A critique of using grounded theory as a research method

George Allan
Department of Information Systems and Computer Applications,
Portsmouth University, UK
George.Allan@port.ac.uk

Abstract: Grounded Theory is a powerful research method for collecting and analysing research data. It was 'discovered' by Glaser & Strauss (1967) in the 1960s but is still not widely used or understood by researchers in some industries or PhD students in some science disciplines. This paper demonstrates the steps in the method and describes the difficulties encountered in applying Grounded Theory (GT). A fundamental part of the analysis method in GT is the derivation of *codes, concepts and categories*. Codes and coding are explained and illustrated in Section 3. Merging the codes to discover emerging *concepts* is a central part of the GT method and is shown in Section 4. Glaser and Strauss's constant comparison step is applied and illustrated so that the emerging *categories* can be seen coming from the *concepts* and leading to the emergent theory grounded in the data in Section 5.

However, the initial applications of the GT method did have difficulties. Problems encountered when using the method are described to inform the reader of the realities of the approach. The data used in the illustrative analysis comes from recent IS/IT Case Study research into configuration management (CM) and the use of commercially available computer products (COTS). Why and how the GT approach was appropriate is explained in Section 6. However, the focus is on reporting GT as a research method rather than the results of the Case Study.

Keywords: Grounded Theory; codes; concepts; emerging categories; emergent theory.

1. Introduction

Traditional research designs usually rely on a literature review leading to the formation of a hypothesis. This hypothesis is then put to the test by experimentation in the real world. On the other hand, GT investigates the actualities in the real world and analyses the data with no preconceived hypothesis (Glaser & Strauss, 1967). Data collection is usually but not exclusively by interviews. Analysis of interview data in qualitative research tends to result in descriptions of an interpretivist view of the events, whereas GT data analysis involves searching out the concepts behind the actualities by looking for *codes*, then *concepts* and finally *categories*. These are explained in the following section.

2. Grounded Theory Codes, Concepts and Categories

Grounded theory coding is a form of content analysis to find and conceptualise the underlying issues amongst the 'noise' of the data. During the analysis of an interview, the researcher will become aware that the interviewee is using words and phrases that highlight an issue of importance or interest to the research. This is noted and described in a short phrase. This issue may be mentioned again in the same or similar words and is again noted. This process is called *coding* and the short descriptor phrase is a *code*.

There were initial doubts about what a code was/is/should be. The literature tells us that coding should be performed with an open mind without preconceived ideas. Glaser & Strauss (1967) insisted that preconceived ideas should not be forced on the data by looking for evidence to support established ideas. Glaser (2001) recommended that if a researcher were uncertain about the process, just analyse the data in front of you and write what you see.

Strauss & Corbin (1998, pp. 65-68) recommended coding by "microanalysis which consists of analysing data word-by-word" and "coding the meaning found in words or groups of words". An example is given in the following section. The data in this case comes from an interview with the IT Manager of a medium sized UK company specialising in customer billing.

2.1 Micro-Analysis Coding

The interview text is shown in the left-hand column and the right-hand column shows the codes that the researcher used in this case.

More than one code may emerge from the same text. The data were revisited many times looking and re-looking for emerging codes. Other issues will emerge, resulting in other *codes* from this and subsequent interviews.

Table 1 : Examples of early codes in grounded theory analysis of Case Study Y data

Interview Text	Codes
From my perspective	Personal view
the main challenge is	Assertion
in changes in technology	Changes in technology
or the product improvement	Changes in product
done by the COTS supplier.	Assertion Changes by Supplier
You	Pronoun shift
can never guarantee that	Assertion Uncertainty
if you are buying several,	Procurement
they will all be the same.	Product consistency Necessary condition
Yes,	Affirmation
when you come to buying PCs	Procurement of hardware
a lot of our products now are delivered with the software already loaded on the PCs	Integrated products Hardware Software
that causes you to go through an inspection.	Extra work Costs in human effort Costs in time
We weren't happy,	Dissatisfaction
it was costing us extra money.	Costs in money
Last year this part of Company Y organised a forum workshop seminar on COTS,	Extra work Action due to COTS shortfall
and as part of that we did a survey of a number of our projects on problems and issues with using COTS	Extra work Implementation difficulty
the short time that components become obsolete.	Short time to obsolescence

