructure Library) lists, synthesizes and details the best IT service management practices, which a supplier provides to his clients.

ITIL’s purpose is to improve the quality of these services from end to end, from supply to support, including the financial aspects. This translates into a service level agreement (SLA) between the supplier and the service’s client. An SLA is a written agreement between an IT service supplier and his client(s), which defines these services’ objectives and both parties’ responsibilities. A true partnership must be developed between the IT supplier and the client, so a mutually benefiting agreement may be established, lest the SLA should quickly fall into discredit and the accumulation of complaints block any improvement of the service quality ([OGC 06], p. 29-30).

ITIL differentiates between, on the one hand, the service’s client, who pays and who is usually a manager, and on the other hand, the service’s user, who uses it daily and who the support team is in contact with. ITIL is based on a process-oriented approach and standardized within a set of books which present the best practices.

ITIL differentiates between the following processes:

- the design of the services and management processes (management of security, availability and contingencies, service levels, service reports, capability and finances);
- the deployment processes (deployment management);
- the control processes (management of configuration and assets, and of changes);
- the resolution processes (management of incidents and problems);
- the suppliers’ processes (management of customer relationships, and of suppliers).

The following schema represents the service management processes.

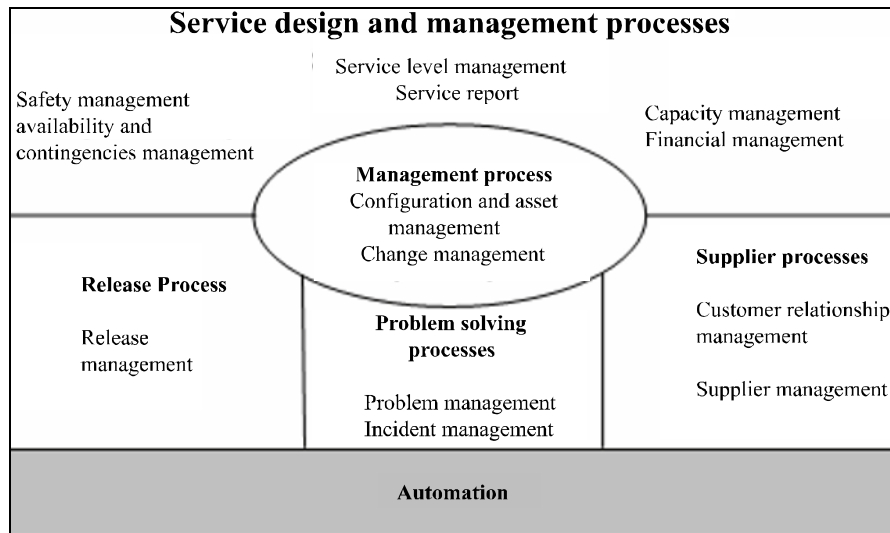


Figure 12.2. *Structure of IT services management processes*
(featured in [OGC 05], p. 9)

ITIL formed the basis of IT services standards, *Specification for IT Service Management* (BS15000) published by the *British Standard Institute*. It was reworked to produce the ISO standard, “IT Service Management” (ISO/IEC 20 000), published in 2005. If this last standard is less detailed than the ITIL documentation, it however references many other standards which specify the good practices in the field of software, such as the standards relative to “Software Life Cycle Processes, ISO/IEC 12207” and “Configuration Management Requirements, ISO 10007.”

12.3.3.8. *Standards relative to software quality*

The first of these two standards describes the characteristics of software quality. The second is about software quality evaluation and requirements. It studies the aspects relative to the process, as well as criteria of quality, measure and metrics. Its definition is therefore broad in scope, covering the standards relative to software characteristics, the subjects of this section, and those relative to the engineering processes which we will study in section 12.3.4.

12.3.3.8.1. Standard on the description of software quality characteristics (ISO 9126)

The standard ISO 9126 identifies the characteristics and subcharacteristics of software quality. This standard is split into four parts:

- the quality model, which identifies software characteristics: functionality, reliability, usability, maintainability, portability, which are themselves split into subcharacteristics;
- the elaboration of metrics for each characteristic and subcharacteristic.

12.3.3.8.2. Standard relative to software product quality evaluation and requirements

The SQuaRE standard (Software product quality requirements and evaluation) is currently elaborated within the ISO, and some of its parts are already available.

This standard features elements from ISO 14598, relative to software quality evaluation processes, which we studied in section 12.3.4.2.3 and ISO 9126, relative to software quality.

12.3.3.9. *Standard relative to the common criteria for information technology security evaluation (ISO 15408)*

The standard relative to the “Common Criteria for IT Security Evaluation” is a “basis for evaluation of security properties of IT products and systems” ([ISO 05b], p. 1).

“IT products or systems should perform their functions while exercising proper control of the information to ensure it is protected against hazards such as unwanted or unwarranted dissemination, alteration, or loss. The term IT security is used to cover prevention and mitigation of these and similar hazards [...] ISO/IEC 15408 can be used to select the appropriate IT security measures and it contains criteria for evaluation of security requirements” ([ISO 05b], p. 8).

This standard is structured into three parts, respectively:

- general concepts and IT security evaluation principles;
- security functional requirements, security assurance requirements;
- security functions evaluation criteria.

12.3.4. *Engineering processes*

These standards describe all the processes which industrial partners must implement to specify, design, develop, exploit, maintain and withdraw systems or software, products or services.

We distinguish between, on the one hand, standards which concern system engineering processes, and on the other hand, standards which describe software engineering processes. If a software component belongs to a system, which might include, among other things, electronic, electrical, mechanical, or thermodynamic components, software engineering processes have many similarities with system engineering processes.

The acknowledgement of human operators, whether they be users, administrators, or entrusted with the technical system's maintenance, is framed by a standard relative to human-centric design processes, whose scope covers computerized systems.

We will now provide a succinct presentation of these various standards on system engineering processes, software engineering processes, and human-centric design processes.

12.3.4.1. *Standards relative to system engineering processes*

In the case of system engineering processes, several standards address the same subjects, with different perimeters. This is explained by the evolution of practices in a field which is still young, and still in construction.

The standards address:

- system life cycle processes (ISO/IEC 15288: 2002);
- processes for engineering a system (ANSI/EIA-632-1998);
- Standard for Application and Management of the Systems Engineering Process (IEEE Interim Standard 1220-1994).

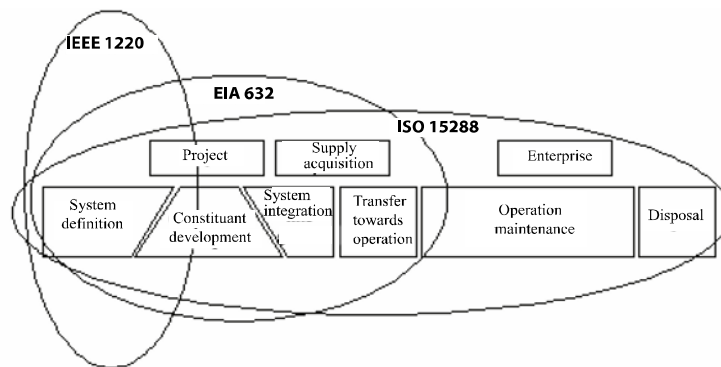


Figure 12.3. *Respective scopes of the three standards, ISO 15288, EIA 632 and IEEE 1220 (featured in AFIS, [AFI 08])*

The three standards describe the business processes of system engineering, through the definition of which activities must be performed, and those activities' products. These descriptions are formulated as activity-specific requirements (what must be done, not how it must be done). Their scopes are different, and their coverage grows deeper as their scope grows smaller; in this way, they complement each other, even more so since the most recent versions of the oldest standards take into account the publication or evolution of the other standards.

These three standards share the same approach about the system's definition and the products to provide. The system includes, on the one hand, the end-products and end-services, which constitute the system and perform its operational functions, and on the other hand, the enabling products and services, which feature the components necessary to the realization of the processes which are applied to the end-products and end-services, but which do not contribute to the system's functionalities. For example, such enabling systems are the products and services used for testing, development, production, training, dismantling. The system is broken down into subsystems, which themselves include end-products and end-services, and enabling products and services on their level, all this in a recursive manner.

We are going to describe, synthetically, these three standards, and define their respective scopes.

12.3.4.1.1. Standard relative to a system life cycle processes (ISO/IEC 15288)

This standard was designed to answer the difficulties met during the management and development of systems which combine hardware, software and human operators, and to compensate for the absence of harmonization and the integration of the disciplines which contribute to those activities, in the fields of science, engineering, management and finances. For example, an organization may use this standard to establish an environment for the processes which must be implemented at its core. An infrastructure which features methods, procedures, tools, and qualified personnel, supports these processes.

This standard provides a common framework to describe the life cycle of systems designed by human beings, and establishes specific processes and the associated vocabulary. It defines all the processes which must be implemented all through a system's life cycle, from its specification to its dismantlement. The acquirer and supplier of the studied system must implement these processes, depending on their respective roles and responsibilities. It also gives the definition of a system, a process, and the related concepts.

These processes, whose mapping is indicated in Figure 12.4, are split into four categories:

- the agreement processes define the activities linked to client-customer relationships, including the acquisition processes which must be implemented by the acquirer, and the supply process which must be implemented by the supplier;
- the enterprise processes define the activities which participate to system engineering management and support on the level of the enterprise;
- the project processes define the management processes through which the projects are controlled;
- the technical processes define the activities linked to the transformation of the client's needs into technical solutions, and to the maintenance of the corresponding products, all through the system's life cycle, from the expression of need to the disposal process.

12.3.4.1.2. Standard relative to processes for engineering a system (EIA 632)

Like the ISO/IEC 15288 standard, which we have just seen, the EIA 632 standard describes engineering processes. But, unlike the former, which covers both the level of the enterprise and the project, as well as the technical level, from the expression of need to the disposal process, EIA 632's scope is smaller, and does not cover exploitation, support and disposal. This standard is older, dating back to 1995 for its first version, while ISO/IEC 15288 was published in 2002. The latter actually takes EIA 632 into account.

The standard relative to processes for engineering a system lists thirty three requirements. Said requirements are structured into five requirement groups relative to the processes which must be implemented. Each group features a certain number of activities.

These five groups, shown in Figure 12.5, concern:

- technical management processes;
- acquisition and supply processes;
- system design processes (definition of needs and definition of the solution);
- product development processes (development, transition, etc.);
- technical evaluation processes (system analysis, verification, validation).

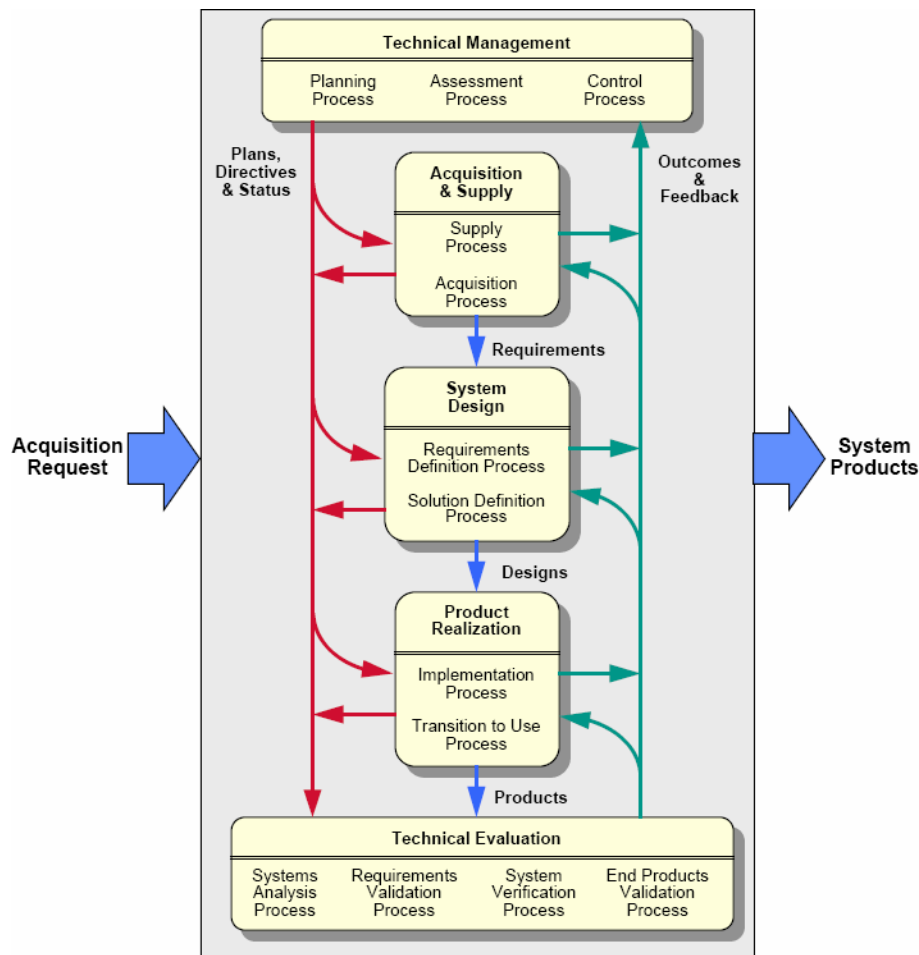


Figure 12.5. EIA 632 standard processes ([EIA 98], p. 4)

12.3.4.1.3. Standard for the application and management of the systems engineering process (IEEE 1220)

The new version of the standard, published in 2005⁸, takes the ISO/IEC 15288 standard into account and complements it by offering technical processes of systems engineering, with added details concerning the transition from the expression of need to the definition of the solution. These standards may be simultaneously implemented. Like the EIA 632 standard, its scope is limited. It covers the early

8. The first edition dated back to 1994.

stages of projects. The IEEE 1220 standard features fourteen requirements relative to system engineering processes (planning, modeling and prototyping, etc.), defines six stages of a system's life cycle (definition of the system, preliminary design, detailed design, fabrication, assembly, production and support, simultaneous engineering) and identifies eight engineering processes, displayed in Figure 12.6 (requirement analysis, requirement validation, functional validation, synthesis, design verification, system analysis and control).

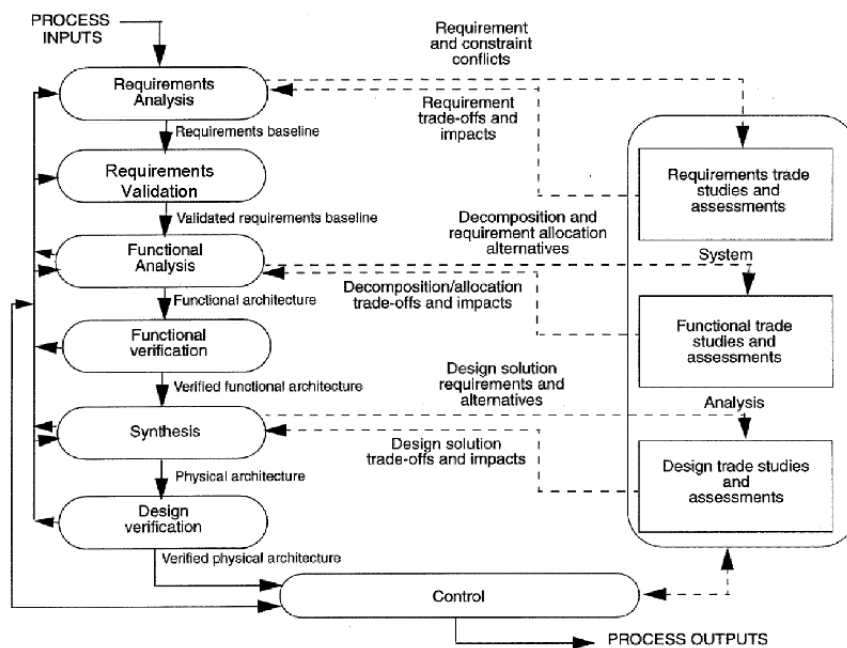


Figure 12.6. Engineering processes detailed within the IEEE 1220 standard ([IEE 05], p. 12)

12.3.4.2. Standard relative to software engineering processes

We have provided an overview of the standards which describe the processes to implement in system engineering. We will now look at the processes which must be implemented in software engineering, as well as the standards relative to the software's measurement process.

12.3.4.2.1. Standard relative to software life cycle process

Like the three system engineering standards we have just mentioned, this standard is about the life cycle process of software (ISO/IEC 12207).

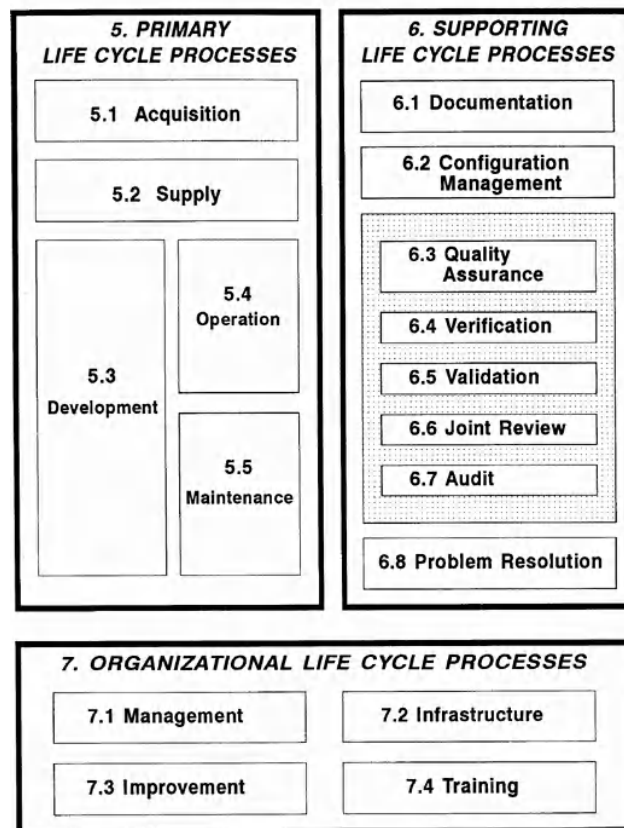


Figure 12.7. *Structure of the standard on software life cycle process ([ISO 95], p. 7)*

Processes are shared with the ISO/IEC 15288 standard, but their organization differs.

The ISO/IEC 12207 recognizes three types of processes:

- primary life cycle processes, including: acquisition, supply, development, operation and maintenance;

- supporting life cycle processes, including: documentation, configuration management, quality assurance, verification, validation, joint review, audit, and problem resolution;
- organizational processes, including: management, infrastructure, improvement, and training.

12.3.4.2.2. Standard relative to software measurement process

The ISO 15939 standard, “Systems and software engineering – Measurement process”, defines a measurement process for software development. The purpose is to identify the activities and tasks which are needed to identify, define, select, apply and improve software measurement within a project, or a dedicated structure of the enterprise. This standard therefore completes the software life cycle process standard we have just mentioned. It defines activities such as the establishment and maintenance of measure capabilities, the measure’s planning and realization, and its evaluation.

12.3.4.2.3. Standard relative to the evaluation of software products (ISO/IEC 14598)

The ISO 14598 standard, “Software engineering – Product evaluation”, describes the software’s evaluation process. This standard complements the previous one. It details the processes of evaluation, and evaluation support, which helps establish evaluation requirements, their specification, the design and execution of an evaluation plan, project management, as well as evaluation report models.

12.3.4.3. *Standard relative to human-centered design processes for interactive systems*

Few standards are dedicated to processes in the field of ergonomics. The standard relative to human-centered design processes for interactive systems [EN 99] was elaborated in the field of software and, from its generic character, may be applied to any software-intensive system.

Like other standards relative to engineering processes, it defines the principles to implement, the activities to lead, and provides an example of an evaluation report structure, to increase ease of use.

The principles are:

- the users’ active participation, and the clear comprehension of the requirements linked to the user and the task;
- the appropriate distribution of functions between the users and the system;
- the iteration of design solutions;

- multidisciplinary design.

The activities are:

- understand and specify the context of use;
- specify the requirements related to the user and the organization;
- produce design solutions;
- evaluate the solutions based on the requirements.

The document “Ergonomics of human-system interaction” [ISO 02c], offers a set of usability methods and rules which help select the most adequate methods depending on criteria such as “the choice of usability methods based on the life-cycle process”, and “the constraints brought by the project’s environment”, the “user characteristics”, the “characteristics of the task to accomplish”, the “used product” and finally, the “capabilities required by the designer or the evaluator”.

Links have been delineated between, on the one hand, system engineering processes, and on the other hand, human-centered design processes, to facilitate the implementation of ergonomic approaches within system engineering processes [RUA 04].

12.3.5. Standards relative to the exchange of engineering data

The managerial independence of a system of systems means that the programs of each of its constitutive systems possess their own processes, methods and tools. It is therefore necessary for the exchanges of technical data between the general contractors and project managers, to be comprehensible for all. The standards relative to the exchange of engineering data define a common language, on the level of the lexicon, the syntax and the semantics. These standards are grouped within a set, “Standard for the Exchange of Product model data”, more well-known under the acronym STEP. We will first introduce this set, before studying the Application Protocol 233 (AP233), which treats the exchange of system engineering and design data. Finally, we will introduce Application Protocol 239, known as PLCS for product life cycle support.

“STEP provides a collection of tools and models to facilitate the transfer of data between engineering tools. This is achieved through an open, generic format that reduces the number of specific interfaces” ([SCO 02], p. 1).

This standard defines a set of requirements relative to the standard format for the exchange of product data. The goal is to allow the exchange of product data between

various trades' applications, offering different data models, for example the exchange of geometric data between two CAO applications. The application fields which gave rise to the standard are the CAO, the GPAO, and Product Data Management. More broadly, this standard allows the exchange of product data between partners (client - supplier), following a standard format. This standard offers a definition, which can be interpreted by a computer, of a product's physical and operational characteristics, as a collection of standardized models and data, which represent reference models for the applications which treat the product's data.

