Lewins, A.

CAQDAS: Computer Assisted Qualitative Data Analysis

Permission reference: 0761972455(302-323)60275

ISN: 0761972455

COMPUTER ASSISTED QUALITATIVE DATA ANALYSIS 18

ANN LEWINS

CONTENTS

Computer assistance in qualitative data analysis has become a respectable and accepted strategy for the management of qualitative data. This chapter refers to several categories of Computer Assisted Qualitative Data AnalysiS (CAQDAS) software. I will focus mainly, however, on a particular group of

software packages, the ‘code-based theory builders’ (Weitzman and Miles, 1995), and some of the ways they seek to assist the researcher in working with large volumes of qualitative information and data. Theory building software programs will help you manage the data and to manage and interrogate your ideas. They have a dominant place in many academic and applied social research settings.

Transcripts of interviews, descriptions and narratives can produce a lot of text. Handling such data can be disorganised and *messy*. When interpreting and recording significance, commonality, exceptions, or tracking a story, it is difficult to keep in touch with all the ideas you may have about the data. You need to keep in touch with examples of the data that demonstrate those ideas and the connections between those ideas. Whether the data are in textual, audio, video, or graphic format, ideas about them nearly always relate to parts of them. In essence, software seeks to maintain an easy contact between the ideas and those parts, while allowing an overview of the whole.

This chapter shows how some of the tasks of analysis, mostly clerical, associated with the analysis of qualitative data can be assisted by a range of software programs. At the end of the chapter, there is a list of resources, from which comparative or software-specific information can be gathered. Many software packages offer similar tools, but the way in which they are provided varies.

One of the important aspects of software use is the ease with which software can be learnt or taught. This is not quite the same as ‘user friendliness’. Researchers involved in substantive projects will have different needs and resources compared to students. The availability of support and the ease with which a software package can become useful are vital considerations. The package should not become an obstacle to getting on with ‘real work’ with data. It is also necessary to touch briefly on the different working styles of analysis, and the methodological debate surrounding them. Your way of working with qualitative data may spring from deeply held beliefs about the nature of knowledge and existence itself, or simply from project design. If you are going to be a user of software it is important to know and to understand your methodological standpoint first, and then to bring a methodology *to* the software, rather than see the software as being the architect of your method.

18.1 THE DEVELOPMENT OF CAQDAS SOFTWARE

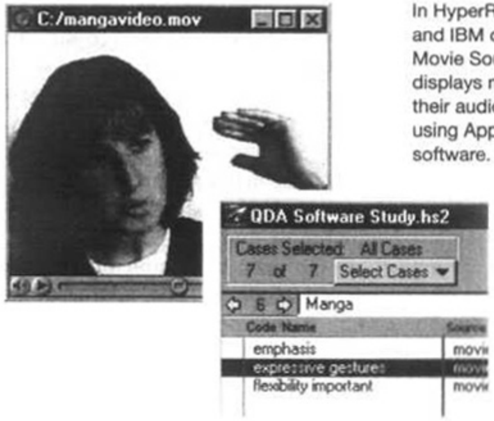
CAQDAS software developed largely from early work by academics who were involved in qualitative data analysis during the late 1980s (Fielding and Lee, 1991/1993). Early programs specifically developed to analyse qualitative (then, mainly textual) data were The Ethnograph (Seidel), Hyper RESEARCH (Biber, Kinder) QSR NUD*IST (Richards and Richards) and a little later ATLAS.ti (Muhr). Some began as collaborative projects with universities but later became independently produced programs which were marketed and distributed by commercial software publishers. The influence of the 'user', too, has been an important factor. In a decade of software development, developers have been very responsive to critique and free upgrades to a package often reflect this. Developments in information technology have enabled an increase in the range of software packages and, similarly, the range and sophistication of tools within each software program.

As packages have been developed to handle textual data in various ways, typologies have been developed to categorise the software programs (Weitz man and Miles, 1995). Major groups of packages concerned with the analysis of textual or other qualitative data have been described as:

> 'Text retriever' and 'Text-based Managers' are mainly concerned with the quantitative 'content' of qualitative data and automatic generation of word/phrase indexes, statistical information on word frequency and the retrieval of text in context. They will often have internal dictionaries and thesaurus facilities. They fall into a wider category sometimes referred to as 'content analysis' packages. Examples include Sonar Professional, CISAID and SIMSTAT. Some content analysis programs, (e.g. Diction 4.0) set out to analyse the 'tone' of speech transcripts to produce, for example, a measure of confidence. (For more examples see http://www.content-analysis.de/quantitative.html)
>
> 'Code and Retrieve' and 'Code-based Theory Builders' have been more concerned with thematic analysis and interpretation of textual data. It is the functionality of the 'code and retrieve' and the 'code-based theory builders' which is used more widely in academic and applied social research settings. Examples include, QSR NUD*IST, ATLAS.ti, HyperRESEARCH, The Ethnograph, KWALITAN, WinMAX. (For more information see http://caqdas.soc.surrey.ac.uk)

The two code-based categories have tended to merge as enhancements to the 'code and retrieve' packages and have caught up with the range of tools available in the 'theory builders'. Packages in these last two categories use the database structure of the software to enable the assignment of 'codes' or labels to chunks of text, and the subsequent 'retrieval' of text segments according to selected code labels. Subsequent interrogation of codes, and their co-occurrence, proximity (or not) etc., in the data allows researchers to test relationships between identified themes and concepts. In the broad typology of software, this extra functionality allows the 'code-based theory building' label to be applied to an increasing range of software programs. The label 'theory builders' must not be misunderstood. No theory is built by the package itself and any theory or conclusions which are drawn are the result of the researcher's own thinking. The addition of networking and modelling tools (see Figure 18.6, p. 316) has added to the researcher's ability to visualise the connections between concepts, issues and themes, while work remains integrated with source data. The development of 'hyperlinking' functionality allows the user to jump instantly between points in the coding schema or text to other related points in the dataset. The incorporation of multimedia data into a working project is now possible in ATLAS.ti, HyperRESEARCH, NVivo, CISAID and CTanks. All make different use of hyperlinks to segments of audio or video files to enable different levels of multimedia integration with the project. Other software developers have resisted the move to multimedia tools (including WinMAX 99 Pro and The Ethnograph Version 5). Such packages have also been modernised and upgraded, but they tend to be smaller in size, easier to learn, and are therefore most useful for those with time constraints or those who can do without the more complex tools. Although the incorporation of multimedia data into projects has increased, the great majority of projects still consider text transcripts of interviews etc. as the best way to access large amounts of data quickly. This is the case even though the transcription of data is a very lengthy process (the use and development of Voice Recognition software has some way to go before it is widespread or fully efficient).

