December 2002

Four Steps to Analyse Data from a Case Study Method

John Atkinson
Charles Sturt University

Four Steps to Analyse Data from a Case Study Method

John Atkinson

School of Environmental and Information Sciences
Charles Sturt University
Albury, Australia
jatkinson@csu.edu.au

Abstract

Four steps are proposed to assist the novice researcher analyse their data that has been collected using a case study method. The first step proposes the creation of a data repository using basic relational database theory. The second step involves creating codes to identify the respective 'chunks' of data. These resulting codes are then analysed and rationalised. The third step involves analysing the case study data by generating a variety of reports. The fourth step generates the final propositions by linking the rationalised codes back to the initial propositions and where appropriate new propositions are generated. The outcome of these steps is a series of propositions that reflect the nature of the data associated with the case studies data.

Keywords

Case study, information requirements analysis, methodological

BACKGROUND

A case study is one of the many qualitative and quantitative methods that can be adopted to collect data for research. Such methods represent part of what is referred to as the research strategy that details the design and data collection approaches to be used in the research (Fowler and Mangione, 1990). Yin (1994), who is one of the most cited researchers with regards to case study research (Markus, 1989), states that the case study method "is an empirical enquiry that investigates a contemporary phenomenon within its real life context". Typically interview techniques are utilised as part of the case study method to address the 'how' and 'why' type research questions. The strategies to collect data using such techniques are well defined however one of the main issues associated with any research is how to interpret the resulting data (Coffey and Atkinson, 1996). In particular there are few specific practical examples available to guide the novice researcher in the analysis of case study data. For example, Yin (1994) outlines a research design strategy for undertaking case studies that involves five components or steps, however there is limited detail provided with respect to the process of analysis of such data. Similarly Miles and Huberman (1994) propose an approach to the analysis of case study data by logically linking the data to a series of propositions and then interpreting the subsequent information. Like the Yin (1994) strategy, the Miles and Huberman (1994) process of analysis of case study data, although quite detailed, may still be insufficient to guide the novice researcher. Therefore, this paper will provide a series of steps to assist novice researchers carry out the successful analysis of case study data. This will be achieved by explaining in terms of a practical example based on seventeen case studies conducted using semi-structured interviews as the data collection technique. It can be assumed that these case studies have already been conducted and the data recorded, perhaps onto a tape recorder, and subsequently transcribed onto a word processor. The remainder of this paper will detail the steps that can be followed to assist in the analysis of such case study data (see Figure 1). This paper has resulted from part of a PhD. (Atkinson, 2002) and should be considered as research in progress.

CASE STUDIES

In Figure 1 the focal point of the case studies is a series of propositions (Miles and Huberman, 1994). Such propositions can be derived from the research questions or from interpreting data from other sources including from the literature and/ or surveys. The propositions are used in two ways to guide the development of the case studies. First, they

greatly assisted in the formulation of the case study questions and second, they serve as the basis for creation of the initial codes for use during the analysis of the case study data.

Once the data collection process for the case studies is commenced, Yin (1994) suggests that the early analysis of the data is a critical step in the overall interpretation of the case studies. Miles and Huberman (1994) outline a number of methods that could be adopted in the early analysis of case studies however no prescriptive practical recommendations are made as to which one to use. To assist in the early analysis of the case studies, in this paper, a decision was made to use the 'Codes and Coding' technique. This technique was selected as it lent itself to being able to link the data back to the research questions and the propositions (Miles and Huberman, 1994). This ability to be able to link the data to these respective components made the task of interpreting the output from the case studies more intuitive.

