



University of HUDDERSFIELD

University of Huddersfield Repository

Gibbs, Graham R.

Computers and Qualitative Data Analysis

Original Citation

Gibbs, Graham R. (2012) Computers and Qualitative Data Analysis. In: Research Methods and Methodologies in Education. Sage, London, pp. 251-258. ISBN 9780857020383

This version is available at <http://eprints.hud.ac.uk/id/eprint/12858/>

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: E.mailbox@hud.ac.uk.

<http://eprints.hud.ac.uk/>

Software and Qualitative Data Analysis

Graham R. Gibbs

In Arthur, J., Waring, M., Coe, R. & Hedges, L.V. (eds) (2012) *Research Methods and Methodologies in Education*, London: Sage, pp. 251-258
ISBN: 9780857020390 & 9780857020383.

Computer Assisted Qualitative Data Analysis (CAQDAS) programs have been developed to help with the sheer amount of data to be analysed and its complexity and density. There are many different programs available but some are better than others at some kinds of analysis and for some purposes. If you are able to choose what software to use before you start analysis then it is important to know which is good at what. Download trial versions and try them out. (See the resources section at the end of the chapter.) The rest of this chapter, however, focuses on one of the most popular, NVivo, now in version 9, but much of the advice applies to any software you might use. There is a very full interactive help system with the NVivo program. There is a pdf versions of all the help information which can be downloaded from the website of the publishers of NVivo, QSR. Unfortunately, at the moment this is only available for version 8. The functionality of version 9 is similar to version 8, but version 9 now uses a ribbon bar instead of the menus of version 8. References below are to pages in the 'Using the software' manual, for version 8, abbreviated to NVUS.

Whilst there are many benefits to be gained from using CAQDAS, there are dangers too. Fielding and Lee have examined the history of the development of qualitative research and its support by computers in the light of the experience of those interviewed in their study of researchers using CAQDAS (Fielding and Lee, 1998). Amongst the issues they identified was a feeling of being distant from the data. Researchers using paper-based analysis felt they were closer to the words of their respondents or to their field notes than if they used computers. This is probably because many of the early programs did not make it easy to jump back to the data to examine the context of coded or retrieved text. In contrast, recent programs excel at this. A second issue, as many users and some commentators have suggested, was that much software seemed too influenced by grounded theory. This approach has become very popular amongst

both qualitative researchers and software developers. However, as Fielding and Lee point out, as programs have become more sophisticated, they have become less connected to any one analytic approach. A related danger that some have pointed to is the over-emphasis on code and retrieve approaches. Indeed, these are core activities of CAQDAS. Some commentators have suggested that this militates against analysts who wish to use quite different techniques (such as hyperlinking) to analyse their data. But, it is clear that coding is central in the kind of analysis best supported by most CAQDAS and although some software does have linking facilities, these are not as well developed as those that support coding.

To use CAQDAS or not

There are several considerations to examine when deciding whether to use software with your project. You might consider your project too small to justify its use. This does not simply mean a small number of participants. Interviews and observations can be long, complex and intensive and produce lots of data to analyse. On the other hand you may intend to use simple structured questionnaires, in which case the data may be quite easy to analyse without computer support.

Above all you need to ask if the software supports your analytic approach or does it help sufficiently to make its use worthwhile. Code and retrieve or thematic coding approaches are well served by the software. These include grounded theory, interpretative phenomenological analysis, framework analysis, template analysis and many kinds of ethnographic approaches that tend to analyse data thematically. Less well served are more intensive and discursive approaches such as discourse analysis, narrative and conversation analysis. Some software does support the special mark-up needed by e.g. conversation analysis and even supports the process of transcription, but these approaches are not reliant on thematic coding and thus the core function of most CAQDAS is of little use.

Setting up the project in NVivo

Start the program and in the welcome screen that appears select New Project and give it a title and description (NVUS, pp. 42-44, 56-7, 63-64). Use the Browse button to save it where you want it. Then the main NVivo window opens. See Figure 1.

