PRATT, J. G. M. (2007) 'Falling Behind': A Grounded Theory of Uncritical Decision Making, *PhD*, School of Management, Faculty of Business, University of Technology, Sydney, pp. 362-368.

Appendix Three: Coding and the Assistance of Qualitative Software

CODING OF TEXTS

The “heart and soul” of textual analysis is coding (Ryan and Bernard, 2003). Many authors who have written on coding have cited the work of Strauss (1987) and Strauss and Corbin (1998; 1990) for their advice on coding approaches (Miles and Huberman, 1994; Neuman, 2003; Ryan and Bernard, 2003). This is because grounded theory analysis begins with the open coding of a set of empirical indictors or actual data of actions and events (Strauss, 1987). This kind of analysis is necessary at the start of the analytical process to generate initial categories, properties and tensions, and help the researcher to think about the relationships between them (Strauss and Corbin, 1998).

Codes are indicators of a concept the analyst derives from them with increasing certainty, based on constant comparison of indicator to indicator (Strauss, 1987). The researcher then codes and names these as indicators of classes of events and actions. By comparing indicator with indicator, the researcher notes similarities and differences which result in coded categories. Indicators are then compared to the emergent concept and codes sharpened to achieve best fit with the data. Coding also involves discovering and amending categories and associated subcategories in the same lines of data, or around them in the broader text, or even in different texts (Strauss, 1987).

Some qualitative authors have suggested that codes are formed from the conceptual lens of the study, whether consciously identified or not (Miles and Huberman, 1994). Grounded theory researchers have argued against this imposing of “extant theories”, or one’s own beliefs on the data, arguing for a more emergent process that helps to keep developed theory true to the subject’s views of the world (Charmaz, 2000: 515) and allow the data to speak for itself (Strauss and Corbin, 1998: 59).

Strauss (1987) has argued that codes may be developed to indicate conditions, interaction among actors, strategies and tactics, as well as consequences (Strauss, 1987: 27-28). Conditions are sets of events or happenings that create the situations, issues and problems pertaining to a phenomenon. They explain why and how people respond in certain ways, and can arise out of time, place, culture, rules, regulations, beliefs, power, etc. They can be seen in phrases like ‘because’, ‘since’, ‘as’ or ‘on account of’. Conditions may have many different properties, including micro or macro influences, be stable or change over time, and combine with others factors. Strauss further argued that researchers should look for causal (influences on phenomena), intervening (mitigate or alter the impact of causal conditions, often out of contingent events) and contextual variables (patterns of conditions that intersect at a particular time and place to create a set of circumstances that led to the observed phenomenon).

Strategic or routine tactics refer to how people act — actions/interactions. They can be routine or more strategic. This coding class represents what people and units do and say.

Finally, consequences may be intended or unintended, and affect the conditions for the next round. Consequences may be seen in phrases such as ‘as a result’ and ‘because of’ (Strauss, 1987).

Glaser (1992) took issue with Strauss’ (1987) framework, and claimed that Strauss’ push for conditions, consequences and strategies failed his quality test for relevance, and results in forcing the data (Glaser, 1992: 53). Glaser proposed instead that codes should emerge through constant comparison.

Codes can range from being descriptive to interpretive, and can also include the setting, definition of a situation, perspectives, ways of thinking, processes, activities, events, strategies, relationships, and methods (Miles and Huberman, 1994). It is probable that there will at times be multiple codes for each chunk of data (Miles and Huberman, 1994). What is most important in coding however, is ensuring that there is a good structure behind the codes, with well supported definitions and terms that are close to the data they describe, where possible (Miles and Huberman, 1994).

To assist researchers in the open coding process, Strauss developed the following rules of thumb (Strauss, 1987: 30):

1. Look for in-vivo codes used by respondents (Glaser and Strauss, 1967);
2. Give a provisional name to each code, whether in-vivo or constructed by the analyst;
3. Ask a battery of questions about words and phrases in line by line analysis;
4. Move quickly to dimensions that seem relevant to given words and phrases; and

5. Remember that these dimensions should quickly call up comparative cases or lead to various conditions, actions/interactions and consequences.