The standard is broken down into several parts, grouped together according to their field of definition. Each field of definition is presented, with its featuring parts. The fields of definition are:

- the description methods concern the data models' design methods and rules, based on the EXPRESS language, to create a complete, formal and non ambiguous description of the static product data model, through notions of objects, relationships and conditions;
- the implementation methods treat the methods needed for the standard's implementation, and concern:
 - the structure of physical exchange files (sequential file),
 - the use of XML to represent the data,
 - the standard data access interface (SDAI),
 - the links with languages to implement the SDAI, the target languages are C, C++, Java and IDL (CORBA environment);
- the integrated resources are the core of STEP, describing the product data with EXPRESS, general data, independent from the trades, such as the product description, the identifier, the name, the text, the units of measure, the dates, as well as the generic frameworks, such as mathematical constructions, digital topology, digital analysis, the shape of objects in 3D;
- the Application Resources specializes the resources integrated within a specific technical context, such as part 42, which treats geometric and topologic representations and is specific to the field of engineering drawing;
- the application interpreted constructs specify the management of geometric data, the representation of edge-based wireframes, shell-based wireframes, etc., independently from the tools which implement said representations;
- the application protocols treat the representation of data models produced within a specific application and include:

- an application activity model, an operational model of the application which supports the produced data model,
- an application reference model, the produced data model which describes the object classes and their relationships,
- the application interpreted model, the standardized product data model, namely the representation of the reference model, built on the object representation model of the integrated resources,
- an application model which may be broken down into Units of Functionality, enabling partial representations of the data models;
- the abstract test suite is designed to validate the data models' formats.

The ISO 10303-233 standard pertains to application protocols, which is why it is more widely known under the name AP233. Its purpose is to define requirements for the representation of engineering data and system design. AP233 provides a neutral mechanism, capable of describing products all through their life cycle. This mechanism is not only adapted to the exchange of files with a neutral format, but is also a basis for the implementation and sharing of the produced databases, as well as a basis for archiving. This application protocol defines the context and scope of the system engineering and design data ([ISO 07]).

The AP233 requirements treat system engineering and design, its modeling, its structure, program management, decision-making support, stakes management, project management, risk management, system behavior, based on states, and finally, the system's operational behavior.

While AP 233 treats systems engineering, AP 239 is about product life cycle support. This application protocol holds significant importance for systems with high life expectancy, such as nuclear power stations, planes, ships, etc.

This application protocol defines sets of data exchange, relative to, among other things, support product arborescence, product failures, support activities and tasks, maintenance planning, operational feedback.

These two protocols are still in the design stage and have not yet been published. The information presented in this chapter aims to give an idea of what these standards might be, but they might very well evolve before their publication.

12.4. Application and adaptation of system engineering standards in the context of systems of systems

All the standards we have studied concern systems which feature hardware and software, or information systems. Not one of those standards is dedicated to systems of systems, even if some of them, which treat business data exchanges between organizations, are completely relevant to this issue. The application of these standards to systems of systems requires, on the one hand, adequate standards to be selected, and on the other hand, most probably, their adaptation.

The standards must be selected according to the characteristics of the system of systems, on a programmatic and technical level.

On the programmatic level, the system of systems' life cycle, and the individual life cycles of its constitutive systems, must be specified, so as to identify the related dependencies and constraints. One must taken into account the various management techniques when several general contractors and/or project managers are stakeholders in the development of a system of systems. Can the systems evolve? If they do, can they do so in a synchronous fashion? The coherence of technical processes, such as configuration management, is a determining factor in the success of a program of system of systems.

On the level of technical processes, this translates into the application, on the one hand, of engineering process standards, and on the other hand, of standards relative to the exchange of technical data, for the disciplines and engineering methods and practices which contribute, directly or indirectly, to the integration of systems within the system of systems. Links and interdependencies exist between the adopted architecture solutions and the engineering methods and practices which must be used to implement these solutions, with consequences on the harmonization of engineering methods and practices through the exploitation of process standards.

These programmatic aspects may have other non-negligible consequences on the technical aspects. If the systems cannot evolve synchronously, their ascending compatibility must be ensured. This means that the systems must simultaneously support services or data exchanges coming from several different versions. This further complicates interface management, from a technical point of view, and requires superior configuration management for each program whose systems contribute to the system of systems.

On the technical level, the type and nature of exchanges between systems, to provide a service on the level of a system of systems, determine which standards must be implemented. Thus, data exchanges between systems via telecommunication tools, such as computer assisted navigation, require the use of

standards relative to telecommunications, for protocols, as well as standards on data exchanges, for semantics.

In some contexts, certain standards could be pertinent, but wouldn't address all the problems. Thus, the standards relative to software characteristics may be used as a basis and enriched to take into account the specificities of systems of systems, for example through the addition of characteristics such as survivability or resource autonomy, when one of the system of systems' systems might be subjected to malevolent attacks, and calls on limited resources to provide its services. A drone may enjoy an important, and yet restricted autonomy. It must go back to its base for refueling.

The same thing applies to the standards relative to the systems' life cycle processes. These standards concern the systems. What makes them pertinent to systems of systems?

The engineering processes of a system of systems do not radically differ from a system's engineering. However, the characteristics specific to systems of systems lead to non-negligible evolutions of system engineering processes. We identify, on the one hand, a new process which is not featured in standard ISO/IEC 15288, the reverse engineering (retroanalysis/retrodesign) process; on the other hand, an activity, which, while not having the status of a process, is still crucial, dedicated to the system of systems' reliability and resilience capability; and finally, five existing processes whose weight and perimeter undergo large evolutions. These processes concern: information management, configuration management, architecture design, integration, and validation.

Indeed, insofar as many systems of systems are constituted of pre-existing systems, we must check that all the information needed for the integration of these systems into the system of systems is available. If this is not the case, a process of retrodesign and retroanalysis must be implemented. This process is not featured in standards IEEE 1220, EIA 632 or ISO/IEC 15288. They apply to unitary technical systems, with the implementation of a top down approach. In this context, reverse engineering does not pertain to good practices, but is a stopgap to perform system maintenance, for example in the case of a third-party application maintenance contract.

Conversely, in the field of systems of systems, when they are built on pre-existing systems, real situations and the difficulties which they may involve must not be ignored. The poor documentation, or lack thereof, must be taken into account. The reasons for such poor documentation are varied: technical (the documents have not been crafted), contractual (the system's general contractor has not, for example, acquired these documents, for varying reasons). Since a system cannot be integrated

within a system of systems without prior knowledge of the services it provides, the quality of these services, its interfaces, etc., reverse engineering must be performed. This activity must be tracked, and resources allocated to it. Its products are documents, models, which help describe the system, perform analysis, such as impact analysis and simulations. These documents and models must be integrated within the system of systems' reference base, which we are about to detail. This reverse engineering must be managed as a project, just like the other activities. If, by chance, these documents do exist and are available, the coherence between the physical system's architecture and the documentation will be evaluated, before it is included in the system of systems' reference base. For small disparities between the recorded configurations and the actual configurations are frequent.

Systems', and systems of systems', reliability and more broadly their resilience capability, are all the more critical because, as we will see, the dynamic integration of, and the inability to perform a complete and systematic validation of, the system of systems, do create problems. This is not a new technical process, but an activity which must be implemented within every technical process. This cannot be reduced to the analysis of failure modes, even if said analysis must be scrupulously led. It must take all identified threats into account. Moreover, it must analyze, from the start, a set of dysfunctional scenarios, using "what ... if ...?" questions. Once more, this kind of analysis is not sufficient in itself, and would lead to non-negligible risks; it must be furthered by systemic analysis (situations which feature accidents without errors or failure) [HOL 06]. Lastly, it must lead to a resilient system of systems architecture, able to support unplanned, unforeseeable situations, confine anomalies or failures to curb waterfall effects, but also able to go back to a trusted behavior after an unplanned event, and finally, able to be adapted, by the users, to the context's evolutions [RUA 09a], [RUA 09b].

Information management takes on a major, even critical, role. The goal is no longer to be the project's memory, which, through excess, becomes static. This process evolves to bear the system of systems' reference base, as well as everything related to the specifications and standards which concern the interfaces between the system of systems' systems, whether it be the maintenance of the service repository, the pivotal models, the semantic coherence of exchanges, or the maintenance of the documents which describe the physical interfaces.

This process is systematically called on by every technical process, so that the documentation is coherent with the actual, physical system, all through its life cycle. It is therefore called on by the change and configuration management process, to perform impact analysis as soon as the evolutions of one of the system of systems' systems lead to modifications in the services and/or the exchange data and/or the interfaces. Finally, it provides configuration management and publishes the service

repository's updates, the pivotal models and the interface documents aimed at every stakeholder involved in the project.

This reference base is maintained, either through first rank general contracting or project management, and hence from a system of systems level, when they exist, or by an *ad hoc* instance, when the system of systems is implemented by a coalition of independent agents. Such is the role of NATO, which maintains a reference base, represented by the STANAG. Moreover, this reference base must be consistent with the reference bases of the various systems which constitute the system of systems, reference bases which are maintained by each system's general contracting entity, or project manager. Several reference bases must therefore be maintained, and data exchanged, between them, which follows the standards on technical data exchange, such as AP 233.

Because of the systems of systems' evolutionary development, configuration management takes on a crucial importance and becomes a critical process. Like what is currently done in the field of software engineering, it is enriched with a change management process (Unified Change Management/Configuration Change Management from Rational/IBM). This helps implement processes which are more dynamic, more reactive and agile, when faced with the evolution of needs as much as the evolution of the system of systems' architecture, and the architecture of its constitutive systems, while following the approach with rigor.

The architecture design process is completely different. The aim is to weave, within a circular approach, both top down and bottom up, the allocation links between system of systems requirements and systems' services, as well as with the new services created by the integration of systems within the system of systems. We can see that the architecture design and integration processes ought not to be separated. Moreover, unlike a unitary system, the architecture is not fixed. At the very least, it evolves with each evolution of the systems which constitute the system of systems. In some contexts, it is much more dynamic; the constitutive systems may join or leave the system of systems on the fly. Such is the case with the *peer to peer* data exchange systems. The architecture of a system of systems must be designed to limit, confine, failures and system anomalies, to curb any waterfall effect. The goal is to design an architecture which furthers the resilience of the system of systems, and therefore overlaps with the aforementioned reliability process.

Like the architecture design process, the integration process for systems of systems is much more dynamic than with unitary systems. Each evolution of the system of systems, or of one of its constitutive system, whether they concern these systems' services and the data they exchange, or their interfaces, the addition of a

new system or the disposal of an existing system; each iteration entails a new integration, which, while it may be partial, can never be cancelled or reduced.

This architecture, which is not fixed, and this integration, which may be partial, give rise to validation problems. For, if the validation of a unitary system tends to be as complex and systematic as possible, such cannot be the case with a system of systems. This does not mean there is no validation, nor that this validation is reduced. Everything which can be validated must be validated. For the rest, we must make sure, via, on the one hand, the architectural design, and on the other hand, its reliability and resilience capability, that the system of systems will properly behave in any unidentified circumstance. Finally, without a confirmed general contracting and project management, with common agreements, the validation of a system of systems loses some of its contractual strength.

If the processes we have just studied see their scope widen, the implementation process will carry less weight. Indeed, the implementation is ensured on the level of the systems which constitute the system of systems, and its role is therefore less important on the level of the system of systems. However, the system of systems will still be affected, since the concerned system will eventually be integrated within said system of systems.

The pivotal services and models must include metadata in order to treat every aspect necessary for the exchange of said services and data (for example what pertains to their versions). Finally, the system of systems' constitutive systems do not necessarily evolve synchronously. This is caused, among other things, by their managerial independence. They each have their own objectives and constraints, which impact their evolution, be it their rhythm, perimeter, scope, content. The constraints might concern the budget or the regulations.

In such a context, the systems must be able to exchange services and data with systems which feature different configurations, older or more recent versions. This implies a bottom up compatibility. But, insofar as a system exchanges services and data, not with one system, but with several different systems, each with its own services and data, each exchange must take this aspect into account. The systems agree on these aspects when, and if, they elaborate contracts between one another. These aspects then belong to the context of the following exchanges. They are implicit and do not need to be recalled for each exchange. On the other hand, if no context has been established and the exchanges are *ad hoc*, on the fly, these aspects must be made explicit for every exchange.

12.5. Implementation of standards in the context of systems of systems

The implementation of standards in the context of systems of systems is not akin to a “one-shot” situation. A standard reference base must be designed, adapted to systems and systems of systems, taking into account the application and adaptation of the standards we have just studied. The purpose is not to make everything uniform by prescribing a method, a tool, a modeling language. It is not, either, to standardize everything. The components which will be integrated within the system of systems must be standardized.

This reference base will evolve, integrating newly published standards, the updates of old standards, and deleting the obsolete standards. For example, in the last few years, standards relative to the WiFi technology have been published (IEEE 802.11). A process of production and regular updating must be designed for this standard reference base. This process must also include a monitoring activity, to keep informed about the latest developments. The people for whom systems of systems hold major operational or financial stakes may participate to standardization groups, so as to be as close to monitoring as possible and contribute to the elaboration of these standards. This reference base must also be transferred to the people who need it, e.g. project leaders, engineers. This reference base must also be defined as something “to apply” and not something “applicable”, from a contractual point of view. The implementation of this reference base within programs and projects which contribute, in one way or another, to one or more stage of the systems’ and the system of systems’ life cycle, must also be verified and validated. Finally, feedback must be collected on the projects, to correct, modify and enrich this standard reference base. Since this standard reference base is implemented within various projects, configuration management must be performed to track its progress and help reach pertinent, opportune decisions.

How can this reference base be structured and organized? The answer depends on your field of activity, and a universal solution, adapted to every situation, is therefore not possible. Still, some simple rules can apply to every case.

Building on architecture views, we must identify the systems which contribute to the system of systems, the links between them, in terms of service and data exchange, matter, energy or data flows, and physical interfaces. The development of an N^2 diagram helps highlight, from a static point of view, these exchanges, the couplings between systems ([AUT 07, MEI 98, MEI 02]). Each cell of this N^2 diagram corresponds to system exchanges. Everything is to be gained from the standardization of these exchanges, leaning on the field’s standards, for example through the implementation of a service-oriented architecture in the field of software.

The N² diagram may also be used to identify these systems' stakeholders, project managers and general contractors [AUT 08]. The stakeholders must exchange technical data about these systems and implement common engineering processes to help the technical management and the project management of the system of systems. Finally, when a system evolves, whatever the reason, the N² diagram helps identify the stakeholders managing the interface systems, and pilot these systems' evolutions.

The specific norms and standards of technical architecture view may be used to manage all the standards of the reference base that is being developed. It offers a framework to characterize and manage the standards which are implemented within the reference base.

12.6. Conclusion

While not claiming to give an exhaustive review of the standards pertaining to the field of systems of systems, this chapter has tried to present the challenges of standardization, and provide elements to make these standards evolve, and a way of implementing these standards within the context of systems of systems.

12.7. Acknowledgements

We would like to heartily thank Alain Dohet, in charge of the DGA division on "Systems of systems", who presented Jean-René Ruault with the opportunity of contributing to the works of standardization. We also heartily thank Denis Champart, director of the *Centre de normalisation de défense* (Defense standardization center), and Marc Mouly, former deputy director of the *Centre de normalisation de défense*, for their advice on the presentation of the standards relating to systems of systems.

We would also like to extend our thanks to Frédéric Autran, Jean-Luc Garnier, Patrice Micouin, who have largely contributed, through discussions or proofreading, to this document's creation.

12.8. Appendix A. Standard relative to business process modeling

The standard which deals with the modeling of business processes characterizes three types of submodels, depending on the use which is made of them:

- Private (internal) business processes “are those internal to a specific organization and are the types of processes that have been generally called workflow” ([OMG 06a], p. 10).

- Abstract (public) processes represent “the interactions between a private business process and another process or participant. Only those activities that are used to communicate outside the private business process, plus the appropriate flow control mechanisms, are included in the abstract process. Thus, the abstract process shows to the outside world the sequence of messages that are required to interact with that business process.” ([OMG 06a], p. 11).

- Collaboration (global) processes depict “the interactions between two or more business entities. These interactions are defined as a sequence of activities that represent the message exchange patterns between the entities involved [...] The collaboration process can be shown as two or more abstract processes communicating with each other. With an abstract process, the activities for the collaboration participants can be considered the “touch-points” between the participants” ([OMG 06a], p. 11 and 12).

Several types of diagrams can be created within or between these submodels, including ([OMG 06a], p. 12):

- high-level private process activities (not functional breakdown);
- detailed private business process, including as-is or old business process and to-be or new business process;
- detailed private business process with interactions to one or more external entities;
- two or more detailed private business processes interacting;
- detailed private business process relationship to Abstract Process;
- detailed private business process relationship to Collaboration Process;
- Abstract Process relationship to Collaboration Process.

The participants, who control the business processes, may have different points of view about the way these processes behave, such as internal or external perspectives. But the standard does not specify any graphic representation that would reproduce those points of view, leaving it to modeling tool editors. Finally, the standard was designed to be expended by the analysts who create the models, and by the modeling tools.

The aim is to describe all the tasks which will have to be performed by a set of agents, the development timelines, validation modes, logical and temporal links between tasks, transition elements from one task to the other, so as to automate and digitally manage the enterprise's processes.

The elements of notation of activities can be grouped into the following basic categories ([OMG 06a], p. 15):

- the flow objects define the behavior of the business processes:
 - events;
 - activities;
 - gateways;
- the connecting objects connect the flow objects to each other or to other information:
 - sequence flow;
 - message flow;
 - association;
- the “swim-lanes” offer two ways of grouping objects:
 - pools;
 - lanes;
- artifacts are used to provide additional information about the process:
 - data object;
 - group;
 - annotation.

The core modeling elements depicted by the notation of activity flows and business processes are as follows:

- Activity, a “generic term for work that company performs. An activity can be atomic or non-atomic (compound). The types of activities that are a part of a Process Model are: process, sub-process and task.” ([OMG 06a] p. 16). Moreover, an activity may be characterized by a set of attributes, such as status (active, ready, cancelled, completed, etc.), inputs, outputs, loop, activities compounded into sub-processes, etc. ([OMG 06a], p. 50-53).

– An event “is something that “happens” during the course of a business process. These events affect the flow of the process and usually have a cause (trigger) or an impact (result) [...] There are three types of events, based on when they affect the flow: start, intermediate, and end.” ([OMG 06a], p. 16). Moreover, those events can be characterized, “e.g. Message, Timer, Error, Rule [...]” ([OMG 06a], p. 19).

– The “gateway is used to control the divergence and convergence of Sequence Flow. Thus, it will determine branching, forking, merging, and joining of paths. Internal Markers will indicate the type of behavior control.” ([OMG 06a], p. 16). There are several types of gateways, e.g. inclusive, exclusive, parallel, forking, merging, event-based, etc. ([OMG 06a], p. 20).

– A “sequence flow is used to show the order that activities will be performed in a process.” ([OMG 06a], p. 17). There are various types of flows: normal, uncontrolled, conditional, default, exception, etc. ([OMG 06a], p. 10).

– A “message flow is used to show the flow of messages between two participants that are prepared to send and receive them.” ([OMG 06a], p. 17).

– An “association is used to associate information with flow objects.” ([OMG 06a] p. 17).

– A “pool represents a participant in a process.” ([OMG 06a], p. 17). “A participant can be a specific business entity or can be a more general business role (e.g. a buyer, seller, or manufacturer” ([OMG 06a] p. 87).

– A “Lane is a sub-partition within a Pool and will extend the entire length of the Pool, either vertically or horizontally. Lanes are used to organize and categorize activities.” ([OMG 06a], p. 17).

– “Data Objects are considered Artifacts because they do not have any direct effect on the Sequence Flow or Message Flow or the Process, but they do provide information about what activities require to be performed and/or what they produce.” ([OMG 06a], p. 17).

– The group/grouping of activities (a box around a group of objects for documentation purposes) does not affect the Sequence Flow. “The groupings can be used for documentation or analysis purposes. Groups can also be used to identify the activities of a distributed transaction that is shown across Pools.” ([OMG 06a], p. 17).

– “Text Annotations (attached with an Association) are a mechanism for a modeler to provide additional information for the reader of a BPMN Diagram.” ([OMG 06a], p. 17).

12.9. Appendix B. Standard relative to the Web services business process execution language

The standard on web services business process execution language features the following concepts:

- “Business processes include data-dependent behavior” ([OAS 07] “Introduction”);
- “the ability to specify exceptional conditions and their consequences, including recovery sequences” ([OAS 07] “Introduction”);
- “long-running interactions include multiple, often nested units of work, each with its own data requirements” ([OAS 07] “Introduction”).

“WS-BEPL defines a model and a grammar for describing the behavior of a business process based on interactions between the process and its partners. The interaction with each partner occurs through Web Service interfaces, and the structure of the relationship at the interface level is encapsulated in what is called a partnerLink. The WS-BPEL process defines how multiple service interactions with these partners are coordinated to achieve a business goal, as well as the state and the logic necessary for this coordination. WS-BPEL also introduces systematic mechanisms for dealing with business exceptions and processing faults. Moreover, WS-BPEL introduces a mechanism to define how individual or composite activities within a unit of work are to be compensated in cases where exceptions occur or a partner requests reversal” ([OAS 07] “Introduction”).

Without giving a systematic view of the WS-BPEL standard, here are some of the major points treated:

- the definition of business processes;
- the definition of partner links and their types;
- the variable properties;
- data handling, including variables, usage of query and expression languages;
- the links between activities, metaphorically called “conversations”;
- the basic activities, such as invoking Web Service Operations, providing Web Service Operations, updating partner links, signaling internal faults, delaying execution, adding new activity types, propagating faults, immediately ending a process;
- structured activities, such as sequential processing, conditional behavior, repetitive execution, parallel and control dependencies processing and processing multiple branches;

- the context surrounding these activities, the scope, which defines their perimeter and boundaries, and includes message exchange handling, error handling, compensation handlers, event handlers, termination handlers, and isolated scopes.

This standard ends with appendixes dedicated to security considerations.

12.10. Appendix C. Ontology definition metamodel specification

This appendix concerns the specification of an ontology definition metamodel.

“ODM-based ontologies can be used to support:

- interchange of knowledge among heterogeneous computer systems;
- representation of knowledge in ontologies and knowledge bases;
- specification of expressions that are the input or output from inference engines” ([OMG 06b], p. 1).