Increasingly, packages do not fit neatly into existing software typologies. ATLAS and KWALITAN both provide code-based theory building tools but also provide quantitative word frequency information, unusual in this category of program. CISAID has strong links with the 'text retrievers' and the family of programs which provide this type of 'content analysis'. It provides sophisticated statistical information about word and phrase content, but at the same time part of its functionality is code-based and this allows it into the 'code-based' category of programs. CTanks, does not fit into any existing category. It provides transcription function keys and early analysis



In HyperResearch (MAC and IBM compatible) the Movie Source Window displays movie files (with their audio tracks, if any) using Apple's Quicktime software.

You may select and code any number of frames, to be replayed when recalling the source material from the Study Window or in a hyperlinked report.

Figure 18.1 *Integrating multimedia files into a project*

tools, allowing the researcher to build hyperlinks from the transcript of commentary to points within the digital sound or video file. Linked multimedia documents remain synchronised with the text commentary; for example, when scrolling a section of the transcript or summarised commentary, the software will play back the relevant sound segment of the file. Multimedia support in HyperRESEARCH software is illustrated in Figure 18.1, where the Movie Source Window sits side by side with a coding window.

18.2 ANALYTIC PARADIGMS USING QUALITATIVE DATA

Qualitative researchers approach projects, the analysis of data, and the use of software from a range of methodological and epistemological perspectives. Many specifically refer to 'Grounded Theory', a 'general method of constant comparative analysis' (Glaser and Strauss, 1967; Strauss and Corbin, 1994). Ideas are generated from what is seen after reading and re-reading the text. You may need to assign labels **(codes)** and annotations or memos to the data, based on your understanding of the data (see Chapters 9 and 14). This might be followed by the refinement of codes into concepts and categories in order *later* to generate theory that is thoroughly grounded in the data. Using similar methods, you may instead be using a less clearly defined descriptive, interpretative or comparative approach. You may start with questions you need to answer, but the theoretical or other framework of your findings will be what your analysis is working towards, not the starting point of the project.

In Layder's 'adaptive theory', he suggests that the process of coding can begin with sensitising concepts and ideas, as 'orienting devices' (Layder, 1998). These concepts are selectively plucked from existing theories to 'crank up' the theory building process, while always retaining 'theoretical openness' to new concepts arising from the data. He suggests that since there is a general acknowledgement that any observation and interpretation is **theory laden,** this means that starting the coding process from a 'clean slate' or *tabula rasa* is difficult. It is better therefore to be explicit about any theories that are contributing a priori to your ideas and knowledge about the subject that you are beginning to study.

Your project and its design may be completely pre-defined, or heavily influenced by theory. An example of this approach is included below (Speller, 2000). The coding framework is initially entirely built around the contributing theoretical concepts.

If you are involved in discourse analysis, you may be interested in language, conversation or discourse as social constructions or cultural expressions (see Chapter 19). Such an approach could be illustrated by taking a hypothetical approach to a recent e-mail exchange on 'qualsoftware' (the Internet discussion list created to enable debate and support the users of qualitative software). One subscriber used the word 'girls' in the context of a comment about people who do transcription. The use of the word 'girls' created an immediate and angry debate. If you used a constructionist, feminist approach to the analysis of this e-mail data, the use of the word '*girls*' within its context here, would be interesting. Thus, the annotations you make to the data may be more concerned with language, patterns and structures of the written word than with the concepts identified by someone analysing the outcomes of the more subject-specific debate concerning transcription *per se*, from the same data.

18.2.1 The suitability and limitations of software

Most programs are geared towards accessing large amounts of data and those smaller segments taken from the whole that represent examples of your thinking. The emphasis on large quantities of data places constraints on the flexible handling or marking up of individual data files. Although this is changing in some programs, you may have to work with 'Text only' formats (plain text – without word-processing characteristics, such as bold and underlining). Often data cannot be freely edited or annotated. The micro examination of a small dataset (for instance, for conversation analysis), is more likely to be hindered by the use of anything more complex than a word processor. The tracking of a sequence of events that is talked about in a disorderly way throughout a narrative, as in a life history, may not be handled with enough sensitivity with current 'theory building' software, Coffey, Holbrook and Atkinson (1996) consider the suitability of existing theory building packages for analysing narrative data, where reduction or fragmentation of the data can be unhelpful.

The **discourse analysis** of large volumes of news reportage might be better handled with a 'content analysis' or 'text retriever' package. This will depend on the researcher's individual needs and style of analysis Sometimes what is required is a more quantitative analysis of content, together with the ability to retrieve text in context. Researchers often express a need for the early support of such quantitative analysis, followed by the Inter ability to assign codes based on identified themes as well. If this is the case, they need a software which contains elements of both the main categories of software described above (section 18.1), or they need two different packages.

Whatever approach you adopt, you should be clear about your needs before deciding to use a software package just because 'most people seem to use one' (see section 18.4). When thinking about making use of CAQDAS software it is useful to prioritise a list of requirements that suit the way you will prefer to work. If it is most important for you to be able to 'mark' the data as you analyse it, then make sure the package you chose will give you edit rights on your data files. If you are more concerned with quick navigation between points in the data than with data reduction in the text, make sure you can easily build hyperlinks between those points in the source data. Many of the tools are common to all theory building CAQDAS packages, but some functions exist only in a few or are exclusive to one.