2.2 Difficulties with Micro-Analysis Coding

This analysis technique of coding by microanalysis of the data, word-by-word and line-by-line, had two drawbacks. Firstly, it was very time consuming. The transcription of each interview contained a mass of data that had to be studied to locate the information relevant to the research topic. Secondly, it led to confusion at times. Dividing the data into individual words caused the analysis sometimes to become lost within the minutia of data. So many words being picked over individually led to confusion. There were times when the focus was lost. Doubts were experienced about what it was that we were looking for.

Further reference to the grounded theory literature uncovered the rift between Glaser and Straus on this issue. Glaser (1992, p. 40) condemned this micro-approach as producing an "over-conceptualisation". This encapsulated exactly what was being experienced and the analysis, from this point on, followed Glaser (1992). That is, identifying key points (rather than individual words) and allowing concepts to emerge. The selection of points key to addressing research questions is in line with qualitative coding analysis (Miles & Huberman, 1984) as a protection against data overload. Dey (1993, pp. 94-97) talks of "bits of data" that are considered important. Therefore, key points in each interview were identified and marked ready for analysis and coding.

2.3 Key Point Coding

The points regarded as important to the investigation were identified in the transcripts, highlighted in italic font and given an identifier attributed sequentially starting at the first interview and continuing on through subsequent interviews to give P1, P2, and so on where 'P' indicates 'key point'. To differentiate key points made longitudinally in subsequent case studies, these identifiers were distinguished with a suffix X, Y or Z. For example, key point 8 made by the staff in case study X appears as P_X8. Thus it is possible to trace back through the interview transcriptions to the actual content and context of each key point. The key point identifiers are shown in the left-hand column of Table 2. The text of the key point is shown in the middle column and the code in the right-hand column.

Table 2: Examples of Key points and codes from the data in Case Study X

Id	Key Point	Code
P_X1	Manual standards on CM were set up as a result of a Requirements Study for better control of in-house software development.	control of software software development
P_X2	Company X required a CM system at minimum cost.	requirement for a CM system
P_X2a	Maestro II was selected as CM tool	CM tool
P_X3	End-to-end CM is defined as the full life-cycle from conception stage through to and including operational maintenance.	CM system
P_X4	Scheduling changes was regarded as an essential and integral part of the software process.	changes software process
P_X5	CM Projects Department has been officially formed.	recognition of CM
P_X5a	All future work will mandate CM on all data streams.	recognition of CM support for CM
P_X6	Processes equate to Configuration identification and configuration control.	CM process
P_X7	CM audits are used to bring other software systems under CM control.	CM process
P_X8	Status Accounting is used to report monthly to the Project Board.	CM process
P_X9	Main difficulty is getting people to buy-in to CM.	people difficulty
P_X10	3rd parties have a preconceived set of established tools and are not willing to see the in-house point of view.	people difficulty tool difficulty
P_X11	Developers saw CM as a control mechanism rather than a helpful tool.	not helpful control people difficulty
P_X12	People in the rapid application development (RAD) team thought that CM slows down their work and perceived CM as "just another layer of administration".	slows down work administration resistance to CM
P_X13	Involving software developers in the CM set-up processes can solve problems with the perception of CM.	people issue
P_X13a	CM perception problems can be solved by involving people in discussions	people issue
P_X22	Software is controlled in pre-production and production using CM.	control of software
P_X27	Company X had a need to develop a version control system for software. [Comment: This implies that there is not one available on the open market]	tool difficulty software version control

The key points P_X2a, P_X5a and P_X13a arose on a second, third or subsequent pass of the data. Many key points had been labelled with their unique identifiers and this avoided re-sequencing on every pass. Revisiting the data and creating multiple passes was to become a common occurrence.