In this context, “an ontology is a specification of a conceptualization for some area; There may be distinct ontologies representing different conceptualizations of the same domain. Ontologies may also differ due to the cost-benefit trade-offs associated with different specifications” ([OMG 06b], p. 14).

Several perspectives, distributed among two subcategories, model-centric perspectives on the one hand, and application-centric perspectives on the other hand, can be implemented and pondered on a scale going from least to most authoritative, from most volatile to most stable. This helps identify which requirements apply to which models depending on how closely the applications are coupled to each other, the evolution rhythm of the models, their authoritative, standardizing, character. For example, “the structure represents published rules of engagement, required for interoperability, that can only be revised by authorized agents in a well-publicized manner” ([OMG 06b], p. 14). On the other hand, in other contexts, the models rapidly evolve, even during the execution of the services which call on them.

“At the core are two metamodels that represent formal logic languages: DL (Description Logics) – non-normative – and CL (Common Logic), a declarative first-order predicate language. While the heritage of these languages is distinct, together they cover a broad range of representations that lie on a continuum ranging from higher order, modal, probabilistic, and intentional representations to very simple taxonomic expression” ([OMG 06b], p. 31).

“There are three metamodels that represent more structural or descriptive representations that are somewhat less expressive in nature than CL and some DLs. These include metamodels of the abstract syntax for RDF Schema, OWL, and Topic Maps” ([OMG 06b], p. 31).

Finally, two other metamodels, used in software engineering modeling, are added: UML, and Entity Relationship diagramming. “Three UML profiles have been identified for use with the ODM for RDF, OWL, and Topic Maps” ([OMG 06b], p. 31).

These metamodels, on which the ODM standard is based, come from different sources (W3C, ISO). We will now give a succinct presentation of these metamodels.

The Resource description framework (RDF) “is a language standardized by the World Wide Web Consortium for representing information (metaknowledge) about resources in the World Wide Web. It builds on a number of existing W3C standards, including XML (Extensible Markup Language), URI (Uniform Resource Identifier)” ([OMG 06b], p. 33). The resource description framework is a data model used to produce formal descriptions of Web resources and their metadata. This enables the annotation and description of unstructured documents (such as audio or video files), provides an interface with structured applications (such as a database), and enables interoperability between structured and unstructured data. A resource description framework is structured into three elements: subject, object and predicate. The subject represents the resource to be described (a video document, for example). The predicate denotes the type of property that can apply to that resource. The object denotes a data or another resource. The RDF semantics can be translated into a first-order logic formula. Request languages have been developed for RDF.

“The Web Ontology Language (OWL) is a semantic markup language for publishing and sharing ontologies on the World Wide Web. It builds on RDF [...] for describing properties and classes: among others, relations between classes (e.g. disjointedness), cardinality, equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes” ([OMG 06b], p. 61).

“Common Logic (CL) is a first-order logical language intended for information exchange and transmission over an open network [ISO 24707]. It allows for a variety of different syntactic forms, called dialects, all expressible within a common XML-based syntax and all sharing a single semantics [...] In general, first order logic provides the basis for most commonly used knowledge representation languages, including relational databases; more application domains have been formalized using first order logic than any other formalism – its meta-mathematical properties are thoroughly understood. [...] First order logic can also provide the formal grounding for business semantics [...] There has been significant effort to leverage

CL as the first order logic basis for the semantics of business vocabulary and business rules (SBVR) specification” ([OMG 06b], p. 93).

The topic map metamodel ([OMG 06b], p.117: topic maps represent a coherent set of topics), with associations and occurrences. A subject is characterized by its name, its occurrences and its roles in associations with other subjects. Occurrences act as a gateway between a subject and its resources. Those resources, such as databases or online documents, feature information about the subject. The topic maps have the same structure as a semantic network. Topic maps are standardized within the document ISO/CEI 13250: 2003 – topic maps.

12.11. Appendix D. UML profile for DoDAF/MODAF (USA Department of Defense and UK Ministry of Defense Architecture Framework)

This appendix will describe the six views of the UPDM profile, and their contents:

- “The acquisition view package contains UML stereotypes that assist the modeler in developing the views defined in the MODAF Acquisition Process. These views support the acquisition program dependencies, timelines and DLOD status to inform program management. They have been introduced to describe programmatic details, including dependencies between projects and capability integration across the all the DLODs. These Views guide the acquisition and fielding processes” ([OMG 07a], p. 18).

- The all views viewpoint describes “the setting in which the architecture exists [which] comprises the interrelated conditions that compose the context for the architecture. These conditions include doctrine; tactics, techniques, and procedures; relevant goals and vision statements; concepts of operations; scenarios; and environmental conditions” ([OMG 07a], p. 31). “An Architecture Description is a collection of Architecture Views, which includes Operational View, Systems View, and Technical Standards View, Acquisition View, and Strategic View. These views are integrated with each other” ([OMG 07a], p. 35). The system’s description includes several architecture views, which are necessary to fully describe its architecture.

- The operational views “support the description of the tasks and activities, operational elements, and information exchanges required to accomplish DoD missions. The operational view package also includes elements for the identification of the operational nodes, assigned tasks and activities, and information flows required between nodes” ([OMG 07a], p. 63).

- The strategic view package includes elements such as “capability and configuration, effects and the relationship between capabilities and the resources required to realize them” ([OMG 07a], p. 136).
- The systems view package contains UML stereotypes which “support the description of systems and interconnections providing for, or supporting, DoD functions. The systems view Package also includes elements to identify the systems resources that support the operational activities and facilitate the exchange of information among operational nodes” ([OMG 07a], p. 165).
- The technical standard views “support the description of the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements. The technical standards view package includes a collection of the technical standards, implementation conventions, standards options, rules, and criteria organized into profile(s) that govern systems and system elements for a given architecture” ([OMG 07a], p. 238).

The acquisition view include:

- “the AcV-1 System of Systems Acquisition Clusters (AcV-1) describe how acquisition projects are organizationally grouped in order to form coherent acquisition programmes” ([OMG 07a], p. 21);
- “the AcV-2 System of Systems Acquisition Programme (AcV-2) provides an overview of a programme of individual projects, based on a time-line. It summarizes, for each of the projects illustrated, the level of maturity achieved across the DLODs at each stage of the CADMID lifecycle, and the interdependencies between the project stages” ([OMG 07a], p. 21);
- “AcVCustom Custom Acquisition View” ([OMG 07a], p. 21);
- “A delivered capability indicates which Capabilities are to be delivered at the successful completion of a specific project and which project will deliver specific Capabilities” ([OMG 07a], p. 23);
- “The Defence Logistics Operation Centre, has the attributes to show the status of a particular capability in terms of whether it is available. These are the attributes used in the pie charts to illustrate project maturity” ([OMG 07a] p. 24); the lifelines are:
 - training,
 - equipment,
 - logistics,
 - infrastructure,
 - organization,

- doctrine/concepts,
- information,
- personnel;

– “A Milestone is a set of key dates for the systems analyzed in the architectural description. Milestone can apply to capabilities and system groups. Projects and forecasts can be associated with a Milestone [...] For software, this could indicate the release of a new version of a suite of applications. [...] It can also be an event in a Project by which progress is measured” ([OMG 07a], p. 26-27);

– “A project is a plan by an organizational unit to procure systems related to operations scheduled for a finite period of time” ([OMG 07a], p. 29).

“All views” include:

– *AV-1: Overview and Summary*: “The Overview and Summary Information provides executive-level summary information in a consistent form that allows quick reference and comparison among architectures. AV-1 includes assumptions, constraints, and limitations that may affect high-level decision processes involving the architecture” ([OMG 07a], p. 37);

– *AV-2: Integrated Dictionary*: “this view is the Integrated Dictionary that contains definitions of all the key concepts and model elements in an architectural model” ([OMG 07a], p. 38);

– *AVCustom: All View Custom View*: “a user-defined view that applies to the all view” ([OMG 07a], p. 39);

– without going into further details, and without identifying each single one, it also includes concepts such as concerns, goals, doctrine, etc.

The operational views include:

– *OV-1: High Level Operational Concept*: “high-level graphical/textual description of operational concept [which] provide detail of the operational performance attributes associated with the scenario/use case” ([OMG 07a], p. 118);

– *OV-2: Operational Node Connectivity Description*: “description of operational nodes, connectivity, and information exchange needlines between operational nodes” ([OMG 07a], p. 119);

– *OV-3: Operational Information Exchange Matrix*: “information exchanged between operational nodes and the relevant attributes of that exchange” ([OMG 07a], p. 120);

- *OV-4: Organizational Relationships Chart*: these charts describe the “organizational, role, or other relationships among organizations” ([OMG 07a], p. 121);
- *OV-5: Operational Activity Model*: these models describe the “capabilities, operational activities, relationships among activities, inputs, and outputs; overlays can show cost, performing operational nodes, or other pertinent information” ([OMG 07a], p. 122);
- *OV-6: Operational Activity Sequence and Timing Descriptions*: these diagrams represent a certain number of information, such as the operational rules model, which constrain the operations; the state transition description, which identifies business process responses to events; and event-trace description ([OMG 07a], p. 123);
- *OV-7: Logical Data Model*: “documentation of the system data requirements and structural business process rules of the operational view” ([OMG 07a], p. 124);
- *OVCustom: Custom Operational View*: “a user-defined view that applies to the operational view” ([OMG 07a], p. 125);
- without going into further detail, or identifying each single one of them, it also includes concepts such as policy, capability resources, process flow controls, etc.

Strategic views include:

- *StV-1: Capability Vision*: “outlines the vision for a capability area over a particular time frame” ([OMG 07a], p. 158);
- *StV-2: Capability Taxonomy*: this diagram “provides a structured list of capabilities and sub-capabilities (known as capability functions) that are required within a capability area during a certain period of time” ([OMG 07a], p. 159);
- *StV-3: Capability Phasing*: this diagram “captures the planned availability of capability at different points in time” ([OMG 07a], p. 160);
- *StV-4: Capability Clusters*: this diagram “provides a means of analyzing the main dependencies between capabilities” ([OMG 07a], p. 161);
- *StV-5: Capability to Systems Deployment Mapping*: this diagram “shows the planned capability deployment as systems, equipment, training, etc. and their interconnection by organization / period of time” ([OMG 07a], p. 162);
- *StV-6: Capability Function to Operational Mapping*: this diagram “describes the mapping between capability elements and operational activities that can be

performed by using them and thereby provides a link between capability analysis and activity analysis”⁹ ([OMG 07a], p. 163);

- *StVCustom: Custom Strategic View*: “a user-defined view that applies to the strategic view” ([OMG 07a], p. 164);

- without going into further detail, and without identifying each one of them, it also includes concepts such as capability, capability configuration, effects, capability resources, etc.

System views include:

- *SV-1: Systems Interface Description*: this diagram describes “systems nodes, systems, and system items and their interconnections, within and between systems nodes” ([OMG 07a], p. 192);

- *SV-2: Systems Communications Description*: this diagram describes “systems nodes, systems, and systems items, and their related communication laydowns” ([OMG 07a], p. 193);

- *SV-3: System-Systems Matrix*: this diagram describes the “relationships among systems in a given architecture; can be designed to show relationships of interest, e.g. system-type interfaces, planned vs. existing interfaces, etc.” ([OMG 07a], p. 194);

- *SV-4: Systems Functionality Description*: this diagram describes the “functions performed by systems and the system data flows among system functions” ([OMG 07a], p. 195);

- *SV-5: Operational Activity to Systems Function Traceability Matrix*: this diagram performs the “mapping of systems back to capabilities or of system functions back to operational activities” ([OMG 07a], p. 195-196);

- *SV-6: systems data exchange matrix*;

- *SV-7: Systems Performance Parameters Matrix*: this diagram describes “performance characteristics of systems view elements for the appropriate time frame(s)” ([OMG 07a], p. 197);

- *SV-8: Systems Evolution Description*: this diagram describes the “planned incremental steps for migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation” ([OMG 07a], p. 198);

- *SV-9: Systems Technology Forecast*: this diagram describes “emerging technologies and software/hardware products that are expected to be available in a

9. We are dealing with a homonym, which does not correspond to the analysis and activity led by interface experts. Those are prescribed activities, not actual activities.

given set of time frames and that will affect future development of the architecture” ([OMG 07a] p. 199);

- *SV-10: Systems Functionality Sequence and Timing Descriptions* (SV-10A, 10B and 10C): this diagram “identifies constraints that are imposed on systems functionality, [...] identifies responses of a system to events, [...] identifies system-specific refinements of critical sequences of events described in the operational view” ([OMG 07a], p. 200);

- *SV-11: Physical Schema*: this diagram describes the “physical implementation of the Logical Data Model entities, e.g. message formats, file structures, physical schema” ([OMG 07a], p. 201);

- *SVCustom: Custom Systems View*: “a user-defined view that applies to the systems view” ([OMG 07a], p. 202);

- without going into further detail, and without identifying each one of them, it also includes concepts such as services, communication links, data exchange, etc.

The technical standards view include:

- *TV-1: Technical Standards Profile*: this diagram “collates the various systems and standards that implement and constrain the choices that can be made in the design and implementation of an architectural framework” ([OMG 07a], p. 347);

- *TV-2: Technical Standards Forecast*: this diagram describes emergent standards and their potential impact on all views for a given architecture, for a given period of time.

- *TVCustom: Custom Technical View*: “a user-defined view that applies to the technical standards view” ([OMG 07a], p. 246);

- without going into further detail, and without identifying each one of them, it also includes concepts such as standards, etc.

12.12. Appendix E. Standard relative to software-intensive systems architecture

The Recommended Practice for architectural description of software-intensive systems (IEEE 1471) includes:

- “a conceptual framework for architectural description” ([IEE 00], p. 4);
- “architectural description practices” ([IEE 00], p. 8), including, among other things, the following elements:

- “AD identification, version, and overview information” ([IEE 00], p. 8);

- “identification of the system stakeholders” ([IEE 00], p. 8).

The conceptual framework describes the following:

- “Architectural description in context [...] The environment, or *context*, determines the setting and circumstances of developmental, operational, political, and other influences upon that system. The environment can include other systems that interact with the system of interest, either directly via interfaces or indirectly in other ways. The environment determines the boundaries that define the scope of the system of interest relative to other systems. A system has one or more stakeholders. Each stakeholder typically has interests in, or concerns relative to, that system” ([IEE 00], p. 4).

- “The stakeholders and their roles. Stakeholders have various roles with regard to the creation and use of architectural descriptions. Stakeholders include clients, users, the architect, developers, and evaluators. Two key roles among stakeholders are the acquirer (or client) and the architect. The architect develops and maintains an architecture for a system to satisfy the acquirer” ([IEE 00], p. 6).

- “Architectural activities in the life cycle. “Architecting contributes to the development, operation, and maintenance of a system from its initial concept until its retirement from use. As such, architecting is best understood in a life cycle context, not simply as a single activity at one point in that life cycle. Architecting is concerned with developing satisfactory and feasible system concepts, maintaining the integrity of those system concepts through development, certifying built systems for use, and assuring those system concepts through operational and evolutionary phases. Detailed systems engineering activities, such as detailed requirements definition and interface specification and the architecting of major subsystems are tasks that typically follow development of the system architecture” ([IEE 00], p. 6). Four system architecture scenarios are identified:

- “architecture of single systems” ([IEE 00], p. 6),
- “iterative architecture for evolutionary systems” ([IEE 00], p. 7),
- “architecture of existing systems” ([IEE 00], p. 7),
- “architectural evaluation” ([IEE 00], p. 7).

- Uses of architectural descriptions. “Architectural descriptions are applicable to a variety of uses, by a variety of stakeholders, throughout the life cycle” ([IEE 00], p.8). Here are a few examples of architectural descriptions use:

- “analysis of alternative architectures;
- business planning for transition from a legacy architecture to a new architecture;
- communications among organizations involved in the development, production, fielding, operation, and maintenance of a system;

- communications between acquirers and developers as a part of contract negotiations;
- “criteria for certifying conformance of implementations to the architecture;
- development and maintenance documentation, including material for reuse repositories and training materials;
- input to subsequent system design and development activities;
- operational and infrastructure support; configuration management and repair; redesign and maintenance of systems, subsystems, and components;
- planning and budget support;
- preparation of acquisition documents;
- review, analysis, and evaluation of the system across the life cycle;
- specification for a group of systems sharing a common set of features” ([IEE 00], p. 8).

Architectural description practices describe the following:

- “An architectural documentation shall contain the following information [...]: date of issue and status, issuing organization, change history, summary, scope, context, glossary, references” ([IEE 00], p. 8-9).
- Identification of stakeholders and concerns. “An AD shall identify the stakeholders considered by the architect in formulating the architectural concept for the system. At a minimum, the stakeholders identified shall include the following:
 - users of the system,
 - acquirers of the system,
 - developers of the system,
 - maintainers of the system,
 - the purpose of missions of the system,
 - the appropriateness of the system for use in fulfilling its missions,
 - the feasibility of constructing the system,
 - the risks of system development and operation to users, acquirers, and developers of the system,
 - maintainability, deployability, and evolvability of the system” ([IEE 00], p. 9).

- Selection of architectural viewpoints. “An AD shall identify the viewpoints selected for use therein. Each viewpoint shall be specified by:
 - a viewpoint name,
 - the stakeholders to be addressed by the viewpoint,
 - the concerns to be addressed by the viewpoint,
 - the language, modeling techniques, or analytical methods to be used in constructing a view based upon the viewpoint,
 - the source” ([IEE 00], p. 9 and 10).
- Architectural views. “An AD shall contain one or more architectural views” ([IEE 00], p. 10);
- Consistency among architectural views. “An AD shall record any known inconsistencies among its architectural views. It should contain an analysis of consistency across all of its architectural views” ([IEE 00], p. 11);
- Architectural rationale. “An AD shall include the rationale for the architectural concepts selected” ([IEE 00], p. 11).

The standard ends with a bibliography and appendixes featuring a terminology, examples of point of views, and relations with other standards.

12.13. Appendix F. Unified modeling language

UML features a certain number of diagrams, which rely on key concepts, to represent the following software constructs:

- the structural and static constructs;
- the behavioral and dynamic constructs.

The key-concepts of the structural and static constructs are:

- “the classes package contains sub packages that deal with the basic modeling concepts of UML, and in particular classes and their relationships” ([OMG 07b], p. 23). This class package is the keystone of data modeling in the context of object-oriented analysis. Among other things, the following elements are featured:
 - the links between classes (association, aggregation, abstraction, etc.),
 - characteristics,
 - behaviors,
 - collaborations,

- constraints,
- logical expression,
- types of data,
- dependencies.

– “The components package specifies a set of constructs that can be used to define software systems of arbitrary size and complexity. In particular, the package specifies a component as a modular unit with well-defined interfaces that is replaceable within its environment” ([OMG 07b], p. 143).

– The composite structure package “refers to a composition of interconnected elements, representing run-time instances collaborating over communications links to achieve some common objectives” ([OMG 07b], p. 161). The following structures are defined:

- internal structures,
- ports,
- collaborations,
- structured classes,
- actions.

– “The deployments package specifies a set of constructs that can be used to define the execution architecture of systems that represent the assignment of software artifacts to nodes” ([OMG 07b], p. 195).

The diagrams which implement these key-concepts for the structural and static constructs are:

- for the concept of class:
 - class diagram,
 - object diagram,
 - package diagram;
- for the concept of component: component diagram;
- for the concept of composite structures: composite structure diagram;
- for the concept of deployment: deployment diagram.

The key-concepts in the behavioral and dynamic constructs are as follows:

- “An “action is the fundamental unit of behavior specification. An action takes a set of inputs and converts them into a set of outputs, though either or both sets may be empty. [...] Some of the actions modify the state of the system in which the action executes. [...] Actions are contained in behaviors, which provide their context. Behaviors provide constraints among actions to determine when they execute and what inputs they have. [...] Basic actions include those that perform operation calls, signal sends, and direct behavior invocations” ([OMG 07b], p. 219).

- “Activity modeling emphasizes the sequence and conditions for coordinating lower-level behaviors. These are commonly called control flow and object flow models. The actions coordinated by activity models can be initiated because other actions finish executing, because objects and data become available, or because events occur external to the flow. Each action in an activity may execute zero, one, or more times for each activity execution” ([OMG 07b], p. 295).

- “The Common Behaviors packages specify the core concepts required for dynamic elements and provides the infrastructure to support more detailed definitions of behavior” ([OMG 07b], p. 419).

- “Interactions are used during the more detailed design phase where the precise inter-process communication must be set up according to formal protocols. When testing is performed, the traces of the system can be described as interactions and compared with those of the earlier phases” ([OMG 07b], p. 455).

- “The State Machine package defines a set of concepts that can be used for modeling discrete behavior through finite state-transition systems” ([OMG 07b], p. 519).

- “Use cases are a means for specifying required usages of a system. Typically, they are used to capture the requirements of a system, that is, what a system is supposed to do. The key concepts associated with use cases are *actors*, *use cases*, and the *subject*. The subject is the system under consideration to which the use cases apply. The users and any other systems that may interact with the subject are represented as actors. Actors always model entities that are outside the system. The required behavior of the subject is specified by one or more use cases, which are defined according to the needs of actors” ([OMG 07b], p. 581).

The diagrams which implement these key-concepts for the behavioral and dynamic constructs are:

- for the use cases: use cases diagram;
- for activity modeling: activity diagram;

- for interactions:
 - interaction diagram,
 - communication diagram,
 - interaction overview diagram,
 - sequence diagram,
 - timing diagram;
- for the StateMachine package: statemachine diagram.

12.14. Appendix G. Systems modeling language

SysML builds on the two constructs of UML, to which are added Crosscutting Constructs.