18.3 How 'CODE-BASED THEORY BUILDING' SOFTWARE CAN HELP

Theory building CAQDAS software packages do not dictate the way or the order in which you might perform various tasks, but they might influence you in terms of the complexity of the tasks you undertake, and your readiness to perform them. Furthermore, they will encourage you to feel that you can change your mind about your analyses. Some of the tasks that they can assist with are listed in Table 18.1.

The main tasks and tools of CAQDAS software are explained in general and later the application of some of the tools will be illustrated and commented upon with three examples. The first is from applied research using a dataset of focus group transcripts. This was the qualitative phase of an assessment of the needs of elderly people commissioned by a local health authority. I was asked to use ATLAS.ti software to assist with my management of the data. The second example is a longitudinal, academic project, with a very large dataset, using QSR NUD*IST. The project, centred around the relocation of the entire mining community of Arkwright, examines notions of identity over five time phases (Speller, 2000). The third is a dataset created for use when teaching about qualitative software. It reports on a group of respondents' views of their community. The three examples each take quite different approaches to the development of coding frameworks and the role of theory. Generally, programs do not mind which approach you take. You can **open code** the text, as you work though it, or you can create codes in advance which focus your attention on theoretical issues and their occurrence in the data. You might work in a mix of both ways. Providing you are explicit about the way you have worked, and you can justify your choices, software will assist you to work in the way that suits you.

Table 18.1 *The main tasks of analysis, assisted by code-based theory building software*

1. Exploration and discovery by searching (for strings, words, phrases in verbatim content).
2. Coding during, after and leading to discovery (sorting, organising, categorising the data).
3. Theory-based coding.
4. Retrieval of coded segments.
5. Adding analytic memos.
6. Making visible indicators in the text.*
7. Linking to other parts of text and other files.*
8. Searching the database and the coding schema, testing ideas, interrogating subsets.
9. Mapping ideas, modelling processes and connections.*
10. Generating reports/output (print-outs or readable files in other applications).

* Not always available in all software programs.

18.3.1 Sorting, organising, interpreting, identifying concepts – categorising

Even if you were working just with transcribed data on paper, you might want to annotate the data, write notes or scribble in the margin. You might highlight sections of the text in different colours to signify different themes or topics. If working on a word-processor with textual data, you might copy these sections of the text under different headings or into different files. Usually you will reduce the amount of text that you are dealing with and, to a certain extent, you will be lifting those **copied and pasted** sections of text out of their context.

Code-based retrieval packages will allow you to apply keywords/phrases (codes) to passages of text and put a label on the significance of a section or segment of the data. You can apply as many codes as you like to the same segments of text. The methodological approach being taken will affect how codes and themes are drawn out from the data. In other words, how the **coding schema** is to be established may be a vital outcome of project design. This in itself is not a software issue, although the flexibility with which you create, modify and refine the coding schema will be improved by the use of software. Codes might be created early on as part of a coding framework that arises from the objectives of the project and are then assigned to data where relevant (a 'top-down' method of coding). If your project is defined by theory, you may establish a coding schema which contains the main elements (e.g. sensitising concepts) from that theory before you begin to code the text. Or codes might emerge from, and be assigned to, segments of text during close work with the data (a 'bottom-up' approach to code creation). Subsequently, codes may encompass more abstract concepts from collections of issues and themes which seem to be related in some way.

The coding process will also allow you to **organise** the data if there are already significant known values that can be ascribed to the respondents – the sex and age of the respondents may be important, for instance, In Speller's example below, it was important to compare data across the five time phases of her longitudinal dataset. The five time phases were codes or values under the variable 'Time'. Each time phase code had to be assigned to the relevant data files. Such organisational coding can he a very important aspect of the coding process in any longitudinal project or where comparison across the source of data or type of respondent will be useful.

18.3.2 The coding schema

However codes are generated, and whatever methodological approach underpins this process, the ‘coding schema’ is the manifestation of the way they are listed or organised within the software program. Most theory building software programs encourage an inherently hierarchical organisation of the coding schema. In these packages, the hierarchy has a number of useful purposes later, when in retrieval or searching modes. For a few programs, such as ATLAS.ti and HyperRESEARCH, the default structure of the coding schema is not hierarchical. Hierarchies or the appearance of hierarchies can be imposed in some windows, but in the main ‘code-lists’ codes will be listed at the same level.

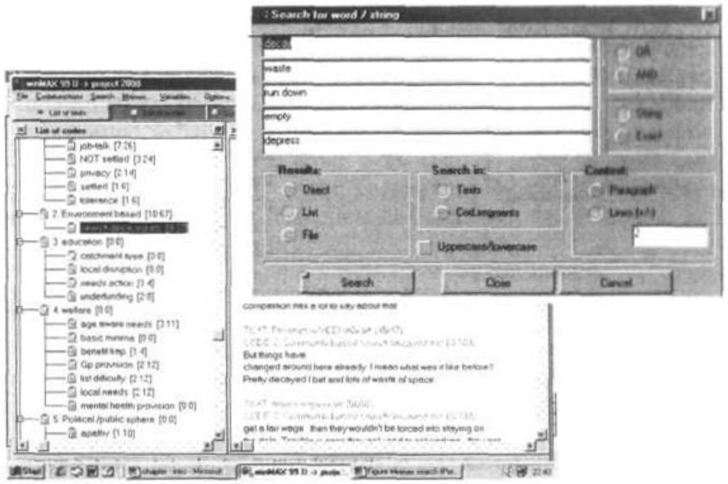
18.3.3 Retrieval

The software enables the **retrieval** of passages of text based on the presence of codes. You can then examine the retrieved segments, thus reducing the amount of data and stepping back from the dataset as a whole in order to focus on certain aspects. Retrieval can happen while in the software, in full contact with other functions, enabling, for example, the re-coding of segments of memoing. Or retrieval can be sent to output or report files which can be opened in other applications (word processor, spreadsheet applications, etc.).

18.3.4 Exploration and discovery by text searching

Discovery achieved by reading and re-reading is likely to be the most thorough method of exploring qualitative data. With large amounts of data, however, this may be impracticable without using the software to locate words and phrases that signal particular topics of interest. This tool is relevant only to textual data and varies slightly from one program to another. The differences, contexts and debate concerning the use of this tool are discussed at length by Fisher with reference to both code-based and text retriever categories of software (Fisher, 1997: 39–66). With most programs, you have the opportunity to search for several ‘strings’, words or phrases at the same time. The ‘finds’, or ‘hits’, can usually be saved, and then coded or autocoded. All textual references containing the finds will be stored where you choose to put them, in the coding schema.