Strauss (1987) also suggested writing frequent memos during the coding process, for writing down the different thoughts, categories, subcategories and possible explanations and further samples. Attempts should also be made to ‘dimensionalise’ the different items coded. Strauss (1987) and Strauss and Corbin (1998) also suggested that not every word be coded, but the document should be scanned for relevant instances, or the different instances or variations of those concepts (Pidgeon and Henwood, 2004: 637). These authors also suggested the substantive literature could be consulted to help examine the data from the core texts, however coding should not be drawn from substantive theory unless it fits the text on a line-by-line basis (Strauss, 1987: 283). Miles and Huberman (1994) finally suggested that when open coding in case studies, code the research data from one site before the next or going back.

Following open coding, grounded theorists then recommend researchers move onto axial coding, although this type of coding can be done at the same time as open coding (Strauss and Corbin, 1998; Charmaz, 2000). Axial coding refers to the act of relating categories to subcategories along the lines or axes of their properties and dimensions (Strauss, 1987: 64). Categories are a problem, issue, event or happening that is defined as significant to the respondents (Strauss and Corbin, 1998). A subcategory is a category that answers questions about the phenomenon of interest, such as when, where, why, who, how and with what consequences, giving greater theoretical power (Strauss and Corbin, 1998). Properties are the general or specific characteristics or attributes of a category, whereas dimensions represent the location of a property along a continuum or range (Strauss and Corbin, 1998: 117). Strauss and Corbin (1998) suggest trying to locate each property of a category along its dimensions, promoting a grouping according to these attributes. Linking of these takes place at a conceptual level, not a descriptive one (Strauss and Corbin, 1998).

> Categories turn description into conceptual analysis by specifying properties analytically (Charmaz, 2000: 517).

Axial coding progresses by:

1. Laying out the properties of a category and dimensions that begins during open coding;
2. Identifying a variety of conditions, actions/interactions and consequences associated with a phenomenon
3. Relating a category to its subcategories through statements denoting how they are related to each other; and

4. Looking for cues in the data that denote how major categories might relate to each other (Strauss and Corbin, 1998: 126).

Other grounded theorists have suggested making use of the six C's in coding at this level: causes, contexts, contingencies, consequences, covariation and conditions (Pidgeon and Henwood, 2004: 639; Kan and Parry, 2004). These line up with the independent variable, dependent variable, context, moderating variables, correlations and intervening variables in "equivalent nomethetic concepts" (Kan and Parry, 2004: 472). Despite the fact that this was criticised by Glaser (1992) for forcing, there is still great value in this framework, particularly if an emerging analysis framework is adopted, such as in this research.

Strauss and Corbin (1998) suggested three additional techniques that could be used to develop axial codes further at this stage of analysis:

1. The flip-flop technique, where concepts are turned upside down or inside out to gain a different perspective on the extremes or opposites to bring out important properties;
2. Systematic comparison to the researcher's own experiences and the literature, to help sensitise the researcher to properties in the data; and
3. Watching for uncritical bias by accepting all explanations of respondents, such as 'always' and 'never'.

After the development of the axial codes, it has been suggested that these codes be "tried out" on the next data set or case study for fit (Miles and Huberman, 1994). A range of questions could then be asked on this data, such as "What are the conditions under which it holds?" "When does it need to be qualified?" (Miles and Huberman, 1994). Strauss and Corbin (1998) proposed that their conditions-action/interaction-consequences matrix be revisited each time to investigate how these categories change in each case.

Following axial coding, the grounded theorist then moves onto selective coding. This stage involves the researcher deciding which category or categories are central to the research project, in pursuit of a core category. Subordinate categories must then be systematically related to the core category (Strauss, 1987: 69). Unrelated minor categores that do not relate to the core category are then encouraged to be discarded (Strauss, 1987).

PRATT, J. G. M. (2007) 'Falling Behind': A Grounded Theory of Uncritical Decision Making, *PhD*, School of Management, Faculty of Business, University of Technology, Sydney, pp. 362-368.