The two constructs are adapted to the context of systems in the following manner:

- The structural and static construct evolves to integrate structure diagrams, including the block definition diagram, the internal block diagram, the parametric diagram. The structuring frameworks are defined in the sections dedicated to model elements, blocks, ports and flows, constraint blocks. “The Model Elements [rely on] the kernel package from UML 2 and includes some extensions to provide some foundation capabilities for model management. The Blocks chapter reuses and extends structured classes from UML 2 composite structures to provide the fundamental capability for describing system decomposition and interconnection, and to define different types of system properties including value properties with optional units of measure. The Ports and Flows chapter provides the semantics for defining how blocks and parts interact through ports and how items flow across connectors. The Constraint Blocks chapter defines how blocks are extended to be used on parametric diagrams. Parametric diagrams model a network of constraints on system properties to support engineering analysis, such as performance, reliability, and mass properties analysis” ([OMG 06c], p. 19).
- The behavioral and dynamic constructs are only superficially modified, with add-ons and evolutions affecting their concepts.
- The crosscutting constructs apply to both structure and behavior. These cross-cutting constructs concern notions of allocation, requirements and profiles and models libraries. The notion of allocation defines “a basic allocation relationship that can be used to allocate a set of model elements to another, such as allocating behavior to structure or allocating logical to physical components.” The notion of requirements “specifies constructs for system requirements and their relationships.”

Finally, the notion of profiles and models libraries defines “the approach to further customize and extend SysML for specific applications” ([OMG 06c], p. 121).

The key-concepts of the structural and static constructs are as follows:

- The model element package includes packages, models, dependencies, constraints, and comments. It is used “to organize the model by partitioning model elements into packageable elements and establishing dependencies between the packages and/or model elements within the package. [...] Packages can also be shown on other diagrams such as the block definition diagram, requirement diagram, and behavior diagrams” ([OMG 06c], p. 21).
- “Blocks are modular units of system description. Each block defines a collection of features to describe a system or other element of interest. These may include both structural and behavioral features, such as properties and operations, to represent the state of the system and behavior that the system may exhibit. Blocks provide a general-purpose capability to model systems as trees of modular components [...] These include modeling either the logical or physical decomposition of a system. Parts in these systems may interact by many different means, such as discrete state transitions, flows of inputs and outputs. The Block definition diagram in SysML defines features of blocks and relationships between blocks such as associations, generalizations, and dependencies. It captures the definition of blocks in terms of properties and operations, and relationships such as a system hierarchy or a system classification tree. [...] A block can include properties to specify its values, parts, and references to other blocks. Ports are a special class of property used to specify allowable types of interactions between blocks. Constraint Properties are a special class of property used to constrain other properties of blocks” ([OMG 06c], p. 31).
- The concept of ports and flows: “This chapter specifies flow ports that enable flow of items between blocks and parts, as well as standard ports that enable invocation of services on blocks and parts. A port is an interaction point between a block or part and its environment that is connected with other ports via connectors. The main motivation for specifying such ports on system elements is to allow the design of modular reusable blocks, with clearly defined interfaces” ([OMG 06c], p. 61).
- “Constraint blocks provide a mechanism for integrating engineering analysis such as performance and reliability models with other SysML models. Constraint blocks can be used to specify a network of constraints that represent mathematical expressions [...] which constrain the physical properties of a system. Such constraints can also be used to identify critical performance parameters and their relationships to other parameters, which can be tracked throughout the system life cycle. [...] Time can be modeled as a property that other properties may be dependent on” ([OMG 06c], p. 75).

The diagrams which implement these key-concepts for the structural and static constructs are:

- for activity modeling: activity diagram;
- for the model element package: package diagram;
- for blocks:
 - block definition diagram,
 - internal block diagram;
- for ports and flows:
 - block definition diagram,
 - internal block diagram;
- for constraint blocks: parametric diagram.

The key-concepts in the behavioral and dynamic constructs are:

- actions are not used, and may be replaced by activity modeling;
- “in UML 2.1 activities, control can only enable actions to start. SysML extends control to support disabling of actions that are already executing.” This is performed by a control operator. Moreover, the activities evolve with the acknowledgement of continuous systems, in order to define continuous (water, energy) and non-continuous (material) flows. Finally, other evolutions concern the optional character and the probabilities, for example in the context of decision-making;
- the common behaviors package is not taken over by SysML;
- the concept of interactions is not drastically modified, although many diagrams are not reused; only the sequence diagram is featured ([OMG 06c], p. 171):
 - the sequence diagram, the only one to transfer from UML to SysML;
 - the interaction diagram, not featured in SysML;
 - the communication diagram, not featured in SysML;
 - the interaction overview diagram, not featured in SysML;
 - the timing diagram, not featured in SysML;
- the state machine package featured in UML does not evolve much. “The UML concept of protocol state machines is excluded from SysML to reduce the complexity of the language” ([OMG 06c], p. 115);
- the use cases are reused as-is.

The diagrams which implement these key-concepts for the behavioral and dynamic constructs are:

- for the state machine package: state machine diagram;
- for interactions: sequence diagram;
- for use cases: use cases diagram.

The key-concepts of the crosscutting constructs are as follows:

– “Allocation is the term used by systems engineers to denote the organized cross-association (mapping) of elements within the various structures or hierarchies of a user model. The concept of “allocation” requires flexibility suitable for abstract system specification, rather than a particular constrained method of system or software design. [...] Allocations can be used early in the design as a precursor to more detailed rigorous specifications and implementations. The allocation relationship can provide an effective means for navigating the model by establishing cross relationships, and ensuring the various parts of the model are properly integrated” ([OMG 06c], p. 129).

– “A requirement specifies a capability or condition that must (or should) be satisfied. A requirement may specify a function that a system must perform or a performance condition a system must achieve. SysML provides modeling constructs to represent text-based requirements and relate them to other modeling elements. The requirements diagram described in this chapter can depict the requirements in graphical, tabular, or tree structure format. A standard requirement includes properties to specify its unique identifier and text requirement. Additional properties such as verification status, can be specified by the user. Several requirements relationships are specified [including relationships] for defining a requirements hierarchy, deriving requirements, satisfying requirements, verifying requirements, and refining requirements” ([OMG 06c], p. 141).

– “The profiles package contains mechanisms that allow metaclasses from existing metamodels to be extended to adapt them for different purposes. This includes the ability to tailor the UML metamodel for different domains. The profiles mechanism is consistent with the OMG meta object facility (MOF)” ([OMG 06c], p. 157).

The diagrams which implement these crosscutting constructs are:

- allocation: numerous diagrams;
- requirements: requirement diagram.

12.15. Appendix H. Good practices of IT service management, ITIL

ITIL (Information Technology Infrastructure Library) describes the best practices of IT service management within two main groups:

- service supply, including [OGC 06]:

- the process of Service Level Management (SLM) validates Service Level Agreements (SLA), Operational Level Agreements (OLA) and guarantees minimum impact over the Quality of Service (QoS). This process implies the evaluation of the changes' impact on the quality of service and the SLA, whether these changes are proposed or already implemented. Some objectives defined in the SLA concern service availability, and therefore require incident management to be performed within a delimited timeframe ([OGC 06], p. 11). The goals of SLM are to maintain and improve IT service quality, via an unbroken cycle of agreements, monitoring and reporting on the respect of IT service levels, and the triggering of actions to eradicate all mediocrity of service, in agreement with business or cost justification ([OGC 06], p. 29). This is achieved through the process's planning, but also, among other things, through the management of user expectations, the creation of a service catalogue, the development of SLA and the related requirements, and the search for an agreement;

- financial management is in charge of calculating the cost of providing IT services, and reporting on the aspects relative to these costs' recovery (charging and billing). This requires tight links with capacity management, configuration management (information on goods) and service level management, so as to identify the services' real cost. Financial management might therefore work in close relationship with Customer Relationship Management (CRM) and IT management during negotiations to set the budgets and the customers' individual expenses ([OGC 06], p. 11);

- capacity management, which ensures that the services' capacity meets the business requirements at all times. It is in direct relationship with business requirements and is not only concerned with the performance of the system's components, whether taken individually or as a whole. Capacity management plays a part in incident and problem management, when those are related to the capacity objectives ([OGC 06], p. 11);

- IT service continuity management focuses on an organization's ability to always provide a predefined level of service so as to generate as few business demands as possible in the event of a serious incident. To achieve efficient continuity of IT services, one needs to balance risk reduction measures and options for recovery, such as back up tools. Configuration management data is needed to facilitate its prevention and planning. The potential impacts which changes of

infrastructure and business might have on continuity plans must be evaluated ([OGC 06], p. 12);

- availability management concerns the design, implementation, measure and management of IT services, to ensure that business requirements are always met. Availability management requires knowledge of what causes the service's disruption, and of the time needed for said service to be restored ([OGC 06], p. 11);

- planning to implement service management. The implementation of service management must be organized, planned, following an approach of project management, a feasibility study, an evaluation of the current situation, etc. ([OGC 06], p. 325-336).

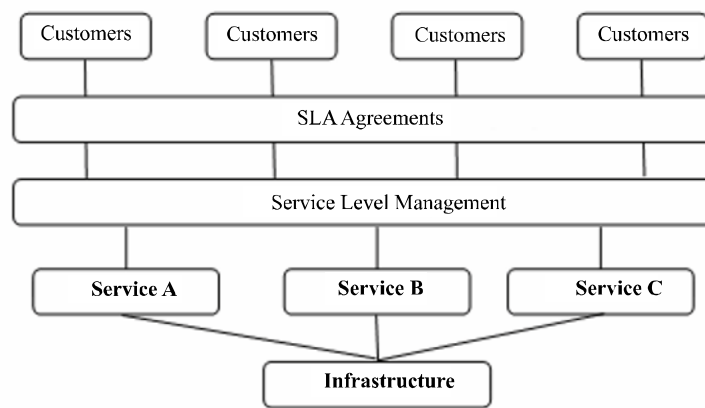


Figure 12.8. *Structure of the links between IT services suppliers and customers*

The above schema displays the links between the services' supplier and its customers, which are based on Service Level Agreements (SLA). Service level management ensures that those agreements are implemented, calling on the various services as needed. These services are provided by the infrastructure of the service supplier. This schema is featured in the presentation of service delivery ([OGC 06], p. 30);

- service support (management of dysfunctions and implementation of technical changes) [OGC 05]:

- the service desk is the only daily meeting place between service suppliers and customers. It is also a central point for the reporting of incidents and service requests. As such, the service desk must keep the users informed of events which concern the services, and the actions which might compromise their daily activities

([OGC 05], p. 13). The incident management process, problem management process, and change management process must be tightly linked, on top of the service provided by the service desk. If they are not properly managed, changes may lead to new incidents. A back-out method must be designed ([OGC 05], p. 13);

- problem management requires precise and comprehensible records of incidents, to clearly and efficiently identify their causes. Problem management also requires close communication with availability management to identify trends and encourage corrective actions ([OGC 05], p. 13);

- configuration management is an essential part for all other service management processes. With up-to-date information, extended and precise about all the components of the infrastructure, the management of changes, in particular, becomes more efficient. Change management can be integrated to configuration management. At the very least, the recording and implementation of changes must be managed by an extended configuration management system, and the evaluation of these changes' impact must be processed by such a system ([OGC 05], p. 11);

- the change management process depends on the accuracy of configuration data, to ensure that the consequences of changes are well-known. Configuration management, release management and change management are tightly linked. The details of the change process are documented within the SLAs, to ensure that users know the request procedures, the time necessary to schedule the task, and its impact. The changes' details must be known by the service desk ([OGC 05], p. 12);

- changes often occur when new hardware, new versions of software and/or new documentation, are needed, created within the enterprise or bought, to be managed or distributed as part of a new release bundle. The procedures which ensure safe and proper deployment should generally be integrated to change management procedures. The release procedures should also be an integral part of incident and problem management, on top of their close relationship with the configuration management database, in order to keep the records up-to-date ([OGC 05], p. 12-13).

On top of these two groups, ITIL defines a collection of good practices relative to:

- the business perspective, including ([OGC 05], p. 5):
 - business continuity management;
 - partnerships and outsourcing;
 - surviving change;
 - transformation of business practice through radical change;

- application management, focusing on the life-cycle of software development projects, and testing IT services ([OGC 05], p. 6);
- ICT (Information Communication Technology) infrastructure management, including ([OGC 05], p. 6):
 - network services management,
 - operation management,
 - local process management,
 - technical support,
 - IT system management.

Finally, ITIL takes other processes into account, such as:

- customer relationship management;
- safety management;
- project management;
- environment infrastructure processes.

12.16. Appendix I. Standard relative to IT services management

The standard relative to IT services management features two documents:

- the specifications ([ISO 05e], p. 5), including:
 - “requirements for a management system” aim at providing a policy and framework for the management and implementation of all efficient IT services ([ISO 05e], p. 5);
 - “planning and implementing service management” builds on the Deming wheel, with the PDCA approach (Plan, Do, Check, Act), adapted into service management planning (Plan), management and service supply implementation (Do), control, measurement and reviews (Check), and continuous improvement (Act) ([ISO 05e], p. 4-7);
 - “planning and implementing new or changed services” aims at ensuring that new and changed services will be provided and managed following the cost and quality defined in the agreement ([ISO 05e], p. 7);
 - “service delivery process” aims at defining, agreeing on, recording and managing service levels ([ISO 05e], p. 8);

- “relationship processes” describe the two correlated aspects of supplier management and customer relationships management ([ISO 05e], p. 11);
- “resolution processes” concern incidents and problems management ([ISO 05e], p. 13);
- “control processes” aim to define and control the components of both services and infrastructure, and maintain a specific configuration ([ISO 05e], p. 14);
- “release processes” aim to deliver, distribute and track one or more changes of release within the operating environment ([ISO 05e], p. 15);
- the code of good practice ([ISO 05f], p. 6) includes:
 - the “management system” describes, among other things, the management of responsibilities, and the documentation of requirements ([ISO 05f], p. 2-4);
 - “planning and implementing service management” describes good practices in terms of service management scope, planning rationales, service implementation, etc. ([ISO 05f], p. 4-7);
 - “planning and implementing new or changed services” describes good practices relative to the subjects which must be taken into account (costs, human resources, existing service levels, commitments in terms of service level agreements, etc.), and the recording of changes ([ISO 05f], p. 7-8);
 - “service delivery process” describes good practices to design a service catalogue, define service level agreements, service level management processes, service reports, etc. ([ISO 05f], p. 8-17);
 - “relationship processes” describes good practices in terms of service review, service complaints, client satisfaction measurement, service definition, etc. ([ISO 05f], p. 17-20);
 - “resolution processes” describes good practices to define priorities, manage incidents and problems, etc. ([ISO 05f], p. 20-24);
 - the “control processes” describe good practices to plan and manage configuration management, identification and control, their status and reports, audits and verifications, the management of changes, their planning and implementation, the feedback on and closing of change requests, urgent changes, the analysis and reports on the management of changes ([ISO 05e], p. 25-29);
 - the “release processes” describe good practices such as the delivery policy, delivery scheduling, software development and acquisition, the design and development of the delivered service, the verification and acceptance of the delivery, documentation, etc. ([ISO 05f], p. 29-33).

12.17. Appendix J. Software engineering – Product quality

The standard ISO 9126 identifies the characteristics and sub-characteristics of software quality. This standard is divided into four parts:

- part 1, quality model;
- part 2, external metrics;
- part 3, internal metrics;
- part 4, quality in use metrics.

12.18. Appendix J.1. Standard ISO 9126, part 1, quality model

Part 1 describes the quality model. This model includes characteristics and sub-characteristics.

Each sub-characteristic presents the definition issued from the standard ([ISO 01], p. 8–11):

- functionality:
 - suitability: “the capability of the software product to provide an appropriate set of functions for specified tasks and user objectives”;
 - accuracy: “the capability of the software product to provide the right or agreed results or effects with the needed degree of precision”;
 - interoperability: “the capability of the software product to interact with one or more specified systems”;
 - compliance: “the capability of the software product to adhere to standards, conventions or regulations in laws and similar prescriptions relating to functionality”;
 - security: “the capability of the software product to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them”;
- reliability:
 - maturity: “the capability of the software product to avoid failure as a result of faults in the software”;
 - fault tolerance: “the capability of the software product to maintain a specified level of performance in cases of software faults or of infringement of its specified interface”;

- recoverability: “the capability of the software product to re-establish a specified level of performance and recover the data directly affected in the case of a failure. [...] Following a failure, a software product will sometimes be down for a certain period of time, the length of which is assessed by its recoverability”;

- usability:

- understandability: “the capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use”;

- learnability: “the capability of the software product to enable the user to learn its application” (for example control of inputs and outputs exploitation);

- operability: “the capability of the software product to enable the user to operate and control it”;

- efficiency;

- time behavior: “the capability of the software product to provide appropriate response and processing times and throughput rates when performing its function, under stated conditions”;

- resource utilization: “the capability of the software product to use appropriate amounts and types of resources when the software performs its function under stated conditions”;

- maintainability:

- analyzability: “the capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified to be identified”;

- stability: “the capability of the software product to avoid unexpected effects from modifications of the software”;

- changeability: “the capability of the software product to enable a specified modification to be implemented”;

- testability: “the capability of the software product to enable modified software to be validated”;

- portability:

- adaptability: “the capability of the software product to be adapted for different specified environments without applying actions or means other than those provided for this purpose for the software considered”;

- installability: “the capability of the software product to be installed in a specified environment”;

- portability compliance: “the capability of the software product to adhere to standards or conventions relating to portability”;

- replaceability: “the capability of the software product to be used in place of another specified software product for the same purpose in the same environment” ([ISO 01], p. 8–11).

12.19. Appendix J.2. Standard ISO 9126, part 3, internal metrics

For each of the aforementioned characteristics and sub-characteristics, the document defines internal metrics and offers a table format to structure the metrics.

The table follows this structure:

Name	Purpose	Method of application	Measurement, formula and data element computations	Interpretation of measured value	Metric scale type	Measure type
------	---------	-----------------------	--	----------------------------------	-------------------	--------------

12.20. Appendix K. Standard on software product quality requirements and evaluation

The standard on software product quality requirements and evaluation (ISO 25,000) presents the following structure:

- quality management: management and scheduling of software product evaluation;
- quality model and guide: description of the external quality measure, internal quality measure and quality in use measure models;
- quality measurement: description of quality measures in terms of:
 - measurement reference model and guide;
 - measurement primitives;
 - metrics for external quality;
 - metrics for internal quality;
 - metrics for quality in use;
- quality requirements; guide for the redaction of quality requirements, to specify software product quality in terms of quality requirements, through the software’s entire life cycle, including acquisition, maintenance and operation;

– evaluation process overview and guide; describes evaluation processes in terms of:

- quality requirements and guide;
- process for developers;
- process for acquirers;
- process for evaluators;
- evaluation documentation model.

The structure of the future standard, software and system engineering: software product quality requirements and evaluation (SQuaRE) is as follows:

- ISO/IEC 25000 - *Guide to SquaRE*;
- ISO/IEC 25001 - Planning and management;
- ISO/IEC 25010 - Quality model and guide;
- ISO/IEC 25020 - Measurement reference model and guide;
- ISO/IEC 25021 - Measurement primitives;
- ISO/IEC 25022 - Metrics for internal quality;
- ISO/IEC 25023 - Metrics for external quality;
- ISO/IEC 25024 - Metrics for quality in use;
- ISO/IEC 25025 - Documentation of evaluation modules;
- ISO/IEC 25030 - Quality requirements and guide;
- ISO/IEC 25040 - Evaluation process overview and guide;
- ISO/IEC 25041 - Process for developers;
- ISO/IEC 25042 - Process for acquirers;
- ISO/IEC 25043 - Process for evaluators.

12.21. Appendix L. Standard on the common criteria for IT security evaluation

The standard on the common criteria for IT security evaluation (ISO 15408) is broken down into three parts:

- Part 1, “Introduction and general model”, defines “general concepts and principles of IT security evaluation and presents a general model of evaluation. Part 1 also presents constructs for expressing IT security objectives, for selecting and

defining IT security requirements, and for writing high-level specifications for products and systems.

- Part 2, “Security functional requirements”, establishes a set of functional components as a standard way of expressing the functional requirements for TOEs (targets of evaluation).

- Part 3, “Security assurance requirements”, establishes a set of assurance components as a standard way of expressing the assurance requirements for TOEs. [...] Part 3 also defines evaluation criteria for Protection Profiles and Security Targets and presents evaluation assurance levels that define the predefined ISO/IEC 15408 scale for rating assurance for TOES” ([ISO 05b], p. 10).

The general model of Part 1 features:

- the security context: “Security is concerned with the protection of assets from threats, where threats are categorized as the potential for abuse of protected assets. All categories of threats should be considered; but in the domain of security greater attention is given to those threats that are related to malicious or other human activities” ([ISO 05b], p. 11). The damages may translate into loss of confidentiality (non-authorized agents access the data), loss of integrity (non-authorized agents modify the data) or loss of availability (the data are no longer available). Risk analysis leads to the elaboration of counter-measures which aim at reducing vulnerability. These counter-measures must be evaluated and create a feeling of trust. Finally, evaluations must lead to objective, repeatable results;

- “ISO/IEC 15408 approach. confidence in IT security can be gained through actions that may be taken during the processes of development, evaluation, and operation” ([ISO 05b], p. 13);

- “security concepts: evaluation criteria are most useful in the context of the engineering processes and regulatory frameworks that are supportive of secure TOE development and evaluation” ([ISO 05b], p. 16). These concepts are:

- “the security environment includes all the laws, organizational security policies, customs, expertise and knowledge that are determined to be relevant. It thus defines the context in which the TOE is intended to be used. The security environment also includes the threats to security that are, or are held to be, present in the environment” ([ISO 05b], p. 17);

- the security objectives; “the results of the analysis of the security environment could then be used to state the security objectives that counter the identified threats and address identified organizational security policies and assumptions” ([ISO 05b], p. 18);

- “the IT security requirements are the refinement of the security objectives into a set of security requirements for the TOE and security requirements for the

environment which, if met, will ensure that the TOE can meet its security objectives” ([ISO 05b] p. 18);

- “the TOE summary specification [provides] a high-level definition of the security functions claimed to meet the functional requirements, and assurance measures taken to meet the assurance requirements” ([ISO 05b], p. 19);

- “the TOE implementation is the realization of the TOE based on its security functional requirements” ([ISO 05b], p. 19);

- the standard offers a framework for the evaluation. Improved objectivity and usability of evaluation results may be obtained through proof and analysis requirements.