In Figure 18.2, I searched the sample dataset I use when teaching about WinMAX software. I was interested in the words, strings or phrases that were used by the respondents when they were talking about wastage and decay in their community environment. The search retrieves segments containing these strings. I could reject the finds which are off-topic. Of course, this sort of search will not find where the speakers or respondents are talking *around* the subject without actually using any of the words put into the search dialogue box, so this tool is not a safe way to uncover everything said about an area of interest. It is, however, a fairly efficient way to look for buzz words or the use of jargon. The advantage of using CAQDAS software over searching files in a word-processing application is that in addition to finding the words, the CAQDAS software usually enables you to save those finds, together with some surrounding context, in a slot in a coding schema.



compension has a lot to say about that

But things have
changed around here already. I mean what was it like before?
Pretty decayed I bet and lots of waste of space.

get a fair wage then they wouldn't be forced into staying on

The search dialogue box, and the saved ‘hits’ at a code, showing the text segments found (across the whole dataset) in the bottom right-hand pane.

Figure 18.2 *The Text search tool in WinMAX 99 Pro*

18.3.5 Making visible indicators in the text

Developments in a few software programs mean that the researcher can use Rich Text Format textual data, rather than plain, ‘text only’ formats. This means that you can integrate normal coding techniques (as above) with visibly ‘marking’ the text by colour coding or underlining, changing font size, etc., as you might if working in hard copy or with a word processor. This might be a suitable way to handle smaller datasets of narrative files.

18.3.6 Searching, making ‘queries’ – testing ideas and theories

The software can help you to ask questions about the relationship between codes as they occur in the text. Do the themes you have assigned to the text appear together? Do they appear close to each other? Mow do they compare

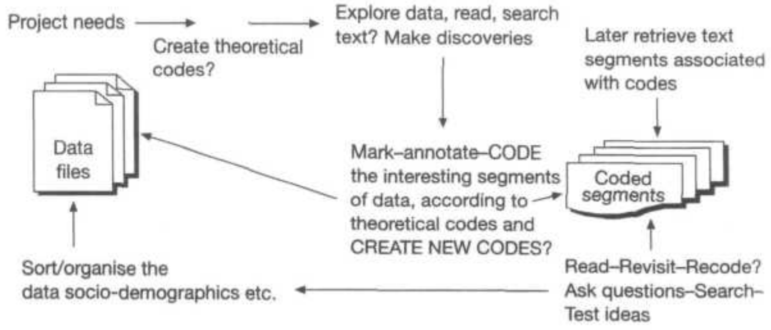

Project needs
Create theoretical codes?
Explore data, read, search text? Make discoveries
Later retrieve text segments associated with codes
Data files
Mark–annotate–CODE the interesting segments of data, according to theoretical codes and CREATE NEW CODES?
Coded segments
Sort/organise the data socio-demographics etc.
Read–Revisit–Recode? Ask questions–Search–Test ideas

Figure 18.3 *Summarising the discovery/coding process*

across different sets of data and across the different groups of respondents? Before this type of searching and question asking can occur, you must have arrived at the point where you have achieved enough consistent coding to rely on the results of the queries you make. Figure 18.3 summarises the processes of discovery, coding and testing ideas.

18.3.7 Analytic memos

The ability to write analytical or procedural notes while working is provided in a number of ways. In some programs you can place annotations at points in the data. In others you are allowed to create one memo per data file. In most packages, you can also write comments or memos to explain codes and concepts – how they are defined and how they change. Memo-writing is an important aspect of the management and continuity of analysis.

18.4 EXAMPLES: HOW THE SOFTWARE HELPED

The working examples described below give some idea of some of the processes followed in two quite different projects. Both use theory building software, but they had different starting points and made use of different coding structures.

18.4.1 Example 1: the analysis of focus group interviews using ATLAS.ti

I had been given a relatively simple brief, to pull out the dominant themes from transcripts of focus groups generated in a health authority 'Needs Assessment for the Elderly' exercise. At this stage, the client had no theoretical concerns and simply needed to be given a list of the themes as I saw them and the segments of data supporting them. Further analysis and writing-up would be based on the material I handed over. The client was also interested in having a measure of how frequently some words and expressions were used. Each group's discussion covered the main concerns that had been flagged up at the beginning (e.g. day care, health care, transport, personal safety). The client had requested that ATLAS.ti be used to assist in the process.

Analytic memoing

These concerns formed the starting point for a topic based group of memos where I wrote down absolutely everything worth noting as I read the data. In ATLAS.ti, drop-down lists of 'Files', 'Text segments', 'Codes' and 'Memos' provide interactive connections to source text during most analytic tasks. For example, I can be writing a memo while reviewing a code and the 'quotations' (text segments) to which it has been assigned. I also kept a central 'journal' memo, where I summarised my work each day, together with the questions and leads that I wanted to follow up at the next working session. As I began to develop other concepts while coding more data, I would begin a new memo in order to explain ideas about them as they occurred. Memo tools are available in all code-based theory building software programs and in some (including ATLAS.ti) you can locate a memo at a point in the text itself (though it will not be physically embedded in or cluttering up the text).