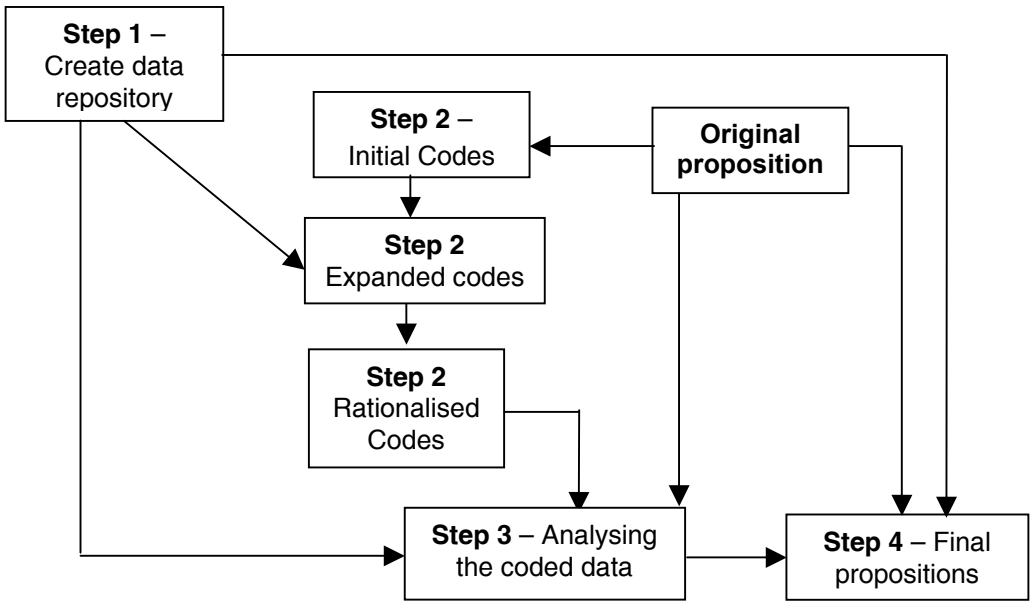


Figure 1: Case study structure

According to Miles and Huberman (1994) the codes and coding technique utilises the case-oriented approach strategy referred to as 'partial ordered displays' to analyse the case study data. This strategy allows for the quick identification of the segments relating to the research questions and any potential themes (Miles and Huberman, 1994:57). The process involves creating codes to be used for the analysis of the case study data and then coding the data. Codes are tags or labels that assign units of meaning to the data and for the quick identification of the segments relating to the research questions and any potential themes (Miles and Huberman, 1994:56). The identification of these segments is facilitated by the creation of meta-matrices to assemble descriptive data from the different cases into a standard format. In essence the process groups all the condensed data together allowing for comparisons to be made between them. Once these segments are identified the analysis of the case study data can be considered more straightforward (Miles and Huberman, 1994).

Depending upon the number of cases it could be possible to include all the data into one very large meta-matrix. Even though there are advantages of containing the data in this way, practically it can be difficult to conceptualise and manipulate. For this reason virtual matrices are a logical alternative and can be created using a computer application (Dey, 1993). However as Weitzman and Miles (1995) indicate, computers do not analyse your data, people do. Therefore a computer application should only be considered in terms of its ability to assist the researcher to understand the data collected (Coffey and Atkinson, 1996). However there are a very large number of commercial packages developed specifically for the analysis of qualitative data (Weitzman, 2000), which may cause some confusion when attempting to select the most appropriate package to use. One commonly cited package is Nudist, however this package is quite complex and can take time to learn all the associated operations (Barry, 1998). On the other hand Microsoft Access offers excellent versatility in

the manipulation of the data and is relatively intuitive to learn. A software package based on Microsoft Access is AnSWR that is suitable for the analysis of qualitative data (AnSWER, 2000). The main benefit for this package explicitly states that it is for use in 'team-based qualitative data analysis' (AnSWR, 2000). The research conducted in this paper was an individual-based project and was not intended to be applied in a team-based project. In addition AnSWR includes additional features such as quantitative data components that can intimidate the novice user. However the benefits of the underlying Microsoft Access application include the fact the software is easy to learn allowing for a relational database to be quickly generated. Microsoft Access also allows for easy user manipulation of the data, and has the option to produce a variety of user-generated reports. These factors empower the analyst to conduct more sophisticated forms of analysis on the case study data because the person driving the analysis process is more in tune with the physical aspects of the data being studying. That is, this software allows the analyst to experiment with different forms of data output instead of just plugging the data into some commercial package and then blindly accepting the result. However to implement a virtual matrix design requires some knowledge of basic relational database theory. This knowledge is critical for the successful analysis of the data and therefore its importance should not be overlooked when using this approach. This includes being able to determine the attributes to be included in the respective tables and predicting the types of output that could/ would be expected.

The remainder of this paper will outline a series of steps that can be used to analyse the data collected from a case study method. These steps do not imply that this approach is the only way case study data can be analysed (Barry, 1998) and it is recommended that they be used in conjunction with the overall case study design frameworks proposed by Yin (1994); and Miles and Huberman (1994).

Create data repository

To be able to analyse the data from the case studies it has to be in a format that allows for easy manipulation. A word processor only allows for crude manipulation of such data and therefore, as proposed above, the use of a database is considered more appropriate. However, to effectively use a database to manipulate the case study data a schema must be developed for this data. To do this the person developing the schema must have a rudimentary understanding of relational database theory. This schema can be implemented using a software application such as Microsoft Access, a relational database generator. The result of this process is a database having a relational format. An example of a typical database schema is shown in Figure 2:

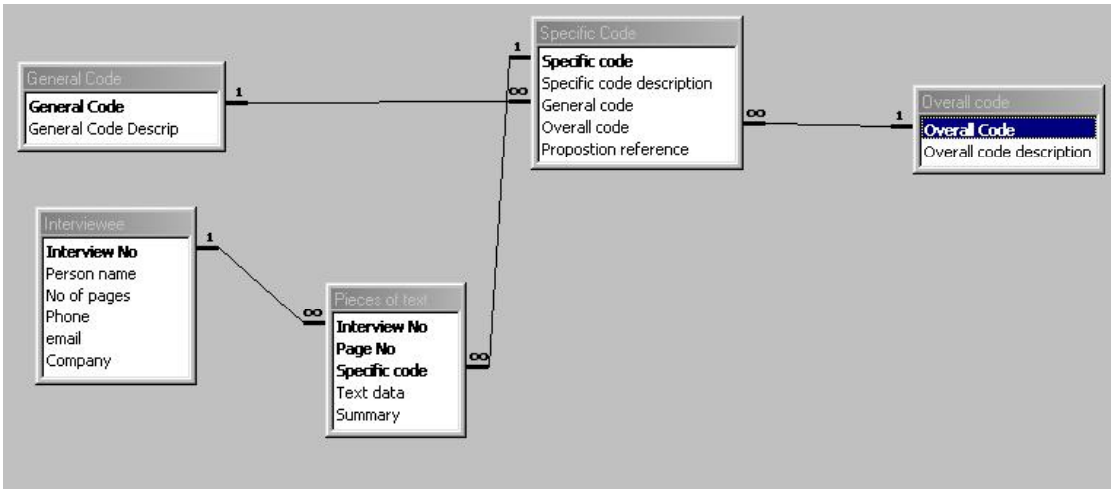


Figure 2: Database schema

The database structure must allow for maximum manipulation of the coded data and for this example, five tables are illustrated. Regardless of the number of tables proposed, it is important that they are generated according to the principles of relational database theory (Codd, 1990). In addition it is important that the resulting tables can be manipulated in such

a way to allow for maximum views of the data. An example of the structure of one of the tables in Figure 2 is shown in Table 1a.

Table 1a defined the actual codes that were utilised to code the data collected through the case studies.

Field	Attribute	Explanation
Specific code	Primary Key	The code to identify the specific chunks (segments) of text within the case studies. It is a 6-letter code.
Specific code description		Explanation of the specific code.
General code	Foreign key	The specific codes grouped under a general code to stratify them.
Overall code	Foreign key	The overriding code that grouped all the specific codes into six overall codes.
Proposition reference		This links the specific code to a specific proposition.

Table 1a: Specific code

Forms created

Once the schema of the proposed database has been defined and implemented using Microsoft Access, the next step is to populate the respective tables with the data. However before this occurs, and to ensure it occurs in the most effective manner, it is recommended that a front-end application be used. Front-end applications allow the data to be entered via forms rather than entering the data straight into the database tables. Forms are a database feature that can be used to ensure the integrity of the data in the database is guaranteed. This is achieved by incorporating specific controls into the forms specifying what data can and cannot be entered into the database. Most importantly this process ensures there is consistency in the quality of the data that is entered into the database. An example of the form used to add data to the database is shown below. The white areas are the areas that allowed for the user to enter data, while the grey areas are the ones to preserve data integrity.