Part 2 is dedicated to security functional requirements. “These requirements describe the desired security behavior expected of a Target of Evaluation (TOE) or the IT environment of the TOE and are intended to meet the security objectives as stated in a PP or an ST” ([ISO 05c], p.xx). These functional requirements are grouped within the following themes:

- “Security auditing involves recognizing, recording, storing, and analyzing information related to security relevant activities” ([ISO 05c], p. 11).

- Communication is concerned with “assuring the identity of a party participating in a data exchange [...] the identity of the originator of transmitted information (proof of origin) [...] the identity of the recipient of transmitted information (proof of receipt). These families ensure that an originator cannot deny having sent the message, nor can the recipient deny having received it” ([ISO 05c], p. 22).

- Cryptographic support: “the TOE Security Functions (TSF) may employ cryptographic functionality to help satisfy several high-level security objectives. These include (but are not limited to): identification and authentication, non-repudiation, trusted path, trusted channel and data separation” ([ISO 05c], p. 25). Cryptographic support addresses the management and operational use of those cryptographic keys.

- User data protection: “User data protection is split into four groups of families that address user data within a TOE, during import, export, and storage” ([ISO 05c] p. 29). These groups are:

- user data protection security function policies,
- forms of user data protection,
- offline storage, import and export,
- inter-TSF communication.

- Identification and authentication: these functions “establish and verify a claimed user identity. Identification and Authentication is required to ensure that users are associated with the proper security attributes (e.g. identity, groups, roles, security or integrity levels)” ([ISO 05c], p. 54).
- Security management has several objectives, including management of data, management of security attributes, the definition of roles.
- Privacy requirements: “these requirements provide a user protection against discovery and misuse of identity by other users” ([ISO 05c], p. 74).
- Protection of the TSF: this relates “to the integrity and management of the mechanisms that provide the TSF [...] and to the integrity of TSF data” ([ISO 05c], p. 80).
- Resource utilization supports “the availability of required resources such as processing capability and/or storage capacity. The family Fault Tolerance provides protection against unavailability of capabilities [...]. The family Priority of Service ensures that the resources will be allocated to the more important or time-critical tasks and cannot be monopolized by lower priority tasks. The family Resource Allocation provides limits on the use of available resources, therefore preventing users from monopolizing the resources” ([ISO 05c], p. 102).
- TOE access “specifies functional requirements for controlling the establishment of a user’s session” ([ISO 05c], p. 106);
- Trusted path/channels provide “requirements for a trusted communication path between users and the TSF, and for a trusted communication channel between the TSF and other trusted IT products. Trusted paths and channels have the following general characteristics:
 - the communications path is constructed using internal and external communications channels (as appropriate for the component) that isolate an identified subset of TSF data and commands from the remainder of the TSF and user data;
 - use of the communications path may be initiated by the user and/or the TSF (as appropriate for the component);
 - the communications path is capable of providing assurance that the user is communicating with the correct TSF, and that the TSF is communicating with the correct user (as appropriate for the component)” ([ISO 05c] p. 113).

The third part is dedicated to security assurance requirements. It includes an evaluation assurance scale, which measures the security assurance, the components from which the levels are elaborated, and the evaluation criteria. These requirements are grouped into the following themes:

- “security assurance requirements” ([ISO 05d], p. 4);
- “the goal of a protection profile evaluation is to demonstrate that the PP is complete, consistent and technically sound. An evaluated PP is suitable for use as the basis for the development of STs” ([ISO 05d], p. 20);
- “the goal of an ST evaluation is to demonstrate that the ST is complete, consistent, technically sound, and hence suitable for use as the basis for the corresponding TOE evaluation” ([ISO 05d], p. 28);
- “the evaluation assurance levels (EALs) provide an increasing scale that balances the level of assurance obtained with the cost and feasibility of acquiring that degree of assurance” ([ISO 05d] p. 39);
- “configuration management” ([ISO 05d], p. 47);
- delivery and operation: “the requirements for delivery call for system control and distribution facilities and procedures that detail the measures necessary to provide assurance that the security of the TOE is maintained during distribution of the TOE”, while addressing threats” ([ISO 05d] p.61). The operation requirements concern the installation and start-up of the system;
- “development” ([ISO 05d], p. 66);
- “guidance documents” ([ISO 05d], p. 99);
- life-cycle support: it translates into “security analysis and the production of the evidence [...] on a regular basis as an integral part of the development and maintenance activities” ([ISO 05d], p. 102);
- “tests encompass four families: coverage, depth, independent testing, and functional tests. Testing helps to establish that the TOE security functional requirements are met” ([ISO 05d], p. 114);
- vulnerability assessment “addresses the existence of exploitable covert channels, the possibility of misuse or incorrect configuration of the TOE, the possibility to defeat probabilistic or permutation mechanisms, and the possibility of exploitable vulnerabilities introduced in the development or the operation of the TOE” ([ISO 05d] p. 127).

12.22. Appendix M. Standard relative to a system’s life cycle process

For each life-cycle process of a system, the standard describes the object, the activities and their results.

These processes are of various types:

- agreement processes;

- enterprise processes;
- project processes;
- technical processes.

Agreement processes “define the activities necessary to establish an agreement between two organizations” ([ISO 02a], p. 4). They include the following:

- “Acquisition process. The purpose of the acquisition process is to obtain a product or service in accordance with the acquirer’s requirements” ([ISO 02a], p. 4).
- “Supply process. The purpose of the supply process is to provide an acquirer with a product or service that meets agreed requirements” ([ISO 02a], p. 7).

“The enterprise processes manage the organization’s capability to acquire and supply products or services through the initiation, support and control of projects. They provide resources and infrastructure necessary to support projects and ensure the satisfaction of organizational objectives and established agreements” ([ISO 02a], p. 8).

They include the following:

- “Enterprise environment management process. The purpose of the enterprise environment management process is to define and maintain the policies and procedures needed for the organization’s business with respect to the scope of this international standard” ([ISO 02a], p. 9).
- “Investment management process. The purpose of the investment management process is to initiate and sustain sufficient and suitable projects in order to meet the objectives of the organization. This process commits the investment of adequate organization funding and resources, and sanctions the authorities needed to establish selected projects. It performs continued qualification of projects to confirm they justify, or can be redirected to justify, continued investment” ([ISO 02a], p. 10).
- “System life cycle processes management process. The purpose of the system life cycle processes management process is to assure that effective system life cycle processes are available for use by the organization. This process provides system life cycle processes that are consistent with the organization’s goals and policies, that are defined, adapted and maintained in a consistent way in order to meet the nature of individual projects, and that are capable of being applied using effective, proven methods and tools” ([ISO 02a], p. 11).
- “Resource management process. The purpose of the resource management process is to provide resources to projects. This process provides resources, materials and services to projects to support organization and project objectives throughout the life cycle. This includes a supply of educated, skilled and

experienced personnel qualified to perform life cycle processes. This process assures that there is effective co-ordination and sharing of resources, information and technologies” ([ISO 02a], p. 12).

– “Quality management process. the purpose of the quality management process is to assure that products, services and implementations of life cycle processes meet enterprise quality goals and achieve customer satisfaction” ([ISO 02a], p. 13).

“The project processes are used to establish and evolve project plans, to assess actual achievement and progress against the plans and to control execution of the project through to fulfillment. Individual project processes may be invoked at any time in the life cycle and at any level in a hierarchy of projects, as required by project plans or unforeseen events. The Project Processes are applied with a level of rigor and formality that depends on the risk and complexity of the project” ([ISO 02a], p. 13).

They include:

– “the purpose of the project planning process is to produce and communicate effective and workable project plans. This process determines the scope of the project management and technical activities, identifies process outputs, project tasks and deliverables, establishes schedules for project task conduct, including achievement criteria, and required resources to accomplish project tasks” ([ISO 02a], p. 14);

– “the purpose of the project assessment process is to determine the status of the project. This process evaluates, periodically and at major events, the progress and achievements against requirements, plans and overall business objectives. Information is communicated for management action when significant variances are detected” ([ISO 02a], p. 15);

– “the purpose of the project control process is to direct project plan execution and ensure that the project performs according to plans and schedules, within projected budgets and it satisfies technical objectives. This process includes redirecting the project activities, as appropriate, to correct identified deviations and variations from other project management or technical processes. Redirection may include replanning as appropriate” ([ISO 02a], p. 17);

– “the purpose of the decision-making process is to select the most beneficial course of project action where alternatives exist. This process responds to a request for a decision encountered during the system life cycle, whatever its nature or source, in order to reach specified, desirable or optimized outcomes. Alternative actions are analyzed and a course of action selected and directed. Decisions and their rationale are recorded to support future decision-making” ([ISO 02a], p. 18);

– “the purpose of the risk management process is to reduce the effects of uncertain events that may result in changes to quality, cost, schedule or technical characteristics. This process identifies, assesses, treats and monitors risks during the entire life cycle, responding to each risk in terms of appropriate treatment or acceptance” ([ISO 02a], p. 19);

– “the purpose of the configuration management process is to establish and maintain the integrity of all identified outputs of a project or process and make them available to concerned parties” ([ISO 02a], p. 20);

– “the purpose of the information management process is to provide relevant, timely, complete, valid and, if required, confidential information to designated parties during and, as appropriate, after the system life cycle. This process generates, collects, transforms, retains, retrieves, disseminates and disposes of information. It manages designated information, including technical, project, enterprise, agreement and user information” ([ISO 02a], p. 21).

“The technical processes are used to define the requirements for a system, to transform the requirements into an effective product, to permit consistent reproduction of the product where necessary, to use the product to provide the required services, to sustain the provision of those services and to dispose of the product when it is retired from service. The Technical Processes define the activities that enable enterprise and project functions to optimize the benefits and reduce the risks that arise from technical decisions and actions. These activities enable products and services to possess the timeliness and availability, the cost effectiveness, and the functionality, reliability, maintainability, producibility, usability and other qualities required by acquiring and supplying organizations. They also enable products and services to conform to the expectations or legislated requirements of society, including health, safety, security and environmental factors” ([ISO 02a], p. 22).

They include:

– “the purpose of the stakeholder requirements definition process is to define the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment. It identifies stakeholders, or stakeholder classes, involved with the system throughout its life cycle, and their needs and desires. It analyzes and transforms these into a common set of stakeholder requirements that express the intended interaction the system will have with its operational environment and that are the reference against which each resulting operational service is validated in order to confirm that the system fulfils needs” ([ISO 02a], p. 23);

– “the purpose of the requirements analysis process is to transform the stakeholder, requirement-driven view of desired services into a technical view of a required product that could deliver those services. This process builds a

representation of a future system that will meet stakeholder requirements and that, as far as constraints permit, does not imply any specific implementation. It results in measurable system requirements that specify, from the developer's perspective, what characteristics it is to possess and with what magnitude in order to satisfy stakeholder requirements" ([ISO 02a], p. 25);

– "the purpose of the architectural design process is to synthesize a solution that satisfies system requirements. This process encapsulates and defines areas of solution expressed as a set of separate problems of manageable, conceptual and, ultimately, realizable proportions. It identifies and explores one or more implementation strategies at a level of detail consistent with the system's technical and commercial requirements and risks. From this, an architectural design solution is defined in terms of the requirements for the set of system elements from which the system is configured. The specified requirements resulting from this process are the basis for verifying the realized system and for devising an assembly and verification strategy" ([ISO 02a], p. 27);

– "the purpose of the implementation process is to produce a specified system element. This process transforms specified behavior, interfaces and implementation constraints into fabrication actions that create a system element according to the practices of the selected implementation technology. The system element is constructed or adapted by processing the materials and/or information appropriate to the selected implementation technology and by employing appropriate technical specialisms or disciplines. This process results in a system element that satisfies architectural design requirements through verification and stakeholder requirements through validation" ([ISO 02a], p. 29);

– "the purpose of the integration process is to assemble a system that is consistent with the architectural design. This process combines system elements to form complete or partial system configurations in order to create a product specified in the system requirements" ([ISO 02a], p. 30);

– "the purpose of the verification process is to confirm that the specified design requirements are fulfilled by the system. This process provides the information required to effect the remedial actions that correct non-conformances in the realized system or the processes that act on it" ([ISO 02a], p. 31);

– "the purpose of the transition process is to establish a capability to provide services specified by stakeholder requirements in the operational environment. This process installs a verified system, together with relevant enabling systems, e.g. operating system, support system, operator training system, user training system, as defined in agreements" ([ISO 02a], p. 32);

– "the purpose of the validation process is to provide objective evidence that the services provided by a system when in use comply with stakeholders' requirements. This process performs a comparative assessment and confirms that the stakeholders'

requirements are correctly defined. Where variances are identified, these are recorded and guide corrective actions. System validation is ratified by stakeholders” ([ISO 02a], p. 33);

- “the purpose of the operation process is to use the system in order to deliver its services. This process assigns personnel to operate the system, and monitors the services and operator-system performance. In order to sustain services it identifies and analyzes operational problems in relation to agreements, stakeholder requirements and organizational constraints” ([ISO 02a], p. 35);

- “the purpose of the maintenance process is to sustain the capability of the system to provide a service. This process monitors the system’s capability to deliver services, records problems for analysis, takes corrective, adaptive, perfective and preventive actions and confirms restored capability” ([ISO 02a], p. 36);

- “The purpose of the disposal process is to end the existence of a system entity. This process deactivates, disassembles and removes the system and any waste products, consigning them to a final condition and returning the environment to its original or an acceptable condition. This process destroys, stores or reclaims system elements and waste products in an environmentally sound manner, in accordance with legislation, agreements, organizational constraints and stakeholder requirements. Where required, it maintains records in order that the health of operators and users, and the safety of the environment, can be monitored” ([ISO 02a], p. 37).

This standard includes appendixes, which help adapt it to satisfy specific circumstances and factors such as the operational environment’s stability and variety, commercial or performance risks, novelty, size and complexity.

Stages describe the requirements for the system life cycle. The standard describes six stages as examples:

- “The concept stage begins with initial recognition of a need or a concept for a new system-of-interest or for the modification to an existing system-of-interest. This is an initial exploration, fact finding, and planning period, when economic, technical, strategic, and market bases are assessed through acquirer/market survey, feasibility analysis and trade-off studies. Acquirer/user feedback to the concept is obtained” ([ISO 02a], p. 42).

- “The development stage begins with sufficiently detailed technical refinement of the system requirements and the design solution and transforms these into one or more feasible products that enable a service during the utilization stage. The system-of-interest may be a prototype in this stage. The hardware, software and operator interfaces are specified, analyzed, designed, fabricated, integrated, tested and evaluated, as applicable, and the requirements for production, training and support

facilities are defined. This stage also ensures that the aspects of future stages (production, utilization, support, and retirement) and their enabling systems' requirements and capabilities are considered and incorporated into the design through the involvement of all interested parties. Feedback is obtained from stakeholders and those who will produce, operate, use, support, and retire the system-of-interest. Outputs are a system-of-interest or a prototype of the final system-of-interest, refined enabling systems or the enabling systems themselves and all documentation and cost estimates of future stages" ([ISO 02a], p. 43).

- "The production stage begins with the approval to produce the system-of-interest. The system-of-interest may be individually produced, assembled, integrated, and tested, as appropriate, or may be mass-produced. Planning for this stage begins in the preceding stage. Production may continue throughout the remainder of the system life cycle. During this stage, the product may undergo enhancements or redesigns, the enabling systems may need to be reconfigured and production staff re-trained in order to continue evolving a cost effective service from the stakeholder viewpoint" ([ISO 02a], p. 45).

- "The utilization stage begins after installation and transition to use of the system. The utilization stage is executed to operate the product at the intended operational sites to deliver the required services with continued operational and cost effectiveness. Planning for this stage begins in the preceding Stage. This stage includes those processes related to use of the product to provide services, as well as monitoring performance and identifying, classifying and reporting of anomalies, deficiencies, and failures. The response to identified problems includes taking no action; maintenance and minor (low cost/temporary) modification (reference support stage); major (permanent) modification and system-of-interest life extensions (reference development and production stages), and end-of-life retirement (reference retirement stage). During this stage the product or services can evolve giving rise to different configurations. The user operates the different configurations and the responsible product supplier manages the status and descriptions of the various versions and configurations of the product or services in use. It is presumed that the organization has available the operational infrastructure which includes facilities, equipment, trained personnel, and instruction manuals and procedures. These items are developed or acquired in order to be available when needed to support utilization" ([ISO 02a], p. 45).

- "The support stage begins with the provision of maintenance, logistics and other support for the system-of-interest's operation and use. Planning for this stage begins in the preceding stages. The support stage is completed with the retirement of the system-of-interest and termination of support services. This stage includes those processes related to operating the support system and providing support services to users of the system-of-interest. This stage also includes monitoring performance of the support system and services and the identification, classification, and reporting

of anomalies, deficiencies, and failures of the support system and services. The response to identified problems includes taking no action; maintenance and minor (low cost/temporary) modification (reference support stage); major (permanent) modification and system-of-interest life extensions (reference development and production stages), and end-of-life retirement (reference retirement stage). During this stage the support system and services can evolve under different versions or configurations. The support organization operates the different versions or configurations and the responsible product organization manages the status and descriptions of the various versions and configurations of the support system and services in use. It is presumed that the organization has available the support which includes the support sites, facilities, equipment and tools, trained support personnel, and maintenance manuals and procedures. The items making up the support infrastructure are developed and acquired in order to be ready when needed to support the system-of-interest” ([ISO 02a], p. 46).

– “The retirement stage provides for the removal of a system-of-interest and related operational and support services. Planning for this stage begins in the preceding stages. This stage begins when a system-of-interest is taken out of service. This stage includes those processes related to operating the retirement system and also includes monitoring performance of the retirement system and the identification, classification, and reporting of anomalies, deficiencies, and failures of the retirement system. Actions to be taken as a result of identified problems include maintenance and minor modification of the retirement system (reference support stage), major modification of the retirement system (reference development and production stages), and end-of-life retirement of the retirement system itself (reference retirement stage). It is presumed that the organization has access to an infrastructure to support retirement, including retirement facilities, tools and equipment, personnel trained in retirement actions, retirement procedures and, as appropriate, access to recycling, disposal or containment facilities. The items making up the retirement infrastructure are developed and acquired in order to be ready when needed to perform retirement functions. This stage is applicable whenever a system-of-interest reaches its end-of-service life. Such end-of-service life can be the result of replacement by a new system, irreparable wear, catastrophic failure, no further use to the user, or not cost effective to continue operating and supporting the system-of-interest” ([ISO 02a], p. 47).

Finally, the standard describes the key-concepts of system engineering, namely:

– the concept of system ([ISO 02a], p. 52):

- “the systems considered in this international standard are man-made, created and utilized to provide services in defined environments for the benefit of users and other stakeholders. These systems may be configured with one or more of the following: hardware, software, humans, processes (e.g. review process), procedures

(e.g. operator instructions), facilities and naturally occurring entities (e.g. water, organisms, minerals). In practice, they are thought of as products or services” ([ISO 02a], p. 52);

- since a system is composed of interacting components, each of these components may be apprehended as a system, with its own life cycle, which might be the object of a separate project;

- the contributing systems are the systems which, with no direct link to the needs and operational environment of the concerned system, contribute to one stage or another, and provide services to the concerned system; contributing systems can be systems which verify and qualify the concerned system, or training systems such as flight simulators;

- the life cycle concepts:

- “every system has a life cycle. A life cycle can be described using an abstract functional model that represents the conceptualization of a need for the system, its realization, utilization, evolution and disposal. The detail in the life cycle model is expressed in terms of these processes, their outcomes, relationships and occurrence” ([ISO 02a], p. 56);

- “the stages represent the major life cycle periods associated with a system and they relate to the state of the system description or the system itself. The stages describe the major progress and achievement milestones of the system through its life cycle” ([ISO 02a], p. 56);

- like every system, the contributing systems have their own life cycle, which is synchronized with the life cycle of the system-of-interest;

- the process concepts ([ISO 02a], p. 58):

- “the life cycle processes defined in this international standard can be used by any organization when acquiring and using and when creating and supplying a system. They can be applied at any level in a system’s hierarchy and at any stage in the life cycle. The life cycle processes are based on principles of modularity (maximal cohesiveness of the functions of a process and minimal coupling among processes) and ownership (a process is associated with a responsibility). The functions these processes perform are defined in terms of specific purposes, outcomes and the set of activities that constitute the process” ([ISO 02a], p. 58);

- the processes which concern responsibilities and agreements within and between organizations include all the processes implemented between the system’s acquirer and supplier, but also within their respective organizations. The processes are as follows:

- organizations may acquire or supply systems; a given organization may supply a system to its customers, and acquire constitutive or contributing systems from other organizations;
- the enterprise processes are strategic, management processes, dedicated to improving the enterprise's business;
- the project processes manage the resources and assets which the enterprise allocates to the project, and concern project management in terms of cost, delay, etc.;
- the technical processes concern the technical activities which are led all through the system's life cycle, transforming the stakeholders' needs into products and services, and which are applied to create and operate the system;
- the implementation of these processes is influenced by multiple factors, social, commercial, technical, organizational, etc. These processes can be implemented in a concurrent, iterative and recursive way.

12.23. Appendix N. Standard relative to the processes for engineering a system

The standard on processes for engineering a system features thirty-three requirements. These requirements are structured into five requirements groups, each relative to the implemented processes, each featuring a certain number of activities.