Word frequencies

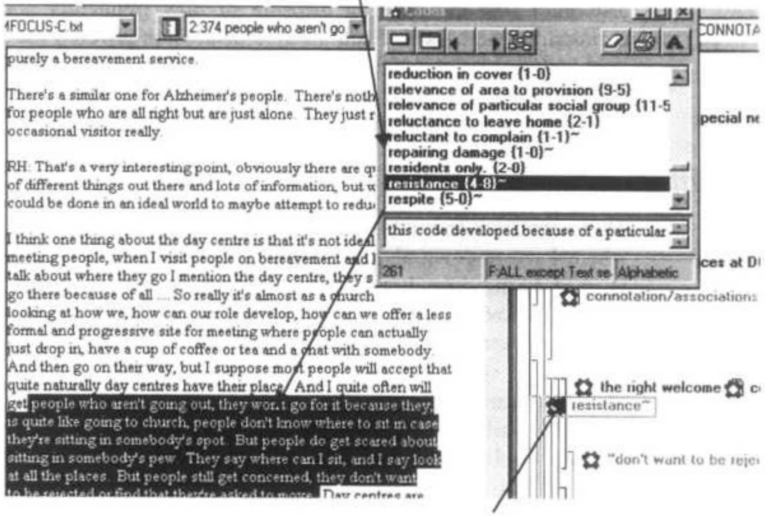
Using the 'Word frequency' tool, I generated tables in which counts for every word in each data file were sorted both alphabetically and in frequency.

Coding

As I read the data I coded the flagged themes as they occurred in the data. This was straightforward. There were more subtle and interesting things in the data that seemed to cross the boundaries of all the major discussion themes. The issue of 'information', its accuracy, relevance and most importantly its source's respectability as perceived by elderly clients, was recurrently mentioned among the social service and care professionals. 'Resistance' to services was connected to the issue of information, but this was another concept that crossed all the main 'thematic' boundaries. So I proceeded with the coding of the data at several levels: obvious themes and more subtle observations. Figure 18.4 shows that in ATLAS.ti, once 'codes' have been assigned, the margin display approximates the way a file looks when marked up in hard copy.

In addition, the codes in the margin provide interactive connections to the sections of text to which the codes relate (in ATLAS.ti a text segment is

One way to assign codes to text: – *drag* the code on to *selected* text



purely a bereavement service.

There's a similar one for Alzheimer's people. There's noth
for people who are all right but are just alone. They just r
occasional visitor really.

RH: That's a very interesting point, obviously there are q
of different things out there and lots of information, but w
could be done in an ideal world to maybe attempt to redu

I think one thing about the day centre is that it's not ideal
meeting people, when I visit people on bereavement and I
talk about where they go I mention the day centre, they s
go there because of all .... So really it's almost as a church
looking at how we, how can our role develop, how can we offer a less
formal and progressive site for meeting where people can actually
just drop in, have a cup of coffee or tea and a chat with somebody.
And then go on their way, but I suppose most people will accept that
quite naturally day centres have their place. And I quite often will
get people who aren't going out, they won't go for it because they,
is quite like going to church, people don't know where to sit in case
they're sitting in somebody's spot. But people do get scared about
sitting in somebody's pew. They say where can I sit, and I say look
at all the places. But people still get concerned, they don't want
to be rejected or find that they're asked to move. Day centres are

reduction in cover {1-0}
relevance of area to provision {9-5}
relevance of particular social group {11-5
reluctance to leave home {2-1}
reluctant to complain {1-1}~
repairing damage {1-0}~
residents only. {2-0}
resistance {4-8}~
respite {5-0}~

this code developed because of a particular

Once coding has been completed, *selecting* the assigned theme label or code in the margin highlights the relevant segment of text

Figure 18.4 *Coding a data file in ATLAS.ti and reviewing a margin display showing how the file has been coded so far*

described as a 'quotation'). Selecting the theme label or a code in the margin highlights the relevant quotation. In the small section of text, about attendance at day centres, one can see in the margin display that the codes, *connotations, Resistance* (i.e. resistance to social services and help) '*the right welcome*' and '*don't want to be rejected*' have been assigned separately but close together in the text. I used quote marks around some codes to remind myself that some of these began life as *In vivo* codes, that is codes which used the actual language of the respondent. Short phrases and words can be dragged into the code list, where they become codes, linked to the originating text.

Having coded the whole dataset in a first pass through the data, I could now work in several ways.

Simple retrieval

I could examine particular themes in isolation by retrieving the data associated with them. In Figure 18.5, a partial report or output document can be seen in which all examples of a selected code were collected together, from across the whole dataset, lifting the text segments out of their original context.

8 quotation(s) for code: *Resistance*

P 1: FOCUS–B.txt – 1:139 (213:220) (Super)

. . . and I often usually ask do you go to the day centre sometimes. 'Oh no' and I find that one of the first signs of loneliness is resisting things of that kind and I'm sure it needs people being brought in by somebody, rather than, I don't know if that sort of person is going to come simply by having facilities here.

P 2: FOCUS–C.txt – 2:82 (242:245)

Day centres are fine for day centre people but there are quite a number of people who won't come to a place like this and therefore somewhere in the community we need to look for other places.

P 2: FOCUS–C.txt – 2:374 (237:242)

... people who aren't going out, they won't go for it because they, is quite like going to church, people don't know where to sit in case they're sitting in somebody's spot. But people do get scared about sitting in somebody's pew. They say where can I sit, and I say look at all the places. But people still get concerned, they don't want to be rejected or find they're asked to move.

P 3: FOCUS–A.txt – 3:389 (107:112)

I mean offering places at the day hospital I don't think will solve this underlying case because they will categorically refuse. They don't want to leave their home because they think that telling them to leave their home is taking their independence. They're attached to their home and you're trying to move them out, so I think there's very little care which goes into people's homes.

P 3: FOCUS–A.txt – 3:390 (116:119)

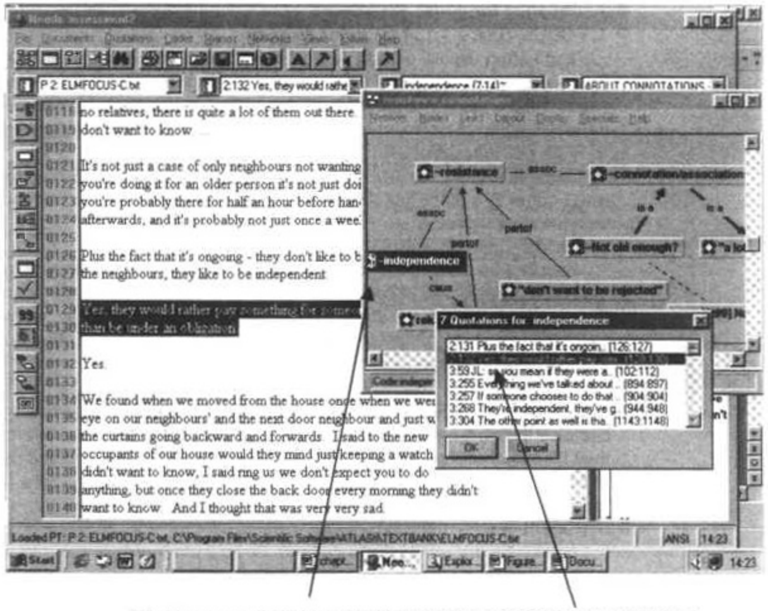
. . . particularly for professional people, they don't see it's for them and they won't even go and try because they've dismissed it as a bingo playing sing-along, It doesn't offer anything for them at all.