The first step in selective coding involves identifying the central category that represents the main theme of the research. It is a few words of what the whole research is about and is itself an abstraction. It must be able to handle lots of variation and can evolve out of a list of existing categories or be more abstract. Strauss (1987: 36) provided some criteria for choosing this core category:

1. Must be central and all other categories relate to it;
2. Must appear frequently in the data;
3. Explanation evolves is logical and consistent — no forcing;
4. Name or phrase sufficiently abstract to help other research in other areas;
5. Theory grows when refined with other concepts; and
6. Able to explain variation as well as main points in data. That is, when conditions vary, the explanation holds. The core category should be able to explain alternative cases in terms of the central idea.

Strauss and Corbin (1998) proposed that analysis should be concluded by mapping the relationship between the conditions, consequences and actions/interactions. They claimed that this will rarely follow a linear path (Strauss and Corbin, 1998: 183), and may involve a complex interplay between many possible conditions residing in the past, present or future anticipated, or even in contingencies considered (Strauss and Corbin, 1998: 184). Further, they also recommend considering the role of individual, group, sub-organisation, family, organisation, institutional, community, regional, national and global influences. Finally, the interplay between macro and micro conditions and how these affect not only each other, but their actions/interactions and consequences and next round effects should also be considered.

This thesis, in adopting a grounded theory analytical framework discussed in chapter two, makes use of the coding protocols suggested by the previous grounded theory authors. These coding procedures have not been conducted manually however, as there are now a range of different qualitative software tools that have been designed to assist with this kind of analysis.

CODING AND THE USE OF QUALITATIVE SOFTWARE

Before computer programs were developed to assist with qualitative data analysis, emergent data analysis approaches involved taking two copies of all work, and cutting up the second by analytic tags to get all the information out on particular topics (Lee and Fielding, 2004). A related challenge was, therefore, moving backward and forward between chronologically and topically organised data. Computer based approaches to storage, searching and operation have been argued to be far superior to manual approaches, and also assist with audit trails (Lee and Fielding, 2004: 534).

Qualitative data analysis software can help to search, mark up, link, reorganise data, represent or store one’s own reflections, ideas and theorising. Some of the most cogent reasons for using such software are for consistency, speed, representation (for example, drawings and diagrams) and consolidation (pulls all the different types of data together) (Weitzman, 2003: 316-317). Researchers need to be careful of the conceptual assumptions that can underpin some programs, such as having hierarchical relationships among concepts. Sometimes relationships will be non-hierarchical; if so, another program should be used or the research should work around this feature by keeping a code map pinned to the wall (Weitzman, 2003).

> Simply put, software can provide tools to help you analyze qualitative data, but it cannot do the analysis for you, not in the same sense in which a statistical package like SPSS or SAS can do, say, multiple regression (Weitzman, 2003: 314).

Most software packages now provide good support for code-based analysis and more conceptual work, with much of the analysis often tied to grounded theory approaches (Lee and Fielding, 2004: 539). Theory building programs have code and retrieve functions that can analyse relationships between codes and data, as well as help develop higher order classifications, formulate propositions which fit the data and test how they apply, as well as develop visual connections between the codes to help conceptualisation (Lee and Fielding, 2004: 532). The role of simple tables and matrices however, such as those used by Miles and Huberman (1994), should also be considered (Lee and Fielding, 2004: 532). A range of possible qualitative software programs was recommended, including Atlas, Hyperresearch, and NVivo (Lee and Fielding, 2004: 532). The next section of this chapter explains and justifies the use of NVivo software in this research.

NVIVO QUALITATIVE SOFTWARE

NVivo, formerly Nudist, is qualitative data analysis software designed explicitly to keep up with most qualitative approaches, including grounded approaches, through its memo, coding, analysis and charting functions (Bazeley and Richards, 2000; Charmaz, 2000; Kan and Parry, 2004; Soliman and Kan, 2004). NVivo has been argued to be effective particularly in consolidating data and theory, exploring the data consistently, and for its speed and representation to assist in theory building (Soliman and Kan, 2004; Weitzman, 2003). Its ease and flexibility of coding and memoing were also noted (Soliman and Kan, 2004: 4). In a grounded study of leadership, NVivo was successfully used to:

- Store and categorise interview transcripts, memos and other documents;
- Create categories through computer-assisted coding;
- Conduct searches relevant to analysis to generate reports;
- Move and link data as higher order themes emerged; and
- Create basic hierarchical models of codes (Kan and Parry, 2004: 473-474).