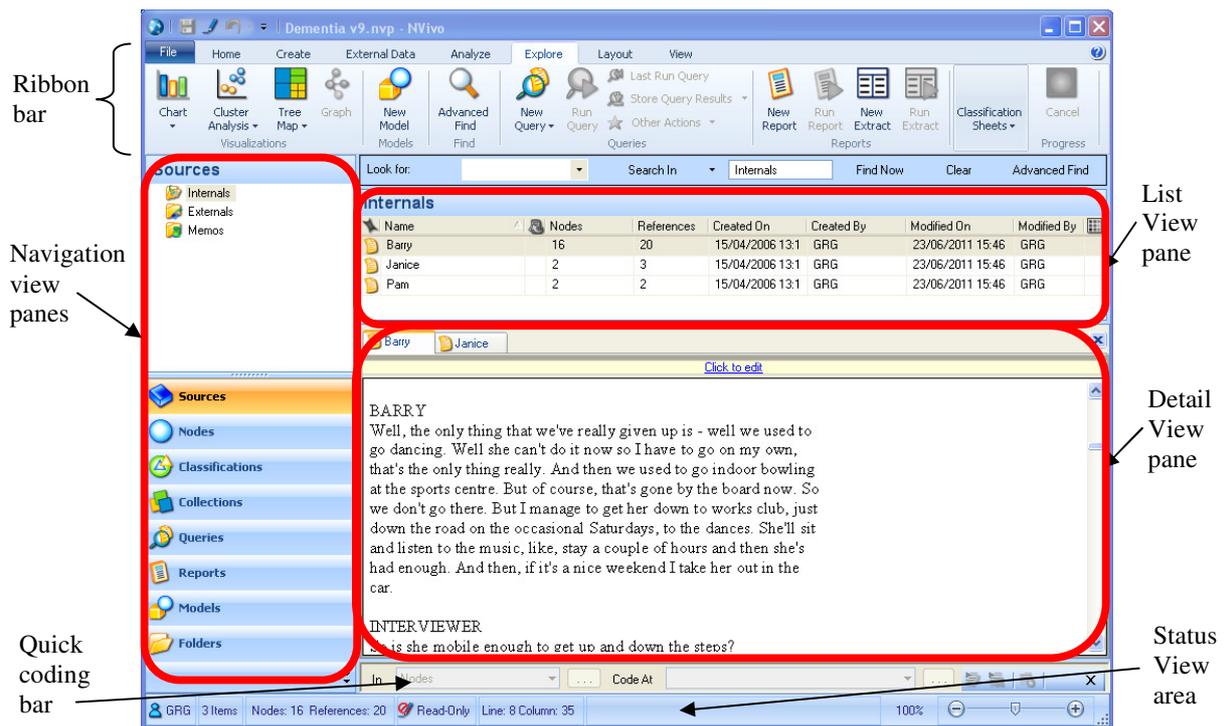


Figure 1 The NVivo main window.

Sources

The new project is just a container for your data. The next step is to introduce some data into it. These data are called sources in NVivo and include documents, video, audio, images as well as a special kind of document called memos. Most commonly the sources are text, including interviews (usually audio or video recordings), observations (including those recorded by video) and field notes. NVivo can import a variety of types of documents including plain text (.txt), rich text format (.rtf) and Word format, both old and new (.doc and .docx) (NVUS, pp. 72-3, 87-95). Once introduced, documents can be examined at any time and their contents appear as tabs in the Detail view pane (NVUS, pp. 82-5). You can also introduce sources that are video, audio or images in a digital format. Many of these will be at a much higher quality than you need in the

project for display on screen. If you can, make the video files smaller by resaving as small mp4 files. For the audio use mp3 files. Images from cameras can be quite large. Again for the purposes of showing on screen these can be resaved at much lower quality and resolution jpegs. You will need to make the changes outside NVivo before you import them into the project. (NVUS, pp. 73-6, 96-102).