Figure 18.5 *Simple retrieval of text associated with the code,* Resistance

In ATLAS.ti, you can also remain linked directly to the whole context by choosing instead to double click on a code to 'List the quotations' in a small window. By clicking once on each quotation, you can sec the quotation highlighted in the whole context of the file in the main window. A similar linkage between codes and text can be found in the network view (Figure 18.6), which helps with mapping ideas and concepts. When analysing focus group data, understanding the context is particularly important because of the collective and individual influences on the way that views are expressed.

Coding strategies and refining codes and concepts

In this project, I was unsure about the level of detail I needed to pick out from the data and represent in individual 'codes'. So at the first stage of my analysis, I was noticing and remarking and labelling everything I found interesting. To use a Grounded Theory term, 1 was **open coding.** Codes generated in this manner began as all having equal status. Initiully. I was anxious that I might forget the detail I was seeing. I begun to accumulate a lot of codes. As I began to refine the coding structure, I realised that *Resistance* was becoming a core concept, to which other themes could be related. Thinking more hierarchically, *Resistance* was a higher order concept within which other codes could be collected. The software could enable me



P 2: ELMFOCUS-C.txt

2:132 Yes, they would rathe

```
0118 no relatives, there is quite a lot of them out there
0119 don't want to know
0120
0121 It's not just a case of only neighbours not wanting
0122 you're doing it for an older person it's not just doi
0123 you're probably there for half an hour before han-
0124 afterwards, and it's probably not just once a wee
0125
0126 Plus the fact that it's ongoing - they don't like to b
0127 the neighbours, they like to be independent
0128
0129 Yes, they would rather pay something for someo
0130 than be under an obligation
0131
0132 Yes
0133
0134 We found when we moved from the house once when we wer
0135 eye on our neighbours' and the next door neighbour and just w
0136 the curtains going backward and forwards  I said to the new
0137 occupants of our house would they mind just keeping a watch
0138 didn't want to know, I said ring us we don't expect you to do
0139 anything, but once they close the back door every morning they didn't
0140 want to know  And I thought that was very very sad
```

7 Quotations for: independence

```
2:131 Plus the fact that it's ongoin.. (126:127)
3:59 JL: so you mean if they were a.. (102:112)
3:255 Everything we've talked about .. (894:897)
3:257 If someone chooses to do that .. (904:904)
3:268 They're independent, they've g.. (944:948)
3:304 The other point as well is this.. (1143:1148)
```

Loaded PT: P 2: ELMFOCUS-C.txt, C:\Program Files\Scientific Software\ATLAS\TEXTBANK\ELMFOCUS-C.txt

Working on a map or 'network' containing concepts generated from data, in ATLAS.ti.

Figure 18.6 *ATLAS.ti 'Network' (mapping) tool showing interactive links between network, codes and text*

to treat this new higher order concept in several ways. One method would be to 'link' the codes together in a Network, with the type of links signalling the nature of the relationship between the collection of codes and the higher concept. Another way would be to make structural changes to the coding schema by merging the detailed codes into *Resistance.* I eventually chose to create a *Resistance related* 'family' of codes, which could then be handled at the detailed sub-code level or at the broader macro-level of the collection or 'Code family'. Yet another similar option is to save a search expression using the 'Query' tool and the 'Supercode' tool, producing repeatable searches for a collection of codes in the data. Supercodes represent complex queries or combinations of codes, and are listed in the normal codes list. Selecting a supercode in the list re-runs the complex search. As part of the refining process, however, I also chose to merge several codes that represented much the same things but had been given different labels. All code-based theory building software will enable you to carry out such merging. Moving away from straightforward 'collections' of codes, I also mapped out some connections that began to express the nature of the relationship between codes (see Figure 18.6).

Mapping concepts and processes

Visualising connections between concepts and codes became part of the coding and 'note-taking' process. I had to be careful, because drawing a link, expressing, for example, a causal relationship, can be appropriate for one situation but not for another, and eventually these links needed to be broadly generalisable. I therefore made loosely associative connections to start with.

A mapping tool is also available in QSR NVivo, though with a different subset of tools. There are also packages, like Inspiration and Decision Explorer, that provide mapping facilities without direct integration with qualitative data. QSR NUD*IST 4.0, which does not have a mapping facility, has a direct interface with both Inspiration and Decision Explorer.

Hierarchical coding schemas

Some programs like WinMAX and QSR NUD*IST have inherently hierarchical coding schema that encourage the researcher to think in a structured way about coding frameworks. In ATLAS.ti, you have to arrange any hierarchical structure and subsequently you can only clearly visualise that hierarchy within specially arranged Network views. Thus the systematic, tidy support that a hierarchical coding schema gives you is lacking. On the other hand, many researchers feel constrained when it is not possible to move codes around freely to make visual connections that express more than just a hierarchical relationship.

To some extent, a hierarchical coding structure can make life within the software package simple. In WinMAX, the 'activated' hierarchy of codes provides very quick access to a set of related themes, and the text segments associated with them. The detailed smaller picture is thus displayed within the inclusive broad-brush picture, with just one click of a mouse button. Complex comparative qualitative cross-tabulations can be generated in QSR NUD*IST using coding hierarchies, as can be seen in the second example (section 18.4.2).