The codes are then analysed and those that relate to a common theme are grouped together. This higher order commonality, is called a *concept*. Other concepts emerge from other groupings of the codes. An illustration is given in Section 4. Concepts are then grouped and regrouped to find yet higher order commonalities called *categories* illustrated in Section 5. It is these concepts and categories that lead to the emergence of a *theory*. If the data has been analysed without a preconceived theory or hypothesis, that theory is truly *grounded* in the data because it came from nowhere else.

3. The Emergence of Concepts

The code "software development" emerged from P_X1. The codes from all other key points were compared with this to see if similar codes occurred often. The following codes were considered to have commonality: "software development" from P_X1; "software process" from P_X4; "control of software" from P_X22; software version control from P_X27. The common characteristic is "Used for control of software development" and this was the first concept to emerge from the data. This was noted in Table 3.

From P_X12 emerged "slows down work", layer of "administration" and "resistance to CM". Combining these with P_X13 and P_X13a led to the concept of "Perceived as extra work". Other combinations of codes led to further concepts and these were added to Table 3. Confidence in the process of coding grew and uncertain subsided with experience of the method.

The process of comparing the codes with each other, to find higher order commonality, produced the concepts from the codes. The concepts from Case Study X are summarised below.

Table 3 : Emergence of concepts from the codes in Case Study X data

Used for control of software development – P_X1, P_X4, P_X11, P_X22, P_X27
Perceived as extra work – P_X12, P_X13, P_X13a
Recognised need for a CM system – P_X2, P_X3
CM recognised by company – P_X5, P_X5a, P_X17, P_X19, P_X23, P_X24, P_X29
Use of established CM processes – P_X6, P_X7, P_X8, P_X30
Difficult to get people to buy-in – P_X9, P_X10
Tools do not have the correct functionality – P_X10
CM active seen as part of other activities – P_X15, P_X16, P_X26, P_X30, P_X31, P_X33, P_X34
Used for control at system level – P_X4, P_X11, P_X18, P_X20, P_X23, P_X24, P_X27, P_X32, P_X35
People support is needed – P_X9, P_X25, P_X29
People are reluctant to practise CM – P_X11, P_X28
CM tool difficulties – P_X10, P_X27
Proof of control – P_X33, P_X34, P_X34a

Gaser & Strauss (1967, p. 37) insisted that there is little point in looking for ideas established by other researchers, as this "hinders searching for new concepts". Therefore, each key point was analysed for new concepts relevant to the current research. It is permissible to note support for a concept that is emerging from the current research. This provides important substantiation of that concept.

4. Categories

By comparing each concept in turn with all other concepts, further commonalities are found which form the even broader categories. Glaser & Strauss (1967, pp. 105-115) described this method of continually comparing concepts with each other as their "constant comparative method".

4.1 The Emergence of Categories from Case Study X

By applying the constant comparison technique to each concept in turn, a common theme were found amongst the following concepts:

Used for control of software development;
Used for control at system level;
Proof of control; Need to control the introduction of software

These share the theme of CM AS A CONTROL MECHANISM. This was the first category to emerge from the data and is demonstrated diagrammatically in Figure 1.

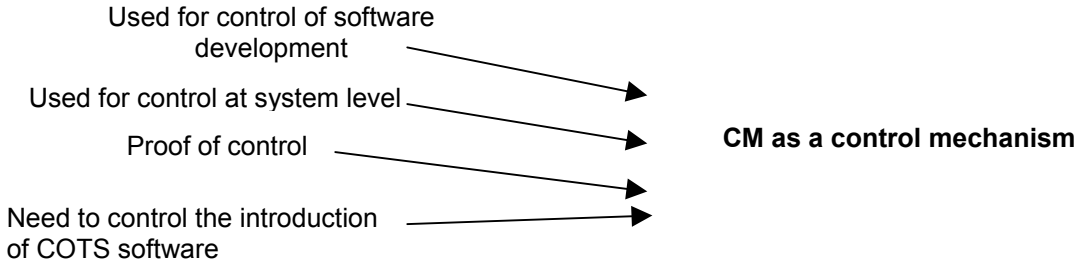

Figure 1 : Diagrammatical emergence of the category "CM as a control mechanism"

By comparing the other concepts and grouping *People are reluctant to practise CM, Perceived as extra work*, *Getting people to buy-in* and *Tools do not have the correct functionality* a category emerged in Figure 2 as "Difficulties with CM".