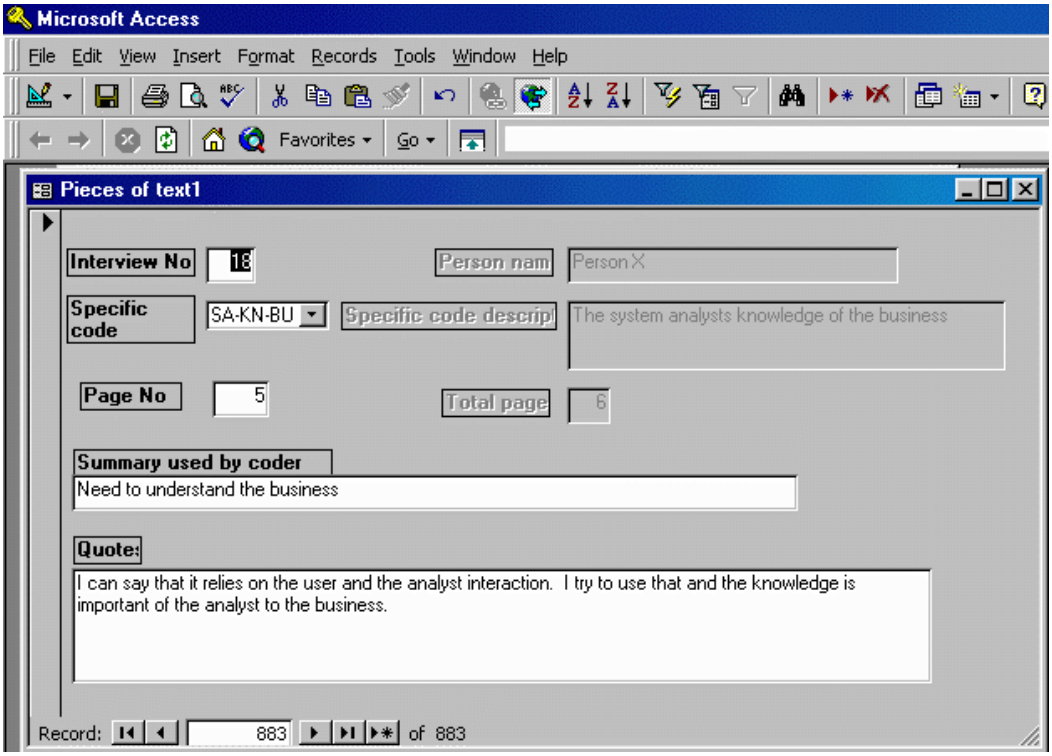


Figure 3: Data entry form

Codes and coding

Once the forms have been created the next step is for data entry to begin. This process requires the data to be organised and managed in some form by assigning 'tags' or 'labels' to the data collected. This process is often referred to as coding (Coffey and Atkinson, 1996). For example a code might identify a word, a phrase, a sentence, or a paragraph important to the research. The code attempts to associate meaning to these chunks of data. (Miles and Huberman, 1994:56) and this requires an appreciation of the data you are working with (Dey, 1993). Therefore, the person undertaking the data collection process is ideally suited to be the person who will also perform the process of coding this data.

It needs to be appreciated that the coding process is an evolutionary process. Figure 4 illustrates that up to three sets of codes can be employed during the process: initial codes, expanded codes and rationalised codes. The process to develop codes for the analysis of the case study data is outlined below.

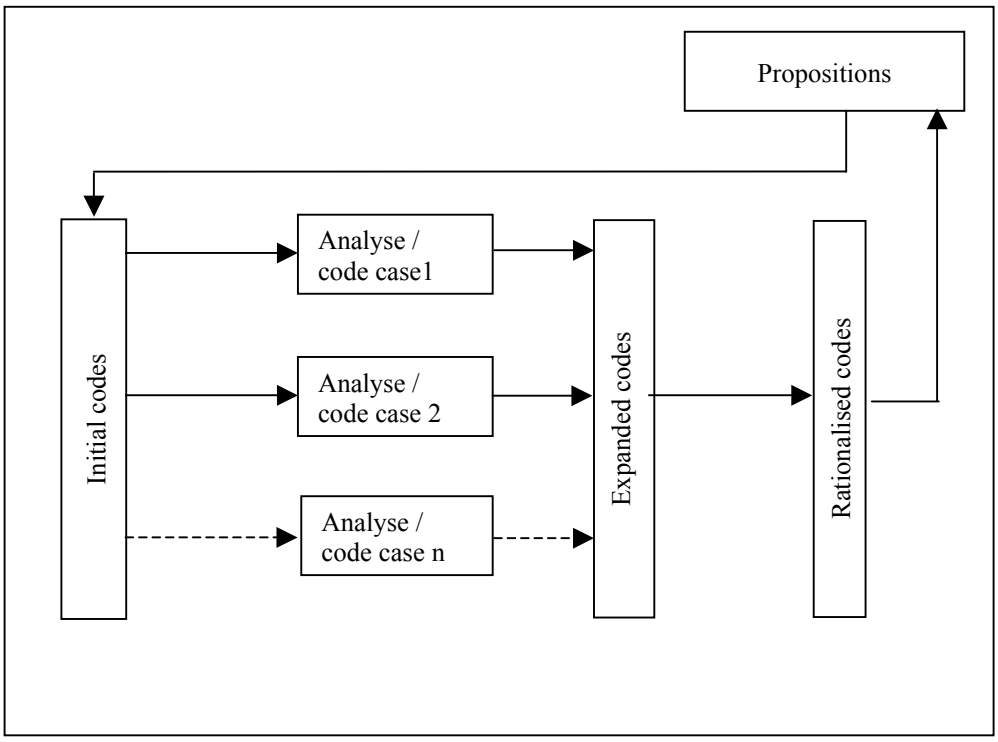


Figure 4: Codes created.

Initial codes

Miles and Huberman (1994) suggest that the generation of an initial set of codes to code the data collected in the case studies can be made by taking into consideration the research questions, hypotheses, problem areas, and/ or key variables. Another approach, and the one adopted here, utilises the propositions proposed in Yin's (1994) model for the starting point for the generation of codes. To create these initial codes each proposition needs to be considered in turn, and an appropriate code or codes needs to be generated to identify aspects of it. The overriding consideration in the allocation of code/s to a proposition is to ensure that the introduced codes addressed as many aspects of the proposition as possible (Dey, 1993). On the other hand, care has to be taken to ensure that the number of initial codes created is not too high. According to Miles and Huberman (1994:58), minimising the number of codes is done to ensure that they are manageable in terms of them being able to be retained in the researcher's short-term memory. It is difficult to state an ideal number of initial codes that should be generated however a number of between 15 and 30 would not be unrealistic.