Searching and asking questions (queries) *to test ideas*

Whatever the inherent structure or lack of structure in a coding schema, the researcher is always able to test ideas, and relationships between themes and issues, by asking questions and performing searches of the database. Some examples of questions asked in this project were:

- Find where *Resistance* co-occurs with Independence.
- Find where *Resistance* is talked about – but only Miming 'Professional care givers' data.
- Find where the men talk about age relevance (when discussing day centres).

Every time you ask a question, the results can either be saved in a report file or, with some programs (NUD*1ST, WinMAX and NVIvo), the results can be coded and then integrated into an existing coding schema, as a secondary layer of coding and analysis. These new codes can then be used inside other questions. In ATLAS.ti, you can save the question itself as .a 'supercode', as mentioned above.

18.4.2 Example 2: a longitudinal project, using QSR NUD*IST

In the second example, the shape of the coding schema was a key element in the way a very large dataset was managed. The hierarchical, structured nature of coding schema in QSR NUD*IST provided particularly powerful searching tools for Speller when analysing more than 100 interview files in her longitudinal study, 'The Relocation of Arkwright' (Speller, 2000). The project examined the relocation of an entire community of 177 householders to a new village (because of subsidence). The project involved the comparative study of 25 villagers during the relocation process, starting three years before the move and ending two years after the relocation. Interviews were carried out at five points in time. Speller's strategy of code creation was initially based entirely on concepts derived from a tranche of existing theory.

The role of theory in the project design and analysis

One of the main purposes of this study was to 'examine residents' place attachment in old and New Arkwright and to isolate aspects of the person–place transactions which may have affected their ability to detach from the old environment and attach to the new environment' (Speller, 2000: 81–3). The households were to receive a high level of compensation and had moved from a working-class environment to a middle-class one. Was the transition a positive or negative experience for residents? Was it seen as a facilitator or inhibitor to forming place attachments and identity? The project was underpinned by a theoretical framework, major elements of which were Place Attachment theory (Brown and Perkins, 1992) and Identity Process theory (Breakwell, 1986, 1992). There was little empirical work available on either theory, so Speller hoped that this study would add to the corpus of work in the area and ultimately refine current theories. Consequently, the principles of each theory were incorporated in a coding schema; a *top-down* approach was used to build the coding framework before coding of the text started. All theory building programs support this approach, but the size of the dataset, its longitudinal structure and the types of question Speller needed to ask made the structure of NUD*IST, its hierarchical coding schema and the nature of its search tools particularly useful.

Coding

Once theoretical concept codes had been created, Speller needed to read the text carefully and assign those codes to relevant chunks of text. She also remained open to new issues and therefore developed codes and concepts grounded in the data. The coding process took six to nine months, even with the help of the software. But when she had finished, she had ensured flexible access to a large body of coded data. She describes the next stage:

> The next stage in the analysis was to print out all the collected quotations within one category and to reread participants' responses in this concentrated form. This highlighted complexities within the category, including nuances of differences which frequently resulted in splitting the category or merging it with others. This process of searching for relationships between themes and emerging patterns also brought to the fore inconsistencies in participants' accounts. These often represented or were an indication of the process of adaptation to a new environment and were therefore considered an important element for inclusion in the analysis. (Speller, 2000: 147–8)

Organisational coding

QSR NUD*IST enables the factual/quantitative coding of a large corpus of data with an 'Import-table' tool. A spread sheet, consisting of a matrix of document names, variables and values not only builds that area of the coding schema, but assigns codes to documents based on that information. Using this facility, the time phase information, the respondents' names, how long they had lived in Arkwright, whether they rented or owned.their homes, etc., were assigned to each document. Once the information had been imported, the researcher could (a) bring individual respondent's live time files together in order to see the respondent's data as a whole, and (b) evaluate change, perceptions of security, etc., across all time phases, either by an individual or sets of individuals, or across all or some individuals within a time phase.

Some sample questions and searches are:

How did talk about 'Previous rituals' feature, but just in Time 5 data for all respondents?

A simple *Intersect* search will produce this answer and can be similarly obtained in most theory building programs.

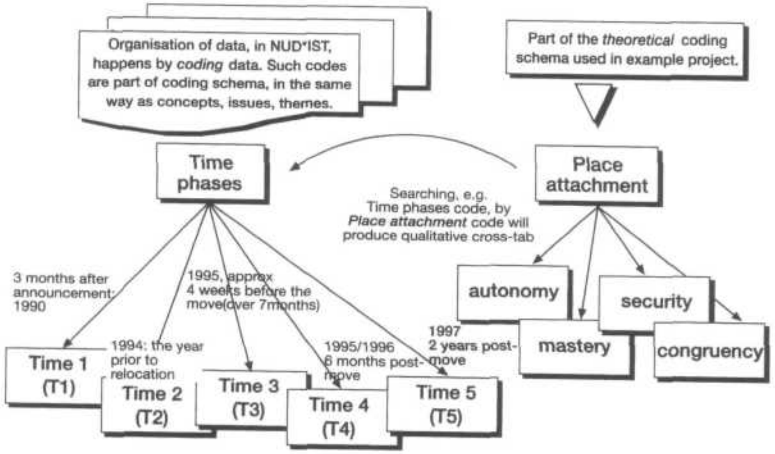
Using the longitudinal dimensions of the data:

How did talk about 'Previous rituals' feature, compar between Time 4 (6 months after the relocation) and Time 5 (2 years after) in data for all respondents?

In QSR NUD*IST, a 'vector' type Index March would produce this result in one step.

An example of a more complex question is:

Did Mary talk about her sense of 'security' in T1, T2 or T3 and also at T4 and T5? If Mary was positive about her sense of security in T1, 2 or 3, did she experience a sense of mastery at T4 and if so, was this sustained or not at T5?