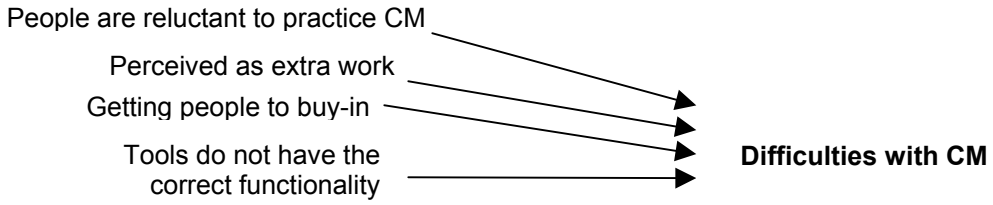

Figure 2 : Diagrammatical emergence of the category "Difficulties with CM"

Grouping *People support is needed*, *Use of established processes* and *There is a need for CM* in Figure 3 gave the category "CM practices".

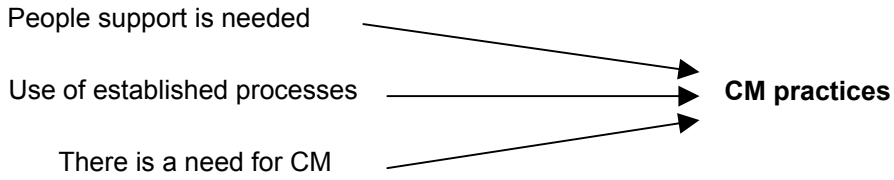

Figure 3 : Diagrammatical emergence of the category "CM practices"

Grouping *Tool selection* with *CM tool difficulties* in Figure 4 gave "Tools issues".

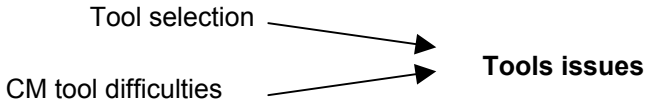

Figure 4 : Diagrammatical emergence of the category "Tools issues"
These categories and their relevant concepts are displayed in Table 4.

Table 4: Emergence of categories from the concepts in the data from Case Study X

CM AS A CONTROL MECHANISM
Used for control of software development - P_X1, P_X4, P_X11, P_X22, P_X27
Used for control at system level – P_X4, P_X11, P_X18, P_X20, P_X23, P_X24, P_X27, P_X32, P_X35
Proof of control – P_X33, P_X34, P_X34a
Need to control the introduction of COTS software – P_X35, P_X36, P_X37
DIFFICULTIES WITH CM
Getting people to buy-in – P_X9, P_X10, P_X11, P_X28
Perceived as extra work – P_X12, P_X13, P_X13a
Tools do not have the correct functionality – P_X10
CM PRACTICES
Use of established CM processes – P_X6, P_X7, P_X8, P_X30
There is a need for CM – P_X2, P_X3, P_X5, P_X5a, P_X17, P_X19, P_X23, P_X24, P_X29
People support is being accomplished but still a problem – P_X9, P_X11, P_X25, P_X28, P_X29
TOOLS ISSUES
CM tool selection – P_X2a
CM tool difficulties – P_X10, P_X27

Data from two other case studies were analysed to further establish or otherwise these categories and discover new categories.

4.2 Emergence of categories from Case Study Y

The second case study was a manufacturer and supplier of equipment computer. The company structure was divisional. The analysis proceeded by comparing the new key points with the concepts and categories thus far established. New concepts will appear. This evidence added further substance in what Glaser & Straus (1967) called substantiation leading to substantive theory.