These groups, activities and requirements are:

- “the acquisition and supply processes are used by a developer to arrive at an agreement with another party to accomplish specific work and to deliver required products, or with another party or parties to have work done and to obtain desired products” ([EIA 98], p. 6):
 - “the supply process is used by the developer when acting as a supplier to establish and satisfy an agreement with the acquirer” ([EIA 98], p. 7);
 - product supply requirement; “for a system, or portion thereof, supplied to an acquirer, the developer (when acting as the supplier) shall establish and satisfy an agreement with the acquirer” ([EIA 98], p. 7);
 - “the acquisition process is used by the developer when acting as an acquirer to establish an agreement with a supplier and to manage supplier performance” ([EIA 98], p. 8);
 - product acquisition requirement; “for a system, or portion thereof, acquired from a supplier, the developer (when acting as the acquirer) shall establish an agreement with that supplier” ([EIA 98], p. 8);

- supplier performance requirement; “the developer (when acting as the acquirer) shall manage supplier performance to ensure that the technical effort to be accomplished by the supplier provides end products that satisfy the assigned requirements” ([EIA 98], p. 9);

- “the technical management processes are to be used to plan, assess, and control the technical work efforts required to satisfy the established agreement” ([EIA 98], p. 9):

- “the planning process is used to support enterprise and project decision making and to prepare necessary technical plans that support and complement project plans” ([EIA 98], p. 10);

- “process implementation strategy requirement; “the developer shall define a strategy for implementing the adopted processes of this Standard as a basis for project technical planning and that is in accordance with the agreement” ([EIA 98], p. 10);

- technical effort definition requirement; “the developer shall define a technical effort that is in accordance with the process implementation strategy” ([EIA 98], p. 11);

- schedule and organization requirement; “the developer shall schedule and organize the defined technical effort” ([EIA 98], p. 12);

- technical plans requirement; “the developer shall create technical plans to ensure an integrated and cost effective technical effort in accordance with the defined schedule and organization” ([EIA 98], p. 13);

- work directives requirement; “the developer shall create work directives that implement the planned technical effort” ([EIA 98], p. 13);

- “the assessment process is used to: (1) determine progress of the technical effort against both plans and requirements; (2) review progress during technical reviews; and (3) support control of the engineering of a system. The product and process metrics selected for assessing progress should provide information for risk aversion, meaningful financial and non-financial performance, and support of project management” ([EIA 98], p. 14);

- “progress against plans and schedules; the developer shall assess the progress of the technical effort against applicable technical plans and schedules” ([EIA 98], p. 14);

- “progress against requirements; the developer shall assess the progress of system development by comparing currently defined system characteristics against requirements” ([EIA 98], p. 15);

- “technical reviews; the developer shall conduct technical reviews of progress and accomplishments in accordance with appropriate technical plans” ([EIA 98], p. 15);

- “the control process is used to: (1) manage the conduct and outcomes of the acquisition and supply processes, system design processes, planning and assessment processes, product realization processes, and technical evaluation processes; (2) monitor variations from the plan and anomalies relative to requirements; (3) distribute required and requested information; and (4) ensure necessary communications” ([EIA 98], p. 16);

- outcomes management requirement; “the developer shall manage the outcomes of the technical effort” ([EIA 98], p. 17);

- information dissemination requirement; “the developer shall ensure that required and requested information is disseminated in accordance with the agreement, project plans, enterprise policies, and enterprise procedures” ([EIA 98], p. 17);

- “the system design processes are used to convert agreed-upon requirements of the acquirer into a set of realizable products that satisfy acquirer and other stakeholder requirements” ([EIA 98], p. 19):

- “the requirements definition process is used to transform stakeholder requirements into a set of system technical requirements” ([EIA 98], p. 20);

- “acquirer requirements; the developer shall define a validated set of acquirer requirements for the system, or portion thereof” ([EIA 98], p. 20);

- “other stakeholder requirements; the developer shall define a validated set of other stakeholder requirements for the system, or portion thereof” ([EIA 98], p. 21);

- “system technical requirements; the developer shall define a validated set of system technical requirements” ([EIA 98], p. 22);

- “the solution definition process is used to generate an acceptable design solution” ([EIA 98], p. 22);

- “logical solution representations; the developer shall define one or more validated sets of logical solution representations that conform with the technical requirements of the system” ([EIA 98], p. 23);

- “physical solution representations; the developer shall define a preferred set of physical solution representations that agrees with the assigned logical solution representations, derived technical requirements, and system technical requirements” ([EIA 98], p. 24);

- “Specified Requirements; the developer shall specify requirements for the design solution” ([EIA 98], p. 26);

– “the product realization processes are used to: (1) convert the specified requirements and other design solution characterizations into either a verified end product or a set of end products in accordance with the agreement and other stakeholder requirements; (2) deliver these to designated operating, customer, or storage sites; (3) install these at designated operating sites or into designated platforms; and (4) provide in-service support, as called for in an agreement” ([EIA 98], p. 26-27);

- “the implementation process requires transforming the characterized design solution into an integrated end product that conforms to its specified requirements” ([EIA 98], p. 27);

- implementation requirement; “the developer shall implement the design solution in accordance with the specified requirements to obtain a verified end product” ([EIA 98], p. 27);

- “the transition to use process results in products delivered to the appropriate destinations, in the required condition for use by the acquirer, and for the appropriate training of installers, operators, or maintainers of the products” ([EIA 98], p. 28);

- transition to use requirement; “the developer shall transition verified products to the acquirer of the products in accordance with the agreement” ([EIA 98], p. 28);

– “the technical evaluation processes are intended to be invoked by one of the other processes for engineering a system” ([EIA 98], p. 29):

- “the systems analysis process is used to: (1) provide a rigorous basis for technical decision making, resolution of requirement conflicts, and assessment of alternative physical solutions; (2) determine progress in satisfying system technical and derived technical requirements; (3) support risk management; and (4) ensure that decisions are made only after evaluating the cost, schedule, performance, and risk effects on the engineering or reengineering of the system” ([EIA 98], p. 30);

- effectiveness analysis requirement; “the developer shall perform effectiveness analyses to provide a quantitative basis for decision making” ([EIA 98], p. 31);

- tradeoff analysis requirement; “the developer shall perform tradeoff analyses to provide decision makers with recommendations, predictions of the results of alternative decisions, and other appropriate information to allow selection of the best course of action” ([EIA 98], p. 32);

- risk analysis requirement; “the developer shall perform risk analyses to develop risk management strategies, support management of risks, and support decision making” ([EIA 98], p. 33);

- “the requirements validation process is critical to successful system product development and implementation” ([EIA 98], p. 33);
- “requirement statements validation; the developer shall ensure that technical requirement statements and specified requirement statements, individually and as sets, are well formulated” ([EIA 98], p. 34);
- “acquirer requirements validation; the developer shall ensure that the set of defined acquirer requirements agrees with acquirer needs and expectations” ([EIA 98], p. 35);
- “other stakeholder requirements validation; the developer shall ensure that the set of defined other stakeholder requirements agrees with other stakeholder needs and expectations with respect to the system” ([EIA 98], p. 35);
- “system technical requirements validation; the developer shall ensure that the set of defined system technical requirements agrees with the validated acquirer and other stakeholder requirements” ([EIA 98], p. 36);
- “logical solution representations validation; the developer shall ensure that each set of logical solution representations agrees with the appropriately assigned subset of system technical requirements” ([EIA 98], p. 36);
- “the system verification process is used to ascertain that: (1) the system design solution generated by implementing Requirement 19 is consistent with its source requirements (selected preferred physical solution representation); (2) end products at each level of the system structure implementation, from the bottom up, (see Clause 6) meet their specified requirements; (3) enabling product development or procurement for each associated process is properly progressing; and (4) required enabling products will be ready and available when needed to perform” ([EIA 98], p. 37);
- design solution verification requirement; “the developer shall verify that each end product defined by the system design solution conforms to the requirements of the selected physical solution representation” ([EIA 98], p. 38);
- end product verification requirement; “the developer shall verify that an end product to be delivered to an acquirer conforms to its specified requirements” ([EIA 98], p. 39);
- enabling product readiness requirement; “the developer shall determine readiness of enabling products for development, production, test, deployment/installation, training, support/maintenance, and retirement or disposal” ([EIA 98], p. 40);
- “the end products validation process is used to demonstrate that the products to be delivered, or that have been delivered, satisfy the validated acquirer requirements” ([EIA 98], p. 41);

- end products validation requirement; “the developer shall ensure that an end product, or an aggregation of end products, conforms to its validated acquirer requirements” ([EIA 98], p. 41).

Moreover, the standard describes a set of environmental factors which affect the aforementioned processes. It differentiates between enterprise, project, and external factors, as well as the influence exerted by the other projects led by the enterprise.

Like the ISO/IEC 15288, the standard defines a number of key concepts:

- “The System Concept [...] consists of both the end products to be used by an acquirer for an intended purpose and the set of enabling products that enable the creation, realization, and use of an end product, or an aggregation of end products” ([EIA 98], p. 45).

- “The system forms the basis for a larger structure, called the building block. The building block provides the framework for application of the processes” ([EIA 98] p. 46).

- “The system element of the building block is the object for which the developer defines the acquirer and other stakeholder requirements using the Requirements Definition Process” ([EIA 98], p. 47).

- “The end products perform the operational functions for the system” ([EIA 98], p. 47).

- “Enabling products perform the associated process or non-operational functions of the system” ([EIA 98], p. 47).

Finally, the standard offers a normative glossary (e.g. acquirer, building block, derived requirements, etc.).

12.24. Appendix O. Standard for the application and management of the systems engineering process

The standard for the application and management of the systems engineering process concerns this process’s requirements ([IEE 05], p. 47). It features 14 general requirements on:

- “systems engineering process” ([IEE 05], p. 11);

- “the enterprise shall develop and maintain policies and procedures for governing the conduct of the SEP. These policies and procedures specify requirements for the planning, implementation, and control of product and process development and human/systems integration” ([IEE 05], p. 11);

– “the project shall prepare and implement the technical plans and schedules necessary to guide the project toward accomplishment of its objectives and proper conclusion” ([IEE 05], p. 12);

– “the project should explore development strategies for developing the system and its capabilities (e.g. waterfall, incremental, evolutionary, and spiral). The capacity to change or enhance the product and life cycle processes should be designed into the system architecture to enable the cost-effective sustainment of the system throughout its life cycle. This design attribute should be established early in the system development to provide a basis for planning each incremental development effort. Evolutionary development strategies should address approaches for managing the introduction of new technologies, evolving requirements, or enhancing product capabilities” ([IEE 05], p. 13);

– “the project should determine and establish suitable models, simulations, or prototypes to support analysis and project decision making” ([IEE 05], p. 13);

– “the enterprise shall capture pertinent design data in a repository for the evolving integrated data package and to provide a shared resource for the exchange and reuse of technical information” ([IEE 05], p. 13);

– “the project shall generate an integrated data package that documents architecture and design information for the support of life cycle processes” ([IEE 05], p. 14);

– “the project shall generate a specification tree modeled after the design architecture appropriate to the level of development. The specification tree is composed of specification elements and interface specifications. Interface specifications document the interface requirements among interacting elements. System interface specifications define interfaces with external systems, platforms, and products. Subsystem interface specifications define interfaces among subsystems, including hardware-hardware, hardware-software, software-software, human-human¹⁰, human-hardware, and human-software interfaces” ([IEE 05], p. 16);

– “the project shall generate a drawing tree to reflect the drawings associated with the hardware elements of the design architecture” ([IEE 05] p. 16);

– “the project shall generate a system breakdown structure to depict the hierarchy of products and processes that comprise the system architecture” ([IEE 05] p. 17);

10. The author is reserved and does not share the same point of view concerning human-human interfaces. Communications and interactions between humans are much more complex and cannot be standardized by interface specifications.

- “the project integrates the various inputs of the engineering and business specialties into the systems engineering effort to meet project objectives” ([IEE 05], p. 17);

- “the project shall conduct technical reviews, to include design reviews (e.g. system, subsystem, component, life cycle processes, test readiness, production approval) and audits (e.g. functional and design configuration), for the purpose of assessing technical progress. Normally, a design review should be conducted at the completion of each application of the SEP” ([IEE 05], p. 19);

- “the enterprise and project shall apply quality-management procedures for the development of products and life cycle processes” ([IEE 05], p. 19);

- “the enterprise and project should establish and maintain product and process quality factors in order to continuously improve products and processes throughout the system life cycle in a manner consistent with enterprise objectives” ([IEE 05], p. 19).

The application of systems engineering all through the systems’ life cycle is standardized into the six following stages:

- “The project should execute the system definition stage to establish the definition of the system with a focus on system products required to satisfy operational requirements. The major events of this stage should include completion of system, product, and subsystem interface specifications, system and product specifications, and preliminary subsystem specifications; establishment of a system baseline; and completion of technical reviews appropriate to the system definition stage. The documentation produced during system definition is required to guide subsystem developments. The technical reviews should evaluate the maturity of the system development and the readiness to progress to subsystem definition” ([IEE 05], p. 21).

- “The project should execute the preliminary design stage to initiate subsystem design and create subsystem-level specifications and design-to baselines to guide component development. The project applies the SEP for the purpose of decomposing identified subsystem functions into lower-level functions and allocating functional and performance requirements to component-level functional and physical architectures in accordance with the following subclauses” ([IEE 05], p. 25).

- “The project should execute the detailed design stage of the system life cycle to complete subsystem design down to the lowest component level and create a component specification and build-to component baseline for each component. The outputs of this stage are used to guide fabrication of preproduction prototypes for development test” ([IEE 05], p. 29).

– During the Fabrication, assembly, integration, and test stage, the project resolves “product deficiencies when specifications for the system, product, subsystem, assembly, or component are not met, as determined by inspection, analysis, demonstration, or test. The purpose of the FAIT stage of subsystem definition is to verify that the products designed satisfy specifications” ([IEE 05], p. 32).

– During the production and support stages, the process corrects “deficiencies discovered during production, assembly, integration, and acceptance testing of products and/or life cycle process products. The project also applies the SEP during support to evolve the product to implement an incremental change, resolve product or service deficiencies, or to implement planned evolutionary growth” ([IEE 05], p. 34).

– “Simultaneous engineering of life cycle processes: The project should accomplish planning activities and apply the SEP (system engineering process) to develop life cycle processes and services for system product development, production, test, distribution, support, training, and disposal. Life cycle processes and services include such items as special tooling and equipment for manufacturing or maintenance; special processes for manufacturing” ([IEE 05], p. 36).

These system life cycle processes include the following:

– Requirements analysis which establishes “what the system will be capable of accomplishing; how well system products are to perform in quantitative, measurable terms; the environments in which system products operate; the requirements of the human/system interfaces; the physical/aesthetic characteristics; and constraints that affect design solutions. The market needs, requirements, and constraints are derived from stakeholder expectations, project and enterprise constraints, external constraints, and higher-level system requirements. The requirements baseline guides the remaining activities of the SEP and represents the definition of the problem to be solved” ([IEE 05] p. 37). Requirements analysis features the following activities:

- define stakeholder expectations;
- define project and enterprise constraints;
- define external constraints;
- define operational scenarios;
- define measures of effectiveness;
- define system boundaries;
- define utilization environment;
- define life cycle process concepts;

- define functional requirements;
- define performance requirements;
- define modes of operation;
- define technical performance measures;
- define design characteristics;
- define human factors;
- establish requirements baseline.

– Requirements validation, on the one hand, verifies that the requirements baseline “represents identified stakeholder expectations and project, enterprise, and external constraints”, and on the other hand, determines “whether the full spectrum of possible system operations and system life cycle support concepts has been adequately addressed. When voids in needs, constraints, etc., are identified or needs are not properly addressed, requirements analysis and validation are repeated until a valid requirements baseline is generated. The validated requirements baseline is documented in the integrated repository and is an input to functional analysis” ([IEE 05], p. 43). Requirements validation features the following activities:

- compare requirements baseline to stakeholder expectation;
- compare requirements baseline to enterprise and project constraints;
- compare requirements baseline to external constraints;
- identify variances and conflicts;
- establish validated requirements baseline.

– “The project shall perform the tasks of functional analysis to accomplish the following two related objectives: to describe the problem defined by requirements analysis in clearer detail, and decompose the system functions to lower-level functions that should be satisfied by elements of the system design. This is accomplished by translating the validated requirements baseline into a functional architecture. The functional architecture describes the functional arrangements and sequencing of subfunctions resulting from decomposing (breaking down) the set of system functions to their subfunctions. Functional analysis should be performed without consideration for a design solution” ([IEE 05], p. 45). Functional analysis features the following activities:

- functional context analysis;
- analyze functional behavior;
- define functional interfaces.

– “The project shall conduct the tasks of functional verification to assess the completeness of the functional architecture in satisfying the validated requirements baseline and to produce a verified functional architecture for input to synthesis” ([IEE 05], p. 48). Functional verification features the following activities:

- define verification procedures;
- conduct verification evaluation;
- identify variances and conflicts;
- establish verified functional architecture.

– “The project shall perform the tasks of synthesis for the purpose of defining design solutions and identifying subsystems to satisfy the requirements of the verified functional architecture. Synthesis translates the functional architecture into a design architecture that provides an arrangement of system elements, their decomposition, interfaces (internal and external), and design constraints. The activities of synthesis involve selecting a preferred solution or arrangement from a set of alternatives and understanding associated cost, schedule, performance, and risk implications” ([IEE 05], p. 49). Synthesis features the following activities:

- group and allocate functions;
- identify design solution alternatives;
- assess safety and environmental hazards;
- assess life cycle quality factors;
- assess technology requirements;
- define design and performance characteristics;
- define physical interfaces;
- identify standardization opportunities;
- identify off-the-shelf availability;
- identify make-or-buy alternatives;
- develop models and prototypes;
- assess failure modes, effects, and criticality;
- assess testability needs;
- assess design capacity to evolve;
- finalize design;
- initiate evolutionary development;

- produce integrated data package;
- establish design architecture.

– “The project shall perform the tasks of design verification for the purpose of assuring that the requirements of the lowest level of the design architecture, including derived requirements, are traceable to the verified functional architecture, and that the design architecture satisfies the validated requirements baseline” ([IEE 05], p. 53). Design verification features the following activities:

- select verification approach;
- conduct verification evaluation;
- identify variances and conflicts;
- verify design architecture;
- verify design architectures of the life cycle process;
- verify system architecture;
- establish specifications and configuration baselines;
- develop system breakdown structure.

– “The project shall perform the tasks of systems analysis for the purpose of resolving conflicts identified during requirements analysis, decomposing functional requirements and allocating performance requirements during functional analysis, evaluating the effectiveness of alternative design solutions and selecting the best design solution during synthesis, assessing system effectiveness, and managing risk factors throughout the systems engineering effort. Systems analysis provides a rigorous quantitative basis for establishing a balanced set of requirements and for ending up with a balanced design” ([IEE 05], p. 45). System analysis features the following activities:

- assess requirement conflicts;
- assess functional alternatives;
- assess design alternatives;
- identify risk factors;
- define trade-off analysis scope;
- conduct trade-off analysis;
- select risk-handling options;
- select alternative recommendation;
- trade-offs and impacts;

- design effectiveness assessment.

- “The project shall perform the tasks of control for the purpose of managing and documenting the activities of the SEP. Outputs and test results, the planning for the conduct of the SEP activities (engineering plan, master schedule, and detail schedule), and technical plans generated by engineering specialties are controlled by the project. The control tasks provide the following: a complete and up-to-date picture of SEP activities and results, which are used in accomplishing other activities; planning for and inputs to future applications of the SEP; information for production, test, and support; information for decision makers at technical and project reviews” ([IEE 05], p. 61). Control features the following activities:

- technical management;
- track systems analysis and test data;
- track requirement and design changes;
- track progress against project plans;
- track progress against engineering plans;
- track product and process metrics;
- update specifications and configuration baselines;
- update requirements views and architectures;
- update engineering plans;
- update technical plans;
- design integrated repository.

The appendixes group elements which concern the role of system engineering within enterprises, an example of system engineering management plan, and finally, information to use this standard within the implementation of the ISO/IEC 15288 standard.

12.25. Appendix P. Standard relative to software life cycle processes

The standard on software life cycle processes (ISO/IEC 12207) recognizes three types of processes:

- “the primary life cycle processes [...] serve primary parties during the life cycle of software. A primary party is one that initiates or performs the development, operation, or maintenance of software products” ([ISO 95], p. 6);

- “the supporting life cycle processes [...] are employed and executed, as needed, by another process” ([ISO 95], p. 6);
- “the organizational life cycle processes [...] are typically employed outside the realm of specific projects and contracts” but are used to “establish and implement an underlying structure” ([ISO 95], p. 8).

The primary processes are as follows:

- “Acquisition process defines the activities of the acquirer [...] of a system, software product or software service” ([ISO 95], p. 6):
 - initiation;
 - request-for-proposal preparation;
 - contract preparation and update;
 - supplier monitoring;
 - acceptance and completion.
- “Supply process defines the activities of the supplier [who] provides the system, software product or software service” ([ISO 95] p. 6):
 - initiation;
 - preparation of response;
 - contract;
 - planning;
 - execution and control;
 - review and evaluation;
 - delivery and completion.
- “The Development Process contains the activities and tasks of the developer” ([ISO 95] p. 16):
 - process implementation;
 - system requirements analysis;
 - system architectural design;
 - software requirements analysis;
 - software architectural design;
 - software detailed design;

- software coding and testing;
- software integration;
- software qualification testing;
- system integration;
- system qualification testing;
- software installation;
- software acceptance support.

– “The operation process contains the activities and tasks of the operator. The process covers the operation of the software product and operational support to users” ([ISO 95], p. 23):

- process implementation;
- operational testing (tests on operation environment);
- system operation;
- user support;

– “The maintenance process contains the activities and tasks of the maintainer” ([ISO 95], p. 24):

- process implementation;
- problem and modification analysis;
- modification implementation;
- maintenance review/acceptance;
- migration;
- software retirement.

The supporting life cycle processes include the following:

– “The documentation process is a process for recording information produced by a life cycle process or activity” ([ISO 95], p. 28):

- implementation process;
- design and development;
- production;
- maintenance.

– “The configuration management process is a process of applying administrative and technical procedures throughout the software life cycle” ([ISO 95], p. 29):

- implementation process;
- configuration identification;
- configuration control;
- configuration status accounting;
- configuration evaluation;
- release management and delivery.

– “The quality assurance process is a process for providing adequate assurance that the software products and processes in the project life cycle conform to their specific requirements” ([ISO 95], p. 31):

- implementation process;
- product assurance;
- process assurance;
- assurance of quality systems.

– “The verification process is a process for determining whether the software products of an activity fulfill the requirements or conditions imposed on them in the previous activities” ([ISO 95], p. 33):

- implementation process;
- verification.