Once these initial codes are generated, the actual coding process began. This involves examining chunks or segments of data from the case studies and associating them with one or more of the initial codes. However it will quickly became apparent that the initial codes

that have been generated are not adequate to fully code the data from all the case studies. In fact, if the codes that had been initially proposed were used to code all the data from the case studies then, the coded outcomes would have been too general, with the possibilities that coded segments would have been lost in coding (Miles and Huberman, 1994:61). They acknowledge that the codes that are initially developed will often change during the process of the fieldwork (*ibid*). Changes in the initial codes can occur for a variety of reasons including the fact that additional codes are required, others become obsolete, while others emerge during the data-collection process.

Therefore the majority of codes that are eventually used to code the case study data will evolve once the coding has commenced. For example, if an important chunk of data is encountered that could not be adequately addressed by one of the available codes; an additional code should be created. However this means that when an additional code is created, there is the possibility that it may also be applicable as a code in the already coded data. Consequently, the previously coded data has to be revisited to consider the inclusion of any newly created codes. One means to reduce this excessive reprocessing time is to generate as many potential codes before the actual coding process commences. This could be achieved by not only considering the propositions but also by using any pilot interviews to generate the initial codes. It is important to appreciate that the actual number of new codes expands quickly during the early coding process. However the generation of new codes reduces to one or two and, eventually after coding roughly half the interviews, no new codes need to be generated. This highlights the importance of understanding the data (Dey, 1993) and generating the codes as quickly and comprehensively as possible in the early stage of the analysis process. Analysts however should be warned that the number of codes may triple or even quadruple in number. These latter codes are referred to as the expanded codes.

Expanded codes

Once the coding process is completed, the codes that were utilised in the analysis of all the case studies are referred to as the expanded codes. Many of these codes may be literally created on the fly, so the next step is to rationalise these in some way. One way to achieve this is to group the expanded codes, typically according to some logical grouping of the codes. The codes may be grouped because they address a particular theme or they are a logical group of codes; however there is no fixed method to group codes. An example to illustrate one type of grouping of codes is given in Table 1b codes have been grouped according to end-user requirements.

Specific code	Specific code description
ER-EF-ER	Effectiveness in eliciting user requirements.
ER-EF-MO	Effectiveness in modelling end-user requirements.
ER-ME-RE	Effectiveness in meeting end-user requirements in terms of what the analyst perceives to be correct.
ER-RE-DIC	End-user does not know what they want initially.
ER-RE-EX	Determine whether end-user requirements can be met by considering the existing data.
ER-RE-OU	End-user requirements are determined considering the outputs of the system.
ER-RE-WE	End-user requirements were clear to the analyst from the outset.
ER-TI-AN	Time spent analysing end-user requirements.
ER-US-OS	End-user requirements could be satisfied with an off-the-shelf product.

Table 1b: End-user requirements

Rationalised codes

The expanded codes represent all the codes that are utilised during the coding of the case study data. However during the actual coding procedure it can become difficult to conceptualise the increasing number of codes used, and in a number of cases, additional codes may have been created when in reality they may not have been required. Therefore, the next step is to rationalise the expanded codes to remove any duplications or anomalies that may have arisen in their creation. An effective way to do this is to create a frequency

distribution of the expanded codes used. This approach creates a visual representation that illustrates which codes are used in which cases. Table 2 provides an example of the affect achieved when a frequency distribution is created for codes required to code 17 case studies.

				CASE NUMBERS																
No.	**SPECIFIC CODES**	**Comments**	**Action**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
1	ER-EF-ER			■	■			■		■				■	■	■	■	■	■	■
2	ER-EF-MO			■	■		■	■	■	■								■	■	
3	ER-ME-RE				■	■	■							■				■		
4	ER-RE-DIC					■						■			■	■		■		■
5	ER-RE-EX													■		■		■		■
6	ER-RE-OU								■								■	■		
7	ER-RE-WE				■		■													
8	ER-TI-AN		Merge					■		■								■		
9	ER-US-OS				■							■	■	■	■	■		■	■	■
10	MB-AN-IN			■	■	■	■	■	■	■	■	■			■	■	■	■		■
11	MB-AN-IT									■	■	■		■		■				
12	MB-AN-ME			■			■	■	■	■	■	■	■					■		
13	MB-AN-ST	MB-AN-IN	Merge					■												
14	MB-AN-UN	A valid code	Leave					■	■											

Table 2: Frequency of codes used in cases

Criteria to rationalise the expanded codes

The visual outcome of Table 2 illustrates the areas where possible rationalisation of the expanded codes could occur. To formalise this process, the following criteria should be adopted to determine whether a code should be either deleted or merged or left in the expanded code set.

- Delete – the code is not used during the coding process and therefore it is deleted.
- Merge – where a code is duplicated, or where it is very similar to another code, it is merged with the code in the group that is used in the majority of case studies.
- Merge or Leave or Delete – in a number of circumstances a code may only be used once, or twice. Under some circumstances a decision must be made as to whether the code should be merged with another one, allow it should stand on its own; or whether it should be deleted from the list of available codes.