Table 5: Emergence of concepts and categories from the data from Case Study Y

CM AS A CONTROL MECHANISM
CM is used for control of software development - P_Y2, P_Y8
CM is used for control at system level - P_Y3, P_Y4, P_Y12a
Proof of control
DIFFICULTIES WITH CM
Getting people to buy-in - P_Y5, P_Y6, P_Y11
Perceived as extra work - P_Y7, P_Y8
Tools do not have the correct functionality - P_Y9, P_Y13
CM PRACTICES
Use of established CM processes - P_Y4, P_Y8
There is a need for CM - P_Y1, P_Y2, P_Y3, P_Y4, P_Y12a
People support
Relationship between CM and project management - P_Y3, P_Y4, P_Y10, P_Y14, P_Y5a
Cradle-to-grave concept of CM - P_Y12a, P_Y3

TOOLS ISSUES
CM tool selection - P_Y12
CM tool difficulties - P_Y9, P_Y13

The two concepts 'Proof of control' and 'People support' had no support from this data analysis. However, two new concepts emerged as:

'Use of CM in project management' - P_Y3, P_Y4, P_Y10, P_Y14

'A cradle-to-grave concept of CM' - P_Y12a, P_Y3
These new categories were integrated into the list of emerging categories.

4.3 Emergence of categories from Case Study Z

The third case study was a medium sized company employing approximately 500 staff providing a service function of customer billing to a national conglomerate organisation. The organisational chart of this company showed functional areas which were strictly defined with a rigorous reporting hierarchy. The organisation was bureaucratic with well-defined management levels.

The analysis of the key points into their codes continued as before by searching for key points in the data and identifying codes. A few key points are illustrated in Table 6 as examples.

Table 6: Example of the key points and codes from Case Study Z

Id	Key Point	Open Codes
P_Z1	Project people are the customers of CM.	CM in project management
P_Z2	CM is being used actively.	need for CM
P_Z3	There is difficulty in capturing the configuration of some systems that SD&E are not directly responsible for.	people difficulties
P_Z4	The biggest difficulty has been to identify how much control CM should have over systems.	people difficulties
P_Z5	Another difficulty is how to achieve CM control consistently.	people difficulties
P_Z6	There is resistance to CM from the owners of systems.	resistance people difficulties
P_Z7	Documentation becomes part of the CM system.	CM process
P_Z23	There should be more of this research going on.	more research
P_Z24	Outsiders see CM as really boring.	people difficulty

A new concept emerged from P_Z23, that of "More research needed in CM".

This new category was integrated into the list of emerging categories and the wording of existing concepts adjusted where necessary to reflect the additions to the data accumulated. For example "Getting people to buy-in" became "People difficulties" and "difficulties with tools" was reworded as "Technical difficulties" to reflect the totality of data collected without loss of relevance or focus.

As subsequent interviews took place, in any case study, the process of constant comparison continued. Key points identified in the transcripts were compared with concepts and categories so far established and adjustments made to categories to reflect accumulated findings. These, in turn, were then used in subsequent analysis. Emerging categories are show diagrammatically in Figure 5.

5. Emerging Theory

By linking the categories and investigating the connections between concepts the theory emerges. From Figure 5 the categories were: **CM as a control mechanism, Difficulties with CM, CM practices** and **Tools issues.** There is not space in this illustrative paper to analyse fully the connections between categories and the concepts. This can be obtained on request. However, the emergent grounded theory of computer CM can be summarised as follows.

"Configuration management is a set of processes that collectively form a control mechanism to assist the effective operations of computer systems. The usefulness of CM will be impaired if the benefits from it are not recognised and supported among technical and business personnel. A sophisticated CM system will assist in identifying where and how components interact and the relationships between them so that a new component may be implemented in the correct place and continue the harmony of the whole system. However, Currently available CM tools can be difficult to use. It is therefore important to select appropriate tools with care.