– “The validation process is a process for determining whether the requirements and the final, as-built system or software product fulfills its specific intended use” ([ISO 95], p. 36):

- implementation process;
- validation.

– “The joint review process is a process for evaluating the status and products of an activity of a project as appropriate” ([ISO 95], p. 38):

- implementation process;
- project management reviews;
- technical reviews.

– “The audit process is a process for determining compliance with the requirements, plans, and contract as appropriate” ([ISO 95], p. 39):

- implementation process;
- audit.

– “The problem resolution process is a process for analyzing and resolving the problems (including non-conformances), whatever their nature or source, that are discovered during the execution of development, operation, maintenance, or other processes” ([ISO 95], p. 41):

- implementation process;
- problem resolution.

Finally, the organizational processes include the following:

– The management process defines management activities, including project management, for the entire life cycle process ([ISO 95], p. 43):

- initiation and scope definition;
- planning;
- execution and control;
- review and evaluation;
- closure.

– “The infrastructure process is a process to establish and maintain the infrastructure needed for any other process” ([ISO 95], p. 45):

- implementation process;
- establishment of the infrastructure;
- maintenance of the infrastructure.

– “The improvement process is a process for establishing, assessing, measuring, controlling, and improving a software life cycle process” ([ISO 95], p. 46):

- implementation process;
- evaluation process;
- improvement process.

– “The training process is a process for providing and maintaining trained personnel” ([ISO 95] p. 47):

- implementation process;

- training material development;
- training plan implementation.

Appendixes help adapt these processes to the enterprise's needs.

12.26. Appendix Q. Standard relative to software measurement process

The purpose of this standard (ISO/IEC 15939) is to:

- establish and sustain the enterprise's measurement commitment;
- identify the data necessary to technical and management processes;
- identify and/or develop an appropriate set of measures, piloted by the aforementioned data;
- identify measurement activities;
- collect, stock and analyze the required data, and interpret the results;
- use the produced information to support the decision-making process and provide an objective communication baseline;
- evaluate the measurement process and the measures;
- communicate the improvements to the measurement process manager.

This standard defines the following measurement activities:

- establish and sustain measurement capacity:
 - establish commitments;
 - hire skilled personnel (recruitment, training);
- plan measurement:
 - identify the necessary information;
 - define priorities within the necessary information;
 - identify candidate measures;
 - select measures;
 - identify measurement criteria;
 - develop measurement plan;
 - acquire and install measurement tools;

- perform measurement:
 - collect data;
 - treat and verify data;
 - analyze and interpret data;
 - communicate analysis results;
- assess software measurement:
 - assess measurement process efficiency;
 - use the criteria identified during planning;
 - identify improvement opportunities;
 - assess measure efficiency;
 - use the criteria identified during planning;
 - identify the measures' positive results.

12.27. Appendix R. Standard relative to software product evaluation

This standard, which concerns software product evaluation, features the following parts:

- Part 1, general overview of ISO/IEC 14598 and ISO/IEC 9126:
 - structure of ISO/IEC 14598 and ISO/IEC 9126;
 - evaluation process;
 - support for evaluation;
 - software quality characteristics and metrics;
 - evaluation process;
 - establish evaluation requirements (establish the purpose of evaluation, identify types of product(s) to be evaluated, specify quality model);
 - specify the evaluation (select metrics, establish rating levels for metrics, establish criteria for assessment);
 - design the evaluation (produce evaluation plan);
 - execute the evaluation (take measures, compare with criteria, assess results);
 - supporting processes;

– Part 2, planning and management:

- evaluation management concepts;
- requirements and recommendations for supporting software evaluation;
- management at organizational level (planning the use and improvement of the evaluation technology, implementation of the evaluation technology, transfer of technology used for evaluation, assessment of the technology used for the evaluation, management of experiences);
- support for project management (support for evaluation planning, ongoing promotion of the quantitative evaluation plan, supporting the evaluation projects, collection of the evaluation results);
- quantitative evaluation plan template;

– Part 3, process for developers:

- evaluation concepts;
- user needs;
- external attributes;
- internal attributes;
- quality indicators;
- evaluation process;
- relation between evaluation and life cycle processes;
- evaluation process requirements;
- general requirements (organizational requirements, project requirements);
- establish evaluation requirements (quality requirements identification);
- specification of the evaluation (external quality requirements and internal quality requirements);
- design of the evaluation (planning the external evaluation and planning the internal evaluation);
- execution of the evaluation (internal evaluation, evaluation of the end product);
- quality evaluation review and feedback to the organization;

– Part 4, process for acquirers:

- software product evaluation - general considerations:

- correlation between evaluation and acquisition processes;
- inputs to the evaluation process (system requirements, integrity level requirements, software requirements specification, evaluations performed by others);
- tailoring;
- evaluation during acquisition of “off-the-shelf” software products:
 - step 1 - establish evaluation requirements (establish the purpose and scope of the evaluation, specify evaluation requirements);
 - step 2 - specify the evaluation (select metrics, select the evaluation methods, etc.);
 - step 3 - design the evaluation;
 - step 4 - execute the evaluation (execute the evaluation methods, analyze the evaluation results, draw conclusions);
- evaluation during acquisition of custom software and modifications to existing software:
 - step 1 - establish evaluation requirements;
 - step 2 - specify the evaluation;
 - step 3 - design the evaluation;
 - step 4 - execute the evaluation;
- evaluation methods;
- example of staged evaluation process;
- Part 5, process for evaluators:
 - evaluation concepts:
 - evaluation starting point (initial agreement, parties involved in the evaluation);
 - characteristics of the evaluation process;
 - evaluation process (evaluation activities, input to and output of the evaluation process);
 - relations between evaluation and life-cycle;
 - evaluation process requirements:
 - general requirements (organization and quality system, requester’s responsibilities, evaluator’s responsibilities);

- establishment of evaluation requirements (elaboration of the evaluation requirements, contents of the evaluation requirements, approbation and reporting, etc.);

- specification of the evaluation;
- design of the evaluation;
- execution of the evaluation;
- conclusion of the evaluation;
- template evaluation report;
- levels of evaluation;
- software product components;
- interactions between requester and evaluator;
- evaluation contract;

- Part 6, documentation of evaluation modules:

- evaluation module concept;
- format for documentation of an evaluation module:
 - module 0: foreword and introduction;
 - module 1: scope (characteristics, levels of evaluation, techniques, etc.);
 - module 2: references;
 - module 3: terms and definitions;

- module 4: inputs and metrics (input for the evaluation, data elements, metrics and measures);

- module 5: interpretation of results (mapping of measures, reporting);

- appendix: application procedure (resources required, evaluation instructions, documentation, etc.);

- development of evaluation modules;
- example of an evaluation module – fault density;
- example of an evaluation module – functionality;
- example of an evaluation module – usability and quality in use.

12.28. Appendix S. Standard on systems engineering, product and design data exchange

The requirements of the standard for systems product, engineering and design data exchange (AP233) are grouped within conformity classes dedicated to the following fields:

– “System modeling” ([ISO 07], “Application module: System modeling”, Part 1477) includes:

- “System behavior”;
- “System structure”;
- “Decision-making process support”;
- “Requirements, analysis and market studies”;
- “Verification and validation”;
- “Configuration management”;
- “Synthesis”.

– “Program management” ([ISO 07], “Application module: Program management”, Part 1466) includes:

- “Representation of the information to define and relate information associated with project management”;
- “Representation of the information to define and relate information associated with risk management”;
- “Representation of the information to define and relate information associated with issue management”;
- “Facility to assign project management information to program management data”;
- “Facility to assign risk management information to program management data”;
- “Facility to assign issue management information to program management data”.

– “System structure” ([ISO 07], “Application module: System structure”, Part 1450) includes:

- “Product data management”;
- “Product structure”;

- “Product breakdown”;
 - “Interface”;
 - “Product as individual”;
 - “System and hierarchy of systems”;
 - “Physical properties”;
 - “Assembly structure”;
 - “Part”;
 - “Value with Unit”.
- “Decision support” ([ISO 07], “Application module: Decision support”, Part 1486) includes:
- “Requirements”;
 - “Trade studies with measures of effectiveness”;
 - “Analysis and analysis representation”;
 - “Verification and validation”;
 - “Justification”;
- “Issue management” ([ISO 07], “Application module: Issue management”, Part 1489) includes:
- “Facility to assign contract information to issue management data”;
 - “Facility to assign product information to issue management data”;
 - “Facility to assign product as realized information to issue management data”;
 - “Facility to assign activity information to issue management data”;
 - “Facility to assign activity method information to issue management data”;
 - “Facility to assign person and organization information to issue management data”;
 - “Facility to assign approval information to issue management data”;
 - “Facility to assign certification information to issue management data”;
 - “Facility to assign project information to issue management data”;
 - “Facility to assign security information to issue management data”;
 - “Facility to assign event information to issue management data”;

- “Facility to assign value with unit information to issue management data”.
- “Project management” ([ISO 07], “Application module: Project management”, Part 1433) includes:
 - “Representation of the information to define and relate information associated with organization”;
 - “Representation of the information to define and relate information associated with project breakdown”;
 - “Representation of the information to define and relate information associated with work structure”;
 - “Facility to assign organization information to project management data”;
 - “Facility to assign project breakdown information to project management data”;
 - “Facility to assign work structure information to project management data”.
- “System behavior” ([ISO 07], “Application module: Risk management”, Part 1467) includes:
 - “Facility to assign property type information to risk management data”;
 - “Facility to assign product type information to risk management data”;
 - “Facility to assign activity type information to risk management data”;
 - “Facility to assign activity method type information to risk management data”;
 - “Facility to assign resource type information to risk management data”;
 - “Facility to assign document type information to risk management data”;
 - “Facility to assign person and organization type information to risk management data”;
 - “Facility to assign event type information to risk management data”;
 - “Facility to assign condition type information to risk management data”;
 - “Facility to assign justification type information to risk management data”;
 - “Facility to assign value with unit type information to risk management data”;
 - “Facility to assign approval type information to risk management data”;

- “Facility to assign state definition type information to risk management data”;
- “Facility to assign date and time type information to risk management data”.
- “System behavior” ([ISO 07], “Application module: System behavior”, Part 1448) includes:
 - “Function based behavior”, with elements to define behaviors and their hierarchy, the properties of shared resources;
 - “State based behavior, with the following elements: the definition of a versioned behavior, and the state-based behavior representation of a versioned behavior”;
 - “Functional breakdown”.

12.29. Appendix T. Standard on the exchange of product model data, products life cycle support

This application protocol dedicated to products life cycle support (AP239) features the following Data Exchange Sets (DEX)¹¹:

D001 – product breakdown for support includes the “exchange of the relationship of the parts assembly structure, derived from a PDM system”;

D002 – faults related to product structures concerns “the output from Fault Analysis programs in a form that can be used to identify required diagnostic and maintenance tasks, and to provide coherent fault reporting”;

D003 – task set describes “tasks to support a work plan”;

D004 – work package definition describes the “exchange and negotiation of a work package for a specific support opportunity including the list of required tasks, location, dates, products and resources”;

D005 – maintenance plan describes “the work required to sustain a product over time including the results of any Logistic Support Analysis”;

D007 – operational feedback concerns “the exchange of the observed configuration, location, state or properties of an actual product, and the communication of work requests to resolve issues arising from feedback on its usage”;

11. The data exchange sets are available at: <http://rdlserver.eurostep.com/>.

D008 – product as individual concerns the “exchange and collation of manufacturing and serialized part information and its relationship to the product assembly structure from which it derived”;

D009 – work package report concerns the “reporting of work completion against a work package definition”;

D010 – System requirements.

Each data exchange set features:

- an introduction;
- the business processes;
- a description of the business process that the DEX is supporting;
- the identification of the process in the AP239 activity model supported;
- usage guidance for the model;
- DEX specific Reference Data;
- the subset of the Information model supported by the DEX;
- EXPRESS information model;
- XML Schema (derived from the EXPRESS).

Finally, the Eurostep organization has developed a set of webservices for the AP239¹²:

- ChangeEvents.wsdl;
- ChangeManagement.wsdl;
- DocumentEvents.wsdl;
- DocumentManagement.wsdl;
- InformationCollectionManagement.wsdl;
- InLifeManagement.wsdl;
- MaintenanceManagement.wsdl;
- PartEvents.wsdl;
- PartManagement.wsdl;
- ProductCharacteristicsEvents.wsdl;

12. List of webservices at: <http://www.plcs-resources.org/plcsws/indexemplehtml>.

- QueryManagement.wsdl;
- RequirementEvents.wsdl;
- RequirementManagement.wsdl;
- SystemManagement.wsdl.

The schemes associated with these webservices are:

- Callback.xsd;
- ChangeEvents.xsd;
- ChangeMsg.xsd;
- CharacteristicsMsg.xsd;
- DocumentEvents.xsd;
- DocumentMsg.xsd;
- Exceptions.xsd;
- Headers.xsd;
- InLifeMsg.xsd;
- InformationCollectionMsg.xsd;
- MaintenanceMsg.xsd;
- PartEvents.xsd;
- PartMsg.xsd;
- ProductCharacteristicsEvents.xsd;
- ProductEvents.xsd;
- QueryMsg.xsd;
- RepresentingBusinessObjects.xsd;
- RequirementEvents.xsd;
- RequirementMsg.xsd;
- SubscriberProfile.xsd;
- System.xsd;
- SystemMsg.xsd;
- Types.xsd.

12.30. Bibliography

- [AFI 08] AFIS INTERNET SITE, available at: <http://www.afis.fr>, 2008.
- [AMA 09] AMALBERTI R., 2009, Violations et migrations ordinaires dans les interactions avec les systèmes automatisés (Violations and migrations occurring during interactions with automated systems), *Journal Européen des Systèmes Automatisés*, vol 43/6 - 2009 - pp.647-660.
- [AUT 07] AUTRAN F., CATTAN D., GARNIER J.L., LUZEAUX D., PEYRICHON M., RUAULT J.R., "Coupling component systems towards systems of systems", *Journées ICSSEA*, Paris, 2007.
- [AUT 08] AUTRAN F., AUZELLE, J.P., CATTAN D., GARNIER J.L., LUZEAUX D., PEYRICHON M., RUAULT J.R., "Coupling component systems towards systems of systems", *INCOSE 2008*, Utrecht, 2008.
- [EIA 98] EIA, *Processes for Engineering a System*, EIA 632-1998.
- [HAN 08] HANDLEY H., SMILLIE R., "Architecture framework human view: the NATO approach", *Systems Engineering*, vol. 11, n° 1, 2008.
- [HOL 06] HOLLNAGEL E., WOODS D., LEVENSON N., *Resilience Engineering: Concepts and Precepts*, Ashgate, Aldershot, 2006.
- [IEE 00] IEEE, *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, IEEE 1471-2000.
- [IEE 05] IEEE, *IEEE Standard for Application and Management of the Systems Engineering Process*, IEEE 1220-2005.
- [ISO 95] ISO/IEC, *Information Technologies – Software Life Cycle Processes*, ISO/IEC 12207: 1995.
- [ISO 99] ISO/IEC, *Information Technology – Software Product Evaluation – Part 1: General Overview*, ISO/IEC 14598-1:1999.
- [ISO 01] ISO/IEC, *Software Engineering – Product Quality – Part 1: Quality Model*, ISO/IEC 9126: 2001.
- [ISO 02a] ISO/IEC, *System Engineering - System Life Cycle Process*, ISO/IEC 15288: 2002.
- [ISO 02b] ISO/IEC, *Software Engineering – Measurement Process*, ISO/IEC 15939: 2002.
- [ISO 02c] ISO-TR, *Ergonomics of Human-system Interaction – Usability Methods Supporting Human-centered Design*, ISO-TR 16982: 2002.
- [ISO 05a] ISO/IEC, *Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*, ISO/IEC 25000: 2005.
- [ISO 05b] ISO/IEC, *Information technology – Security Techniques – Evaluation Criteria for IT Security – Part 1: Introduction and General Model*, ISO/IEC 15408-1: 2005.

- [ISO 05c] ISO/IEC, *Information technology – Security Techniques – Evaluation Criteria for IT Security – Part 2: Security Functional Requirements*, ISO/IEC 15408-2: 2005.
- [ISO 05d] ISO/IEC, *Information Technology – Security Techniques – Evaluation Criteria for IT Security – Part 3: Security Assurance Requirements*, ISO/IEC 15408-3: 2005.
- [ISO 05e] ISO/IEC, *Information Technology – Service Management – Part 1: Specification*, ISO/IEC 20000-1: 2005.
- [ISO 05f] ISO/IEC, *Information Technology – Service Management – Part 2: Code of Practice*, ISO/IEC 20000-2: 2005.
- [ISO 07] ISO/Cd, *Product Data Representation and Exchange: Application Protocol: System Engineering and Design*, ISO/CD 10303-233, 2007.
- [MEI 98] MEINADIER J.P., *Ingénierie et intégration des systèmes*, Hermès, Paris, 1998.
- [MEI 02] MEINADIER J.P., *Le métier d'intégration des systèmes*, Hermès, Paris, 2002.
- [NF 99] NF EN ISO, *Human Centred Design Processes for Interactive Systems*, NF EN ISO 13407: 1999.
- [OAS 07] OASIS, *Web Services Business Process Execution Language Version 2.0*, 2007, (details and specifications available at: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>).
- [OGC 05] OGC, ITIL, *Best Practice for Service Support*, Office Government Commerce, The Stationery Office (www.tsoshop.co.uk), 2005.
- [OGC 06] OGC, ITIL, *Best Practice for Service Supply*, Office Government Commerce, The Stationery Office (www.tsoshop.co.uk), 2006.
- [OMG 03] OMG, *UML 2.0, Unified Modeling Language: Superstructure*, reference ptc/03-08-02, details and specification available at: <http://www.omg.org/>, 2003.
- [OMG 06a] OMG, *BPMN, Business Process Modeling Notation Specification*, reference dtc/06-02-01, details and specification available at: <http://www.bpmn.org/>, 2006.
- [OMG 06b] OMG, *Ontology Definition Metamodel Specification*, reference ptc/06-10-11, details and specification available at: <http://www.omg.org/>, 2006.
- [OMG 06c] OMG, *OMG Systems Modeling Language (OMG SysML™) Specification*, reference ptc/06-05-04, details and specification available at: <http://www.omg.org/>, 2006.
- [OMG 07a] OMG, *UPDM, UML profile for the Department of Defense Architecture Framework (DoDAF) and the Ministry of Defence Architecture Framework (MODAF)*, reference c4i/2007-02-01, information available at: <http://syseng.omg.org/UPDM.htm>, 2007.
- [OMG 07b] OMG, *UML 2.1.1, Unified Modeling Language: Superstructure, reference UML Superstructure Specification, v2.1.1*, information available at: <http://www.omg.org/>, 2007.
- [RUA 04] RUAULT J.R., "System engineering and human factors engineering; bridging the processes", *Proceedings of INCOSE'04*, Toulouse, France, 15 pages (CD ROM), 2004.

- [RUA 09a] RUAULT J.R., “Adaptabilité des systèmes à logiciel prépondérant”, *Journal Européen des Systèmes Automatisés*, vol 43/6 - 2009 - pp.683-710.
- [RUA 09b] RUAULT, J-R., COLAS, C., SARRON, J-C., LUZEAUX, D., Ingénierie système et résilience des systèmes sociotechniques, *5^e Conférence Annuelle d’Ingénierie Système AFIS 2009*, Paris, France, 15 pages (CD ROM), 2009.
- [SCO 02] SCOTT W., SMITH J., JOHNSON J., *An Initial Application of AP-233*, 2002.
- [ZIF 07] ZACHMAN INSTITUTE FOR FRAMEWORK ADVANCEMENT., information available at: <http://www.zifa.com/framework.html>, 2007.

Conclusion

Introduction

Some authors, whose names we will politely omit, mix human beings, organizations and technical systems in hurried concepts, without marking the difference between their specific characteristics, within a rationale close to syncretism in which everything is linked, everything is in everything and vice versa. On the contrary, the 12 chapters of this book have demonstrated the richness, even the polysemy, of the notion of system. Such richness, such complexity, takes us all the farther from a basic definition.

Let us keep in mind a few important, structuring points which will have to be studied in depth to offer the systems of systems the proper concepts, methods and tools.

Evolutions and stakes

We have seen the characteristics of systems of systems, their constitutive systems, their organization in terms of architecture, and their operation modes.

If, at first, systems engineering treated the systems mostly, sometimes exclusively, in terms of *product*, recent evolutions, partly linked to the development of systems of systems, lead to the apprehension of systems in terms of *service*. These evolutions are not without consequence. As a service, the system modifies the contracting models, the processes and activities which must be implemented. In this way, we try to contractualize quality of service through time, in terms of performance, availability, absence of anomaly. Validation is not only performed

before the operational use of the system, but during all its operational exploitation. We do not purchase a system, rather we purchase its use, a service. We are still in a technological context when the service treats information; after a few adjustments, the same concepts are implemented. In the case of human services, however, in which the result depends on the quality of the interaction between the service's client and the supplier, and the object of transformation is not a technological object, but the service's client (health service, education service), we rapidly leave the purification of our concepts, methods and tools behind. It is no longer a problem of adaptation, but entails important evolutions. One of the first and main consequences concerns the contracting mode. The supplier doesn't guarantee the results, but the means. If we try to reduce the variables within the realization of a product or a technical service – which is, incidentally, what defines quality of service – it is much more difficult, or even perfectly illusory, in the context of human services. Of course, mechanisms, such as protocols, help control the variability inherent to any living system, but the abusive use of such mechanisms might penalize the interaction between the client and the service supplier and, *in fine*, deteriorates the quality of service more than it improves it. Indeed, the intersubjective relationships that come into play and largely contribute to the client's satisfaction can neither be modeled, nor simulated, with the engineer's concepts and mathematical tools. But are we still talking about the same thing?

We are facing a singular situation. The word “system” is polysemous. On the one hand, we have systems engineering, and on the other, systemics. Through lack of rigour, some authors mix them up, which we find deplorable, since this considerably reduces the capacity of progress and enrichment of the concerned disciplines.