Applying the above criteria will result in a reduction in the number of expanded codes. Typically this may mean a reduction of about twenty-five percent in the number of the original expanded codes. This new subset of codes is referred to as the rationalised codes. Table 3 illustrates an example of the above process to create a rationalised code from a number of expanded codes. The column ‘Reason for Action’ explains why a particular ‘delete’ or ‘merge’ or ‘leave’ action was carried out with respect to a code/s. This example shows that four expanded codes were identified as being similar to one another, and therefore for this example deemed that they could be merged into one of the existing codes. This rationalised process finds similarities between the expanded codes by applying the criteria outlined above. It is this type of logic that needs to be repeated for all the expanded codes to create the rationalised code set.

No.	Expanded Codes	Explanation	Action	Reason for Action	New rationalised Code
8	ER-TI-AN	Time spent analysing end-user requirements.	Merge	All relate to the time to carry out the DSS analysis. MB-TI-WI was the one that was retained as it was used the most and therefore the other codes were merged into this one.	MB-TI-WI
35	MB-SU-TI	The time available for analysis.	Merge		
36	MB-TI-WI	Time to follow written instructions.			
49	SA-TI-AL	The time the systems analyst has allocated to DSS analysis.	Merge		

Table 3: Rationalisation of the expanded codes

Analysing the coded data

After data entry is completed and the codes have been rationalised, the case study data can now be more closely analysed. The nature of any relational database design is that it assists in the analysis process because the designer of the database has already considered theoretical outputs from the database. That is, before the physical database is created, the designer of the database would have considered the structure of the tables. Typically this is achieved by considering, besides other things, the expected output that will occur from the database. This prediction of the output drives the potential findings of the research; however it can also be the failing of the research. In exploratory research, the researcher must be willing to experiment with a variety of outputs/ reports. If this in not done there is the potential that important findings in the data may be simply overlooked as it is difficult to analyse unprocessed data and make appropriate conclusions. The analyst should spend time in generating a variety of reports, regardless of how trivial they may appear to be. Unfortunately it is difficult to be prescriptive in the types of reports that should be generated as this can bias your thinking into ones that you believe may be appropriate. As an illustration, two reports are shown below to guide the novice researcher in the type of thinking that might be appropriate.