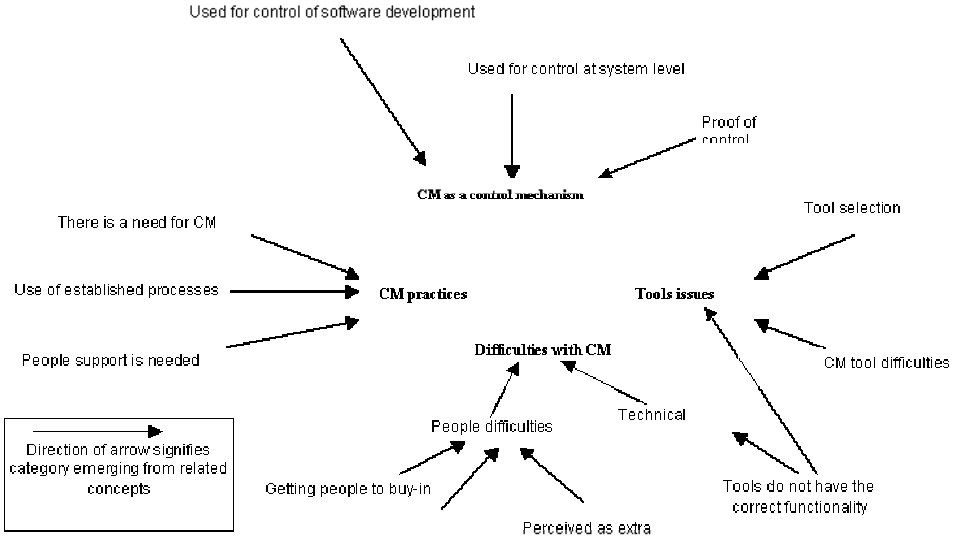

Figure 5. : Emergent categories derived from grounded theory analysis of interview data in 3 case studies 1999-2002

6. Discussion

The reader will see that the four categories and some of the concepts are embedded in this summary. This is how GT leads from codes to concepts to categories to theory. The resultant theory does not need separate justification and testing because it came from live data. More data will be collected in future case studies and the theory will be improved upon and amended to reflect real life.

6.1 Grounded Theory and Literature Search

Some people have interpreted the GT method to mean fieldwork before literature search but this is a misconception of the original premise put forward by Glaser & Strauss (1967, p. 169) who encouraged researchers to "use any material bearing in the area". This is taken to include the writings of other authors. Strauss & Corbin (1998) saw the use of literature as a basis of professional knowledge and referred to it as literature sensitivity and Dey (1993, p. 66) saw it as "accumulated knowledge".

The research used to illustrate the GT method investigated current practices in large UK organisations when maintaining computer systems supporting their core businesses in the commercial world. A review of the relevant literature established current thinking in the areas of configuration management and the use of commercially available components. However, this literature review did not lead to any hypotheses of sufficient interest. Combine this with the fact that GT investigates actualities in the real world (which CM practices are) and the grounded theory approach was appropriate for this research. The choice of case study as the research method to collect the data is justified in the next section.

6.2 Grounded Theory and Case Studies

The use of the case study as a research paradigm is well known. Yin (1994) advocated case study research to investigate contemporary phenomena within real-life contexts especially when the boundaries between phenomena and contexts are not clear. In this research into CM practices, the

boundaries are not clearly defined between phenomena (the practices) and contexts (the companies) or within phenomena, or within contexts.

However, there are certain tensions between Yin's version of the case study paradigm and the GT methodology. Yin (1994, p. 13) suggested that the case study "benefits from the prior development of theoretical propositions to guide data collection and analysis" whereas Glaser & Strauss (1967) insisted that a grounded theory approach should have no pre-conceived ideas or hypothesis.

A criticism of the case study as a research method is that there can be no generalisation of findings but Yin (1994) defended the position that case studies do lead to building theories applicable in the world at large. Grounded theory specifically attempts to investigate the real world, usually through interview data. It discovers the concepts grounded in the data and uses those concepts to build theory. The use of both these methods, therefore, minimises this criticism.

There is, therefore, no real anomaly between the use of case study and the use of GT. Both research methods collect data using interviews.

6.3 Grounded Theory and Interviews

Greater reliability can be placed on the data gathered in an interview over that gathered by a list of self-completion questions in a survey. In a face-to-face situation an experienced interviewer can tell whether the respondent is the appropriate person to answer the questions. Respondents are able to discuss issues in detail (Hague, 1987) and it is possible to use diagrams to clarify points.

Another possible criticism of GT is a lack of rigour due to careless interview techniques and the introduction of bias. In this research the case study interviews were guided to avoid leading questions and the introduction of bias. A working awareness of bias is imperative in all interview research. Transcriptions were checked for context and content accuracy before analysis began. These precautions and the fact that conclusions drawn are grounded in actual data helps minimise the risk of bias.