Indeed, systems engineering is a discipline, the engineer's science to conceive systems. The systems are designed by human beings, heteronymous systems. The human designer is looking for complete control of the system, which must behave as has been defined. Risk analysis and reliability activities are led to plan different behaviors and design a set of solutions to control these behaviors. In this context, a system's autonomy resides in the elaboration, by the designer, of a defined set of rules and laws within the system, so that the latter can adapt its behavior according to the environment's demands. We cannot talk, *stricto sensu*, about the system's autonomy, since the system itself doesn't create its own rules¹. The highest control of behavior is associated with the reduction of variability. This concerns technological systems. If, in the various definitions we can find in literature, the human being belongs to the system, it is a fallacy, for such a thing is quickly eluded and none of the concepts, methods and tools of systems engineering offer the capacity of taking human behavior into account. This is not surprising. In a

1. Autonomous; from the Greek word *autonomos*, “which governs itself with its own rules” (new etymologique and historic dictionary, Larousse, 1971).

heteronomous system, in which the engineer is looking for the highest degree of control, the existence and standardization of the behavior of an autonomous agent are neither coherent nor easy. Apart from the heteronomy, the system is characterized by a clearly defined perimeter, which also marks the limits between the system and its environment, with specified interfaces between them. The system dimension resides, for the most part, in the fact that numerous acts of engineering are called upon to specify, conceive, design, validate, maintain and take apart that kind of system. Systems engineering is based on the analysis and the breaking down of the system into subsystems, in a recursive manner. This point is important, for it differentiates systems engineering from systemics². Since systems engineering is interested in the system to do and the processes and activities needed for that purpose, it is the counterpart of project management, which concerns the management of the means put to work to create this system, while respecting the deadlines and the financial envelope. We are in the field of control.

If systems engineering aims for mastery and control, systemics have another goal, to understand an ensemble of phenomena in the field of living forms and materials. The concerned systems are not designed by human beings, and display autonomous behavior. The question is raised as to what concerns organizations, both autonomous and designed by human beings. Apart from the autonomy of the observed systems, what differentiates systemics is the inability to analyze or break down the systems. Systemics tends to demonstrate other major properties of systems, such as the capacity to maintain, generate, reproduce and adapt oneself within a process of conjoint and interdependent evolution of the system and its environment. Moreover, these systems are also characterized by blurry boundaries. These major properties make the system *a priori* uncontrollable through the concepts, methods and tools of systems engineering. Finally, systemics deal with dimensions and concepts, such as the paradox, or the trust that people can have for each other, dimensions and concepts that are not familiar to engineers and the mathematical tools they have at their disposal, and which are however necessary in a human services rationale. Paul Watzlawick, who just left us a year ago, has largely contributed to the subject ([WAT 72, WAT 75, WAT 78]).

The situations which we currently have to deal with, such as the need to control the ecosystems' dynamics in order to avoid huge catastrophes, bring us to a paradox. On the one hand, the concepts, methods and tools of systemics which help report these complex phenomena, do not help control said phenomena. On the other hand, the concepts, methods and tools of systems engineering, which help control these

2. The definition of the notion of system, from the standpoint of systems engineering, belongs to what Morin calls the *system analysis* in systemic theory [MOR 05].

systems, are incapable of reporting and maintaining natural systems. Faced with this paradox, two solutions appear. The first one, frequently implemented, consists of using, without proper judgment or a reflection on the relevance of that action, the current concepts, methods and tools of systems engineering on the living, complex, autonomous and dynamic living systems. Despite justifications under cover of mastery of complexity, this first solution is basically akin to the “garbage can model” described by James March. The second solution, which we recommend, is more demanding and does not necessarily receive a warm welcome from a *quick and dirty* thinking. This solution consists of making the systems engineering corpus evolve, in order to design the conceptual tools necessary and sufficient to reach the objective of improved control over living systems, without however seeking illusory control, as defined by systems engineering.

How should these concepts, methods and tools evolve?

Obviously, it is not possible to validate a system, such as an organization, with a validation protocol, by theorizing that the organization will keep on behaving like it did during validation. It is just as obvious that for an ecosystem, there is no acquirer demonstrating needs and no supplier designing a system to fulfill said needs. Just as radically, we cannot treat the interaction between a system and its environment, in terms of interfaces and flows exchanges. Indeed, interaction, in keeping with the principles of coevolution, can modify the system’s organization and operation. The rules of command and control are far from embracing all the complexity structuring the laws of social regulation within a group of human beings.

If an answer cannot be reached in this conclusion, for it alone would require, at the very least, its own book, we can identify the actions which must be led. In the various domains described by systemics, it is necessary to identify the phenomena subsets which must be controlled by human beings in order to avoid ecological and human disasters. For each of these phenomena subsets, it is necessary, on the one hand, to understand, among other things, their dynamics, the interdependence they have to other phenomena, the modes of regulation governing them; on the other hand, it is necessary to define the objectives of control of these phenomena in terms of desired or feared evolutions. Then, it is necessary to make the concepts, methods and tools of systems engineering evolve, in order to control these phenomena, to evaluate the state of these phenomena and channel the evolution of the states of these phenomena depending, on the one hand, on the defined goals and on the other hand, on the evolution laws of these phenomena, e.g. chaotic.

These are the important, even critical, stakes. To reduce complexity through a Cartesian breakdown rationale is to condemn ourselves to failure.

What should be remembered?

The specification, design, realization, validation, and implementation of systems of systems put into sharp light the need to make the concepts, methods and tools of systems engineering, which help control human-designed systems, evolve, enriching them against the yardstick of rich production in the field of systems, and the first works on services science. Listening to the sirens who singsong that the current concepts are enough, is dooming ourselves to failure. Like Ulysses strung to the mast, the ones who will succeed are the ones who, with perseverance, will have made the concepts evolve and progress in order to reply to the challenges we are confronted with.

Bibliography

- [ABE 05] ABE T., *What is Services Science?*, The Fujitsu Research Institute, Research Report, n° 246, 2005.
- [MOR 05] MORIN E., *Introduction à la pensée complexe*, p. 28, Le Seuil, Paris, 2005.
- [WAT 72] WATZLAWICK P., HELMICK-BEAVIN J., JACKSON D., *Une logique de la communication*, Points Seuil, Paris, 1972.
- [WAT 75] WATZLAWICK P., WEAKLAND J., FISCH R., *Changements*, Points Seuil, Paris, 1975.
- [WAT 78] WATZLAWICK P., *La réalité de la réalité*, Essais Seuil, Paris, 1978

List of Authors

Michel CHAVRET
SETEC ITS
Lyon
France

Lui KAM
Cabinet Simon-Kucher & Partners
Paris
France

Dominique LUZEAUX
DGA
Bagneux
France

Jean-Pierre MEINADIER
Consultant
France

Patrice MICOUIN
Senior Consultant
Marseille
France

Frédéric PRADEILLES
CS Systèmes d'Information
Le Plessis-Robinson
France

Jean-René RUAULT
DGA
Bagneux
France

Danièle VÉRET
Cabinet Granrut Avocats
Paris
France

Jean-Luc ZOLESIO
THALES
Palaiseau
France

Index

A

accreditation, 354-355, 376
acquisition cycles, 6, 8
activity, 120-123, 160-161, 172-179,
189-190, 261-262, 432-435, 496-
497, 505-508
adaptability, 4, 7, 19, 79, 163, 170,
350, 378, 467
AP233, 56, 403, 430, 432, 504
architecture, 41, 230, 326, 331, 446,
451
architecture framework, 246, 332-
333, 407-413
artificial system, 13, 33, 95-96, 154,
316
asymmetrical conflicts, 8

B

business, 404, 406, 439, 443
business process modeling notation,
404

C

capability, 7-11, 36-47, 53-56, 68-69,
73-83, 236-239, 324-328, 333-339,
434-437, 446-450, 457-460, 466-
468, 477-478

co-evolution, 5, 7
collaboration, 19, 71, 127, 137, 172,
196, 198, 231, 254, 296, 323, 360,
418, 440
collaborative working environments,
296
commitment, 20, 74, 128, 130, 139,
187, 192, 202, 338, 499
common tactical situation, 19
communication technologies, 3, 5, 6,
16, 56, 208, 224, 333
communities, 3, 5, 17, 37, 304, 328,
359, 360, 374
competition, 27, 28, 62, 71, 76, 103-
104, 147, 165, 173, 299, 304, 364,
384
competitive bidding, 128-129, 142,
147
complexity, 5-10, 59-61, 107-111,
150-154, 160-161, 165-166, 175-
179, 323-326, 346-348, 366-368
component, 23-24, 30-31, 64-65, 80-
83, 209-210, 262-263, 329-333,
343-348, 389-390
concept analysis, 332
contract, 8, 35, 121-122, 125-148,
326, 333, 342, 415, 434, 453, 495,
498, 503, 505

contractual relationships, 3
 costs, 55-61, 338-339, 348-356, 371-372
 coupling, 9, 20, 37, 78, 214, 275, 338, 354, 386, 416, 481
 crisis management, 25, 173, 176, 178, 198, 208, 272, 273, 280-286
 cultural aspects, 186, 253, 292, 304
 culture, 164, 181-184, 189, 198, 201, 282, 289-290, 292, 307, 314-315, 357, 393

D

data, 406, 430, 504, 507
 decision making, 58, 67, 69, 150, 160-161, 167, 170, 175-178, 180-183, 188-189, 193, 195, 209, 483, 485, 488
 decomposition, 457-458, 492
 defense, 6-11, 14-17, 22, 43, 54, 56, 59, 62-70, 75, 121, 137, 156, 193, 208, 214-215, 278, 328, 333, 343, 345, 371, 376, 385-386, 391
 delays, 6-7, 44, 48, 55-58, 67, 72, 74, 127-130, 132, 193, 220, 296, 314, 338, 349, 351-352, 364, 371, 383, 392
 delivery, 5, 125-126, 130-134, 137, 142, 338, 340, 462-465, 473, 495, 497
 design, 61, 146, 160, 166, 236, 331, 363, 504
 development, 3, 245
 digital age, 3
 dismantling, 55, 424
 disposal, 10, 22, 39, 50, 52, 104, 108, 112, 146, 151, 174, 217, 240, 307-308, 326, 371-372, 425, 437, 478-481, 486, 490

E

embedded technologies, 6

emergence, 5, 13, 15-17, 29-30, 36, 41, 99, 138, 154, 157, 163-164, 173, 192, 194, 274, 296, 312, 363
 emergent behavior, 13-16, 32, 77, 82-83, 89, 95, 151, 157, 364, 367
 end-user, 54, 58, 72-73, 119, 213, 219, 267, 314, 384-385
 engineering, 34-57, 64-69, 111-115, 149-151, 315-316, 324-325, 336-344, 347-349, 364-365, 368-369, 374-381, 389-394, 399-404, 421-435, 457-458, 469-470, 480-485, 489-490
 process, 35, 40-41, 44, 49, 55-57, 64, 66, 78, 292, 315, 324, 336, 342, 353, 359, 390, 393, 399-404, 421-430, 433-434, 439, 470, 490
 tools, 325, 338, 430
 work, 68
 enterprise, 4, 10, 21, 28, 43, 72-73, 78, 80, 83, 101, 179, 192, 200, 226, 230, 296, 334, 386, 402, 409-412, 417, 425, 429, 441, 463, 474-476, 482-484, 487-491, 499
 environment, 3-8, 41-42, 48-49, 92-93, 96-100, 151-154, 161-180, 192-193, 201-202, 227-230, 297-298, 306-307, 351-356, 430-431, 464-471, 476-478
 event, 70, 91, 96, 100, 109, 127, 136, 139-141, 145-148, 161, 187, 248, 257, 262, 267, 272-275, 279, 281, 285-289, 327-328, 339-340, 346-353, 371, 390, 405, 435, 442, 444, 448-449, 461, 505-506
 evolutionary development, 77, 82, 89, 157, 436, 492
 extended enterprise, 3, 10, 16, 68
 extranets, 18

F

financial leasing, 122
 function, 25, 44, 46, 53, 55, 59, 65,
 76, 78, 80, 95, 109, 138, 171, 173,
 180, 183, 185, 210-211, 243, 270,
 308, 313, 343, 346-347, 353, 359,
 390-391, 460, 467, 471
 functional
 architecture, 37, 43-44, 55, 220,
 330, 379, 491-493
 view, 44, 229
 functionalities, 15, 24, 26, 39, 122,
 255, 264, 267, 273, 288, 313, 316,
 331-332, 337, 341, 345, 347, 358-
 359, 378, 385, 400, 412, 424

G

geographical distribution, 15-16, 78,
 82, 89, 354, 360
 globalization, 292-317
 guarantee, 9, 28, 33, 41, 44-45, 54,
 79, 122, 130, 135-137, 210, 266,
 276-277, 285, 327, 339, 347, 349,
 351, 353-355, 369, 373, 375, 378,
 392

H

healthcare, 25, 235-241, 245-251,
 254-258, 305, 308
 High Level Architecture, 345
 human factor, 58-59, 62-67, 76, 123,
 149, 160, 162, 164, 189, 195, 198,
 202, 239, 285, 306, 412, 491

I

implementation, 10-11, 25-26, 46-49,
 55-57, 70-71, 73-76, 184-185, 201-
 202, 229-231, 235-237, 244-248,
 253-257, 270-273, 338-340, 403-
 405, 430-434, 437-438, 450-453,
 462-465, 482-487, 494-501

incremental approach, 337, 357
 information, 3-6, 14-26, 48-49, 52-
 65, 77-81, 109-113, 123-125, 137-
 138, 145-146, 157-158, 161-164,
 174-175, 183-202, 207-208, 211-
 220, 223-231, 235-253, 256-258,
 275-289, 326-334, 345-347, 354-
 365, 392-395, 400-402, 406-416,
 432-434, 441-442, 445-453, 475-
 477, 483-485, 504-508
 information technologies, 14, 183-
 193, 199-201, 236-239, 245, 247,
 256, 275, 401, 412
 Information Technology
 Infrastructure Library, 403, 420,
 461
 innovation, 57-58, 72, 188, 291, 313,
 357, 360, 381
 integrated product, 162
 intellectual property rights, 120, 136,
 146
 interaction, 10-13, 31, 38, 48, 63, 65,
 96, 99, 151, 154, 180, 187, 199,
 235, 240, 258, 287, 325, 336, 354,
 357, 364, 405, 411, 430, 443, 447,
 457-459, 476
 interconnection, 18-19, 24-25, 43, 54,
 107, 111, 113, 208-212, 217, 224,
 275, 277, 332, 449, 457
 interdependence, 153, 171, 177-178,
 191, 243, 293, 295, 447
 interdependent components, 12
 interface, 45-46, 58-60, 330-338,
 488-489
 Internet, 15, 18, 24, 62, 89, 102, 105,
 111, 113, 127, 137, 145, 154-157,
 212, 231, 255, 277, 294, 296, 303,
 312, 360, 381, 414-416
 interoperability, 80-82, 113-115, 198-
 199, 215-220, 224-226, 245-248,
 282-285, 363-366, 369-382, 385-
 386, 400-401, 444-445
 intranets, 18

ISO/IEC 15288, 11, 34, 40, 111-112, 115, 324, 336, 339, 361, 403, 423-428, 434, 487, 494

L

legal

framework, 123, 198

qualification, 119

life cycle, 6-9, 39-40, 50-52, 55-56, 61-62, 323-325, 352-358, 371-372, 402-403, 423-435, 452-453, 473-482, 488-501

M

managerial independence, 14-16, 32, 78, 81-82, 89, 156-157, 400, 430, 437

market, 6, 11, 16-17, 26-27, 35, 57-58, 62, 76, 129, 147, 179, 211, 274, 302-304, 308, 312-313, 327, 352, 384, 478, 490, 504

metamodeling, 343, 365-370, 373, 378, 382-387, 393, 418

model, 71, 146, 229, 232, 323, 334, 344, 352, 363, 366, 378, 391, 404, 439, 444, 454, 457, 466, 507

model-driven engineering, 332, 346, 352, 360, 378, 385, 392-393

model transformation, 61, 343, 384, 386

mutual intelligibility, 161, 176, 186, 189, 196-197

N

need, 36, 56, 70

network, 3, 162, 190, 193, 195, 198, 245,

network-centric operations, 19, 30, 63, 150, 184, 189, 192, 193, 195-198

network-centric warfare, 18, 192

O

Object Management Group, 364, 404, 413

obligation, 69, 122, 127-129, 133, 135-139, 142, 361

off-the-shelf, 8-9, 21, 57, 69, 82, 492, 502

ontology, 90, 195, 252-253, 369, 373, 392, 407, 444

operational

capability, 8, 38

independence, 14, 17, 27, 81, 82, 89, 156-157, 216

view, 246, 333-334, 411-414, 446-451

operator, 58-60, 159-161, 281-282, 287-288, 477-478

optimization, 4, 6, 17, 23-29, 33-34, 38, 54, 60, 65, 72, 79, 82, 224-225, 237, 281, 284-285, 292, 298, 300-304, 330, 349

organization, 72, 159-160, 166, 173, 176, 180, 186, 190, 198, 201

organizational aspects, 180, 199, 266, 290, 336, 402

organizational culture, 182, 201

ownership, 3, 4, 46, 50, 59, 69, 105, 112, 120-121, 129, 136, 138, 142, 146, 394, 481

P

partnership, 4, 75, 101, 121, 126, 129, 147-148, 219, 226, 360, 420

physical architecture, 44, 251, 328, 380, 393, 489

process, 404, 422, 439, 443, 473, 482, 487, 494, 499,

procurement, 121, 486

product, 16-17, 27-29, 35-36, 55-57, 63-65, 240-241, 338-340, 374-375, 430-432, 466-469, 476-490, 494-497, 500-508

professional culture, 181, 243
 program management, 432, 446, 504
 project management, 51-52, 55-56,
 73-75, 436-439, 482-483, 497-498
 protection of information, 18
 public law, 129, 130, 134-135, 148

Q, R

quality of information, 19, 196
 regulation, 6, 48-49, 57-58, 201, 244,
 300, 303-311, 326
 relational, 13, 61, 91, 445
 relationships, 11-12, 92-93, 96-98,
 103-104, 163-165, 173-174, 297-
 298, 339-340, 367-369, 418-419,
 431-432, 449-450, 457-460
 responsibility, 32, 36, 38, 40, 55, 71,
 72, 74, 75, 119, 125, 128, 135, 137-
 160, 184-185, 282, 324, 333, 376,
 481
 reuse, 11, 40, 51, 56-57, 332, 336,
 345, 349, 353-356, 364, 365, 368,
 372, 374, 376, 380-393, 415, 453,
 488
 risks, 55-56, 69-70, 74-75, 304-305,
 376-377, 435-453, 476-478

S

sensemaking, 161, 176, 186-188,
 193, 196-198
 service, 239, 352, 407, 443, 461, 464
 service oriented architecture, 403,
 409, 414
 simulation, 36, 48-56, 68, 97, 164,
 214, 281, 283, 289-290, 325, 344-
 347, 352-356, 365-367, 370-377,
 385-393
 simulation based acquisition, 49, 51,
 372
 situation awareness, 208
 skill, 52, 55, 198, 288, 377

social
 networks, 162-165, 243
 sciences, 149, 150, 160-162, 166,
 192, 195, 198, 201, 202, 239
 software, 122, 143, 451, 466, 468,
 494, 499-500
 software engineering, 38, 43-44, 56,
 61, 66, 332, 341, 364, 373-375,
 378, 381, 383, 393, 412, 423, 427,
 429, 436, 445
 stages, 9-10, 44-45, 47-49, 73-75,
 371-372, 478-479, 480-481, 489-
 490
 standard, 40-43, 57-58, 248-251,
 376-378, 393-394, 403-408, 421-
 434, 438-440, 443-447, 464-474,
 480-482, 499-500
 standardization, 34-35, 57-58, 166-
 172, 175-178, 253-254, 302-303,
 399-400, 438-439
 state, 64-65, 90-92, 97-100, 157-158,
 240-242, 305-306, 351-352, 399-
 402, 412-413, 456-460
 structure, 12-13, 41-43, 48-49, 92-98,
 106-107, 161-173, 177-179, 241-
 243, 251-252, 431-432, 443-446,
 455-457, 468-469, 486-488, 504-
 508
 supply, 9, 17-18, 28, 40, 62-63, 72,
 105, 123, 139, 239, 240, 262, 296,
 299, 326, 338, 361, 420, 425, 428,
 461, 464, 474, 482, 484
 system modeling language, 403
 system of systems, 3, 36, 58, 62, 66,
 68, 89, 107, 111, 119, 149, 159,
 166, 192, 235-236, 312, 323, 326,
 358, 399, 400, 403, 433, 438
 systemic, 4-5, 58, 151, 180, 212, 290-
 291, 297-304, 309, 316-317, 364-
 368, 374, 381, 412, 435
 systems engineering, 36-37, 40-41,
 111-115, 150-151, 154-156, 160-
 166, 323-324

T

technical

documentation, 120

view, 209, 333, 414, 476

technological system, 90, 95-102,
105-111, 115, 201-202

training, 7-8, 171-172, 181-182, 289-
290, 477-478, 485-486, 498-499

transfer of ownership, 142-143

transportation, 24-25, 32, 63, 104-
107, 119, 131, 154-156, 159, 223,
226, 231, 246, 268, 402, 404, 407

U

unified modeling language, 378, 403,
413, 418

universal business language, 403, 406

UPDM, 403, 414-415, 418, 446

user, 47-48, 57-65, 68-69, 210-213,
239-241, 334-337, 355-356, 359-
360, 383-384, 429-430, 448-451,
460-461, 466-467, 471-472, 476-
479

V

validation, 35-37, 52-53, 342-356,
374-375, 390-393, 425-429, 434-
437, 486-487, 504-505

value, 3-5, 9-10, 16-20, 24-29, 64-65,
68-69, 71-72, 75-76, 100-105, 266-
267, 295-301, 308-313, 325-327,

value chains, 3, 17-18, 29, 297

verification, 112, 114, 127, 342, 346,
348-349, 352, 354-356, 368, 372-
375, 393, 414, 425, 427, 429, 460,
465, 477, 486, 492-493, 497

virtual organizations, 3