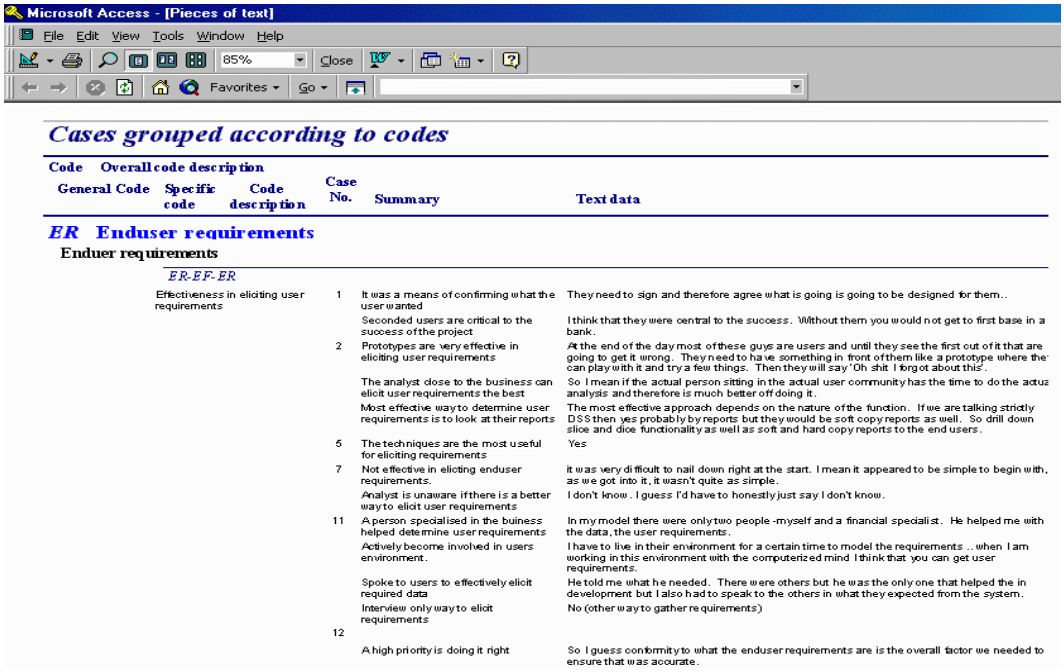


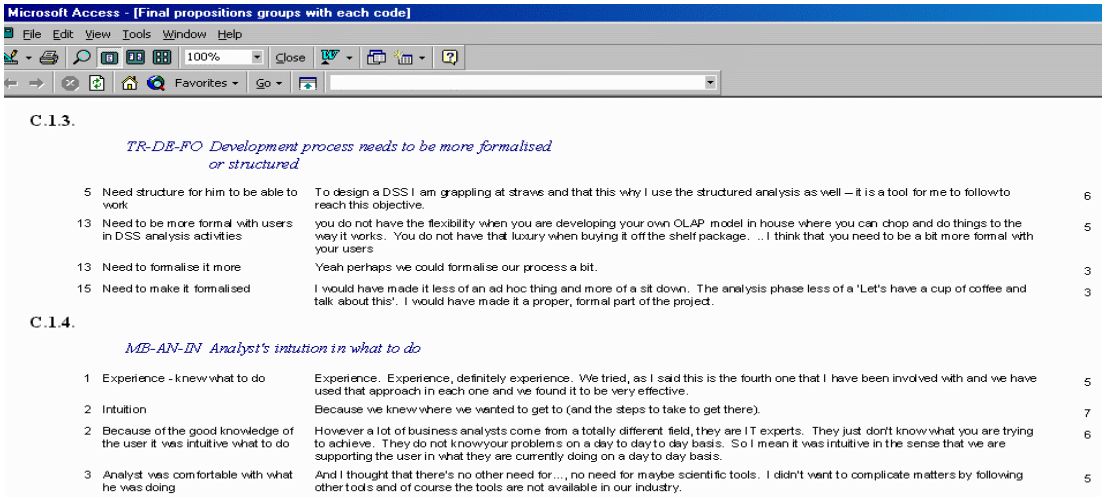
Figure 5: Report by code

The first report should group cases under the respective codes (Figure 5). This report illustrates how each piece of text from the case studies is grouped or coded. That is, when the case studies are analysed, appropriate segments of data are coded into the database.

This report allows for a visual check to ensure that these segments of data are coded appropriately. An example of this report is shown above

.

A second report could group the codes and the case data under the respective propositions (Figure 6). This report succinctly illustrates how the case study data supports each proposition that as indicated is a fundamental component of any case study (Yin, 1994). A part of such a report is illustrated below:


Microsoft Access - [Final propositions groups with each code]

File Edit View Tools Window Help

C.1.3.

TR-DE-FO Development process needs to be more formalised or structured

5	Need structure for him to be able to work	To design a DSS I am grappling at straws and that this why I use the structured analysis as well – it is a tool for me to follow to reach this objective.	6
13	Need to be more formal with users in DSS analysis activities	you do not have the flexibility when you are developing your own OLAP model in house where you can chop and do things to the way it works. You do not have that luxury when buying it off the shelf package. .. I think that you need to be a bit more formal with your users	5
13	Need to formalise it more	Yeah perhaps we could formalise our process a bit.	3
15	Need to make it formalised	I would have made it less of an ad hoc thing and more of a sit down. The analysis phase less of a 'Let's have a cup of coffee and talk about this'. I would have made it a proper, formal part of the project.	3

C.1.4.

MB-AN-IN Analyst's intuition in what to do

1	Experience - knew what to do	Experience. Experience, definitely experience. We tried, as I said this is the fourth one that I have been involved with and we have used that approach in each one and we found it to be very effective.	5
2	Intuition	Because we knew where we wanted to get to (and the steps to take to get there).	7
2	Because of the good knowledge of the user it was intuitive what to do	However a lot of business analysts come from a totally different field, they are IT experts. They just don't know what you are trying to achieve. They do not know your problems on a day to day to day basis. So I mean it was intuitive in the sense that we are supporting the user in what they are currently doing on a day to day basis.	6
3	Analyst was comfortable with what he was doing	And I thought that there's no other need for..., no need for maybe scientific tools. I didn't want to complicate matters by following other tools and of course the tools are not available in our industry.	5

Figure 6: Rationalised codes grouped by the propositions

The benefit of using these computer-generated reports is to ensure a variety of combinations of data can be quickly generated and displayed. It is important that the analyst experiments with a variety of outputs/ reports before proceeding to the next step of analysis.

Final propositions

The analysis of the case study data is centred on the propositions and the rationalised codes (Figure 1). As detailed above, the propositions can be derived from the research questions and from interpreting data from other sources including from the literature and/ or surveys. On the other hand, the rationalised codes are primarily generated during the actual coding of the 'chunks' of data from the case studies. Table 4 illustrates a cross section of rationalised codes that were used in the analysis of data from a particular case study. For this example it shows that only two of the codes are actually derived from the original propositions, the remainder being generated during the coding of the case study data. This later group of codes is referred to as the generated codes.