A real advantage of GT is that analysis starts as soon as data collection begins in the first interview. Glaser (2002) holds that analysis can start during the first interview if the researcher identifies concepts that are striking at that time. However, it is not sufficient simply to inspect data and label interesting points, the data have to be analysed in a systematic and rigorous manner to discover the concepts leading to the categories. This is an iterative process that requires a great deal of time, patience and analytic skill.

7. Conclusions

These conclusions are written as a personal reflection of my experiences of GT.

Glaser & Strauss (1967) emphasis that the researcher should have "no preconceived ideas" when collecting and analysing data. I had two problems with this. Firstly, there has to be some agenda for research by interview. Busy people in industry and commerce expect meetings to have an agenda and research projects to be scoped. Time and resource constraints prohibit unfocused investigation. It is clearly not possible to investigate specific working practices in UK industry without some focus to work toward but this is not what Glaser & Strauss meant. They were referring to preconceived bias, dogma and mental baggage which, in this case, may be taken to be preconceived ideas about working practices embedded in the researcher's mind (Glaser 2002). Interviews were therefore focused using open stimuli such as "Tell me about the working practices to do with CM" or "What happens to new components?" Extra time then had to be budgeted for the analysis as the interview transcripts were voluminous.

Secondly, I was looking for a clearly defined coding process or mechanism. Grounded Theory demands more in analysis than simple inspection of the data. However, Glaser & Strauss (1967) and later Glaser (1978; 1992) do not instruct the reader in a prescribed mechanism for performing the coding. They describe the conceptualisation of coding. I was not sure what I was looking for. What was "a code"? Was it a statement of importance, in which case, what was important *if we were to have no preconceived ideas*? In previous research I had always had a hypothesis on which to focus. Was a code a statement of interest? At the outset I was unsure what was of interest.

Both these difficulties were overcome by identifying the Key Points in the interview data and concentrating the analysis on these. However, another difficulty experienced was in knowing when coding should be ended. Did every relevant statement in the text have to be

identified and used? Was one statement enough if made by a person in authority? Was it enough to have three of four statements containing the same code. How many times did a code have to occur to be substantive. When performing the constant comparisons between concepts to find emerging categories, how many concepts need to be included to form a category. Glaser (2002) advises that "one is enough if it is significant". One concept can contribute to the emerging theory.

I was also uncertain about when to finishing the analysis. Glaser (1978) discusses *saturation* as the key to knowing when to stop. However, it took many attempts before I was at all confident when to stop the analysis and form the theory. Latterly I found that the theory could be allowed to emerge right from the start. I use the term "allowed to emerge" to mean that concepts and categories should be noted and merged as soon as they are noticed and this is the start of the theory. The researchers' mind-sets are used to waiting until all data are collected before starting analysis and all analysis is completed before drawing conclusions. In GT this is not the case and this needs to be understood and appreciated. Analysis can start with the first interview.

In conclusion, the Grounded Theory method is recommend as a powerful way to collect and analyse data and draw meaningful conclusions. This recommendation applies to any researcher in the hard sciences as well as the social sciences.

References

Dey, I. (1993) *Qualitative Data Analysis*, London, Routledge

Glaser, B.G. & Strauss, A.L. (1967) *The Discovery of Grounded Theory*, New York, Aldine

Glaser, B.G. (1978) *Theoretical Sensitivity*, Mill Valley California, Sociology Press

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

Glaser, B.G. (2002) Private conversations in Paris

Hague, P. (1987) *The Industrial Market Research Handbook*, London, Kogan Press

Miles, M.B. & Huberman, A.M. (1984) *Qualitative Data Analysis - A Sourcebook of New Methods*, California, Sage

Strauss, A.L. & Corbin, J. (1998) *Basics of Qualitative research techniques and Procedures for Developing Grounded Theory*, USA, Sage

Yin, R. (1994) *Case study research: Design and methods* (2nd ed.), Beverly Hills, CA, Sage Publishing.