Organisation of data, in NUD*IST, happens by *coding* data. Such codes are part of coding schema, in the same way as concepts, issues, themes.
Part of the *theoretical* coding schema used in example project.
Time phases
Place attachment
Searching, e.g. Time phases code, by *Place attachment* code will produce qualitative cross-tab
3 months after announcement: 1990
1995, approx 4 weeks before the move(over 7months)
1994: the year prior to relocation
1995/1996 6 months post-move
1997 2 years post-move
Time 1 (T1)
Time 2 (T2)
Time 3 (T3)
Time 4 (T4)
Time 5 (T5)
autonomy
mastery
security
congruency

Making use of hierarchical structure of coding schema. Time phase codes have been applied to documents from each time phase, while the concepts found in files, have been coded to relevant segments of text across all files irrespective of time phase. (Graphic generated in Inspiration, a mapping tool with an interface with QSR NUD*IST.)

Figure 18.7 *Part of the longitudinal coding schema in QSR NUD*IST*

To assist in answering this question, QSR NUD*IST will produce a breakdown, or qualitative cross-tabulation by all time phases, of all the different elements of place attachment theory (including *sense of security* and *mastery*), built into one report. Such a level of output might require 20 or more separately performed searches in other programs (see Figure 18.7).

18.5 CONCLUSION

Not all analyses can or should happen within the program. This may be obvious, but it is important to make the point since so much of our lives revolve around the expectation that computers are part of the way we work. Alan Simpson commented in a recent e-mail discussion on his approach to qualitative analysis in a project on how community mental health nurses try to meet the needs of people with severe mental illness.

> I am slowly beginning to make use of NVivo but most of my thinking and ‘analysing’ has been going on during interview and during transcription which I do myself. I find myself writing notes to myself, making vague or not so vague connections at the time or later as I am driving home, etc. – all long before I import into NVivo and start coding.

Michael Agar writes, in an essay that critiqued software in the light of his own needs,

> That critical way of seeing in my experience at least, comes out of numerous cycles through a little bit of data, massive amounts of thinking about that data, and slippery things like intuition and serendipity. An electronic ally doesn’t have much of a role to play.
>
> I need to lay a couple of stretches of transcript on a table so I can look at it all at once. Then I need to mark different parts in different ways to find the pattern that holds the text together and ties it to whatever external frame I am developing. The software problem would be simple to solve. You’d need to be able to quickly insert different colored marks of different kinds at different points so you could see multi-threated DNA laid on the text so you could look at the patterns that each threat revealed and then the patterns among the patterns. (Agar, 1993; 193)

Agar describes viewing data on paper, wanting more context and more data at a time. The use of software will probably never fix this particular problem. Agar’s reservations were expressed in the context of the discourse analysis of one trial transcript. Analytic needs when working with many cases of files may be different. However, there is nothing to stop you working in a combination of ways, whatever the sources of data. Why not work with the software for certain tasks but have paper versions of the data on hand? Agar’s essay was first published in 1991; much has changed since then, and at least two programs, ATLAS.ti and QSR NVivo, have added tools that seem to have been reactions to parts of Agar’s critique (mapping tools and colour coding). So nothing remains the same for long.

In my day-to-day work, as a software teacher and advisor, I see that some of the problems of using computers are the result of a lack of resources– such as hardware, money and time to familiarize. The most common constraint is time, especially since many of the problems associated with effective use of software are linked to the problems of qualatative data analysis itself. In too many projects, the data collection and data preparation phases spread into the time allotted for analysis. Aggravating these difficulties is the lack of formally prescribed techniques lot analysing quantative data as a result of the variety of analytic paradigms and the personal nature of the relationship between data and the researcher. These issues, when added to the need to choose and become familiar with a program, combine to make the researcher’s problems more complex.

What CAQDAS software can do now is to increase the efficiency of your access to data and your flexibility and your preparedness to re-think. It can be used at a very basic level and still he of benefit, and it can increase the number and complexity of tasks which can be performed. Always remember, though, that increasing your choices and tools in this way makes demands on your time.

18.6 CAQDAS RESOURCES

The CAQDAS Networking Project: Offers support to a worldwide research community. Provides platforms for debate, information and training concerning computer assistance in the analysis of qualitative data. Based at University of Surrey.	**CAQDAS Networking Project Online Resources** **Advisory help-lines** a.lewins@soc.surrey.ac.uk Tel: +44 (0) 1483 259455 **World Wide Web page** http://caqdas.soc.surrey.ac.uk Bibliography – Articles – Events – Courses – Seminars – Download demo versions of software. Links to software developer pages Links to other related resources Stop-press news, upgrades etc. **Internet Discussion List – qual-software** To join ‘qual-software’ Internet discussion group send a message to: jiscmail@jiscmail.ac.uk in the text enter: join qual-software Yourfirstname Yoursecondname see message archives below
Mailbase message archives	http://www.jiscmail.ac.uk/
Message archives for qual-software	http://www.jiscmail.ac.uk/lists/qual-software.htm/
Social Science Information Gateway	Multi-discipline information resource http://www.sosig.ac.uk/
Sociological Research Online	Electronic journal: Theoretical, empirical and methodological discussions, political, cultural topics and debates. Free Issues from March 1996–September 1998. http://www.socresonline.org.uk/
Free software	‘CDC EZ-text’ created by Centers for Disease Control and Protection http://www.cdc.gov/hiv/software/ez-text.htm ‘AnSWR’ created by Centers for Disease Control: Information Systems Support Services (CISSS) http://www.cdc.gov/hiv/software/answr.htm

18.7 FURTHER READING

Qualitative data analysis methodologies are systematically described in Miles and Huberman (1994), and an appendix in the same volume summarises a typology of software and a description of the tools supporting such methods of working. Weitzman and Miles (1995) reviews and describes practical processes which are served by a range of software programs. Similarly, Dey (1993) provides a readable, ‘user friendly’ assistance to qualitative data analysis with software use in mind. Books describing the practical processes of substantive project work, oriented to specific software use, can be very helpful. A good example of this is Bazely and Richards (2000) who work systematically through a project using QSR Nvivo.

Murphy et al. (1998) compiled a review of the literature concerning qualitative methodologies (294 pages), for the NHS R&D Health Technology Assessment Programme. The volume is available in hard copy and online at http://www.hta.nhsweb.nhs.uk/. This and other online papers concerning the use of CAQDAS software are accessible via links from the CAQDAS Networking Project web pages, at http://caqdas.soc.surrey.ac,uk/news.htm.