Despite NVivo's strengths as an appropriate software tool to assistrounded data analysis, there have been several criticisms noted of this package. First, Glazer (1998) has argued against the use of software in general, as he claims that it tends toward pattern analysis and description, rather than theory building (Soliman and Kan, 2004: 3). Others have argued that using NVivo can lead to lazy research and shortcuts (Soliman and Kan, 2004), and the risk of becoming locked into a "one-dimensional view of qualitative research" (Charmaz, 2000: 521).

Soliman and Kan (2004) have countered these criticisms, however, by asserting that the methodology and researcher must drive the software, and not the other way around. They argue that when researchers do not reflect upon the conceptual assumptions behind the software, then these criticisms can be justified (Soliman and Kan, 2004: 4). This is not the case when a well thought through methodological framework and researcher direct the software however (Bazeley and Richards, 2000).

A final criticism of the NVivo software that Soliman and Kan noted was that at higher levels of theoretical abstraction, presumably when moving towards core category codes, the software became slower, immobile, and visually constrictive. These authors suggested that more manual orientated approaches, such as using large paper displays, be used instead.

PRATT, J. G. M. (2007) 'Falling Behind': A Grounded Theory of Uncritical Decision Making, *PhD*, School of Management, Faculty of Business, University of Technology, Sydney, pp. 362-368.

REFERENCES

BAZELEY, P. & RICHARDS, L. (2000) *The NVivo Qualitative Project Book,* London, Sage.

CHARMAZ, K. (2000) Grounded Theory: Objectivist and Constructivist Methods, IN DENZIN, N. K. & LINCOLN, Y. S. (eds.) *Handbook of Qualitative Research,* Thousand Oaks, Sage Publications Inc, pp 509-535.

GLASER, B. G. (1992) *Basics of Grounded Theory Analysis: Emergence vs Forcing,* Mill Valey, CA, Sociology Press.

GLASER, B. G. & STRAUSS, A. L. (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Hawthorne, N.Y., Adline Publishing Company.

KAN, M. K. & PARRY, K. W. (2004) "Identifying paradox: A grounded theory of leadership in overcoming resistance to change", *The Leadership Quarterly,* 15, pp 467-491.

LEE, R. M. & FIELDING, N. G. (2004) Tools for Qualitative Data Analysis, IN HARDY, M. & BRYMAN, A. (eds.) *Handbook of Data Analysis,* London, Sage, pp 529-546.

MILES, M. B. & HUBERMAN, A. M. (1994) *Qualitative data analysis: an expanded sourcebook*, Thousand Oaks, CA, Sage Publications.

NEUMAN, W. L. (2003) *Social Research Methods: Qualitative and Quantitative Approaches,* Boston, Allyn and Bacon.

PIDGEON, N. & HENWOOD, K. (2004) Grounded Theory, IN HARDY, M. & BRYMAN, A. (eds.) *Handbook of Data Analysis*, London, Sage, pp 625-648.

RYAN, G. W. & BERNARD, H. R. (2003) Data Management and Analysis Methods, IN DENZIN, N. K. & LINCOLN, Y. S. (eds.) *Collecting and Interpreting Qualitative Materials,* Thousand Oaks, CA, Sage Publications, pp 259-309.

SOLIMAN, J. & KAN, M. K. (2004) Grounded Theory and NVivo: Wars and Wins*QualIT2004*, Brisbane, November 24-26.

STRAUSS, A. & CORBIN, J. (1998) *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory,* Thousand Oaks, Sage Publications.

STRAUSS, A. L. (1987) *Qualitative Analysis for Social Scientists,* Cambridge, Cambridge University Press.

STRAUSS, A. L. & CORBIN, L. (1990) *Basics of qualitative research: grounded theory procedures and techniques,* Thousand Oaks, California, Sage Publications, Inc.

WEITZMAN, E. A. (2003) Software and Qualitative Research, IN DENZIN, N. K. & LINCOLN, Y. S. (eds.) *Collecting and Interpreting Qualitative Materials,* Thousand Oaks, CA, Sage Publications, pp 310-339.