Rationalised code	Rationalised code description	Ref. to original proposition
ER-EF-ER	Effectiveness in eliciting user requirements.	
ER-EF-MO	Effectiveness in modelling end-user requirements.	Prop. 10
ER-ME-RE	Effectiveness in meeting end-user requirements in terms of what the analyst perceives to be correct.	Prop. 14
ER-RE-DIC	End-user does not know what they want initially.	
ER-RE-EX	Determine whether end-user requirements can be met by considering the existing data.	
ER-RE-OU	End-user requirements are determined considering the outputs of the system.	
ER-US-OS	End-user requirements could be satisfied with an off-the-shelf product.	

Table 4: End-user requirements

The greater occurrence of generated codes is not unusual and one could expect to find up to 75% of all the rationalised codes are created in this way. Compared to the initial codes, these generated codes cannot be directly associated with one of the original proposition. Therefore the final step in the analysis of the case study data is to attempt to link each of the rationalised codes back to at least one of the original propositions. This step may necessitate the creation of new propositions or even the deletion of existing ones that are no longer associated with one of the rationalised codes. By implication the codes that are used in the coding process reflect the essence of the case study data. Therefore the outcome of this step will be a set of final propositions, that have been supported by the rationalised codes and the data collected from the case studies.

The process is to link the rationalised codes (that is the initial codes plus the generated codes) to the propositions. Table 5 illustrates an example of how this can be represented. It involves linking each of the rationalised codes to one or more of the propositions. At the end of this process all the rationalised codes will be associated with one or more propositions. However, it may be discovered that some of the codes cannot be logically associated with one of these initial proposition in which case a new proposition is created (see proposition no. 4). It may also be discovered that one of the initial propositions does not have any codes associated with it; this means that it may be necessary to delete that proposition/s from the final group of propositions proposed (see proposition no. 3).

Number	Proposition	Associated rationalised codes
1	The choice of tools and techniques to be used during DSS analysis is defined in procedures that the systems analyst is able to follow.	MB-AN-IT MB-PR-GU MB-FL-AN
2	The choice of tools and techniques to be used during DSS analysis is guided by the analyst's understanding of a particular methodology for the DSS development process.	MB-AN-ME MB-AD-ME MB-CO-ME
3	The choice of tools and techniques to be used during DSS analysis is directly related to the techniques utilised in the analysis of traditional information systems.	
4		MB-NO-DI TR-DE-FO ER-EF-ER

Table 5: Linking of rationalised codes to the propositions

The outcome of this step represents the culmination of the analysis of the case study data. Specifically, a series of propositions are proposed that that were derived from the data collected through the case studies. In this step the rationalised codes drive the development of the final propositions because it is these codes that truly reflect the nature of the data collected during the case studies.

CONCLUSION

This paper outlined four steps that can be applied in the analysis of data collected through a cases study method. It offers the novice researcher the guidance to be able to carry out these steps by illustrating them with a practical example. The focal point and the outcome of these steps is a series of propositions that are can be used as a means to support the important issues arising from the case studies. Further research could validate these propositions by using the data from the actual cases to support them and then generate a definitive set of findings.

REFERENCES

Alreck, P. L. and Settle, R. B. 1985, *The survey research handbook*. Irwin, Illinois, USA.

AnSWER Analysis software for word-based records (online). 2000, http://www.cdc.gov/hiv/software/answr/howto.htm#Why%20Use%20AnSWR Accessed 26 September 2002

Atkinson, J. S. 2002, *Tools and techniques used during systems analysis activities for decision support systems – an exploratory study.* PhD thesis, Monash University

Barry, C. A. 1998, Choosing qualitative data analysis software: Altas/ti and Nudist compared. *Sociological Research Online* (online), **3:3.** http://www.socresonline.org.uk/3/3/4.html Accessed 20 September 20020

Codd, C. J. 1990, *The relational mode for database management*. 2nd edn. Addison Wesley, Reading

Coffey, A. and Atkinson, P. (1996). *Making sense of qualitative data. Complementary research strategies*. Sage, Thousand Oaks.

Dey, I. 1993, *Qualitative data analysis. A user-friendly guide for social scientists*. Routledge, London

Fowler, F. J. J. and Mangione, T.W. 1990, *Standard survey interviewing*: *minimizing interview-related error*. Sage : Newbury Park CA.

Markus, M. L. 1989, Case selection in a disconfirmatory case study in J. I. Case and P. R. Lawrence (eds) *The information systems research challenge*: *Qualitative research methods.* Harvard Business School, Boston, MA.

Miles, M. B. and Huberman, A. M. 1994, *Qualitative data analysis*. 2nd edn. Sage, Thousand Oaks.

Weitzman, E. A. 2000, “Using computers in qualitative research” in N. Denzin and Y. Lincoln (eds.) *Handbook of qualitative research*, 2nd edn. Sage , Thousand Oaks.

Weitzman, E. A. and Miles, M.B. 1995 *Computer programs for qualitative data analysis.* Sage, Thousand Oaks.

Yin, R. K. 1994, *Case study research*: *design and methods*. 2nd edn. Sage, Thousand Oaks.

COPYRIGHT