Units of analysis

You need to decide what are the cases in your study because in NVivo cases can have attributes and can combine together several different sources (documents, video, images etc.) This analytically separates the sources of information about the cases (e.g. the transcripts of your interviews or your video of interaction in the classroom) from the cases themselves. By the time you start using NVivo you should have a pretty good idea what these cases are. Sometimes they reflect your sampling strategy, for example you may have undertaken snowball sampling of young people who played truant when at school age, in which case people are your cases. More commonly cases reflect your research questions or your research design. For instance you may be investigating differences in classroom innovation between different teachers and different schools. Then your cases will probably be teachers (or classes) and schools. It is usually best to set up these cases when setting up the project, although if more appear or additional values for existing cases are discovered during the study these can be added to the project. In NVivo cases are a kind of node and they are found in the nodes list pane (NVUS, pp. 160, 162-172).

Attributes means variables, usually categorical variables, about the cases. For example, if the cases are people they might be age, gender, education or work experience of the people, or if they are places they might be population, crime rate, state, or if events they might be date, duration, size, type etc (NVUS, pp. 160-3). In a mixed methods study you might well have collected some attribute data of this kind about the cases in the quantitative part of the study. In which case you can introduce this data to the NVivo project as a spreadsheet or an SPSS data file called a casebook (NVUS, pp. 165-72). Normally it is best to introduce this data before any qualitative data as it is a quick way of establishing cases. But you can also add attributes to cases

after you have set up the project, either from data you have collected in the field or on the basis of analysis of your qualitative data (See Miles and Huberman, 1994, 102-9).

Security

As you build your project and develop your analysis, you will create files and structures you won't want to lose. Do regular saves of your data. By default, NVivo prompts every 15 minutes for you to save your data or produce backup files, so that if the program or the computer crashes at any time, you will not lose all your work.

Most of the data you create is very compact. Information about coding and links takes only a little space. Along with the documents you have introduced, this information is all kept together in the project file. However, you may opt to keep larger files you may have, such as video and audio files, outside the project file. You should keep backups of both the project file (for NVivo this is the .nvp file) and other large data items that you are keeping outside the NVivo project file. You don't normally change primary data such as audio and video after it has been introduced into the project so these may only need occasional backups. But the main project file will change every time you do some analysis so this needs regular backing up. The most convenient back up media are flash memory – usually in the form of memory sticks – and removable hard disks. Memory sticks will be fine if you just have documents and a few pictures, but if you have lots of video or audio files then a removable hard disk will be needed.

Also keep the data confidential. Don't allow others to see your data if you have given respondents assurances that what they have told you is confidential or will be kept anonymous. Password protect your computer and your NVivo project if you can and don't leave the program running unattended on a PC in an open office.

Coding

Coding is one of the core activities in most CAQDAS programs and NVivo is no exception. Coding means applying labels, the codes, to passages of text, sections of video or audio, or regions of images. Most commonly this is done as a way of indicating all the content that is about some theme you have identified. At its simplest, this labelling or coding process enables researchers quickly to retrieve and collect together all the text and other data that they have associated with some thematic idea so that they can be examined together and different cases can be compared.

NVivo calls codes 'nodes'. Typically when you first create nodes they can just be kept in a list. As you develop your coding ideas you can arrange the nodes into a hierarchy or tree shown in the node list view pane. (NVUS, pp. 137-40). Nodes can be re-ordered, renamed, deleted, split and combined and nodes in a hierarchy can be reorganised by moving them from one branch of the tree to another. (NVUS, pp. 145-52)

You can separate out the construction of a coding scheme (the nodes) from the act of coding the text and other sources. Thus you can create nodes in NVivo, perhaps with definitions and even attached memos, without any coding, i.e. without using them to label any sources. Such nodes, often called *a priori* codes, are based on your literature review, your experience in the field or your initial hunches and can be used later for coding. On the other hand a very common approach and one often combined with *a priori* codes is to read the text (or view the video) directly and create new nodes and code the source content to them and/or to *a priori* codes as you do so (NVUS, pp. 182-92, 197-200).

Once some sources have been coded, you can inspect what you have done in a couple of ways. With text that has been coded you can display coding stripes, to the right of the text, to show how the passages have been coded (NVUS, pp. 202-5). See Figure 2. Alternatively you can retrieve all the sources coded at a specific node. Just double-click the node name (in the node list

pane) and a new tab will open in the detail view pane showing the text and other sources that have been coded at that node (NVUS, pp. 153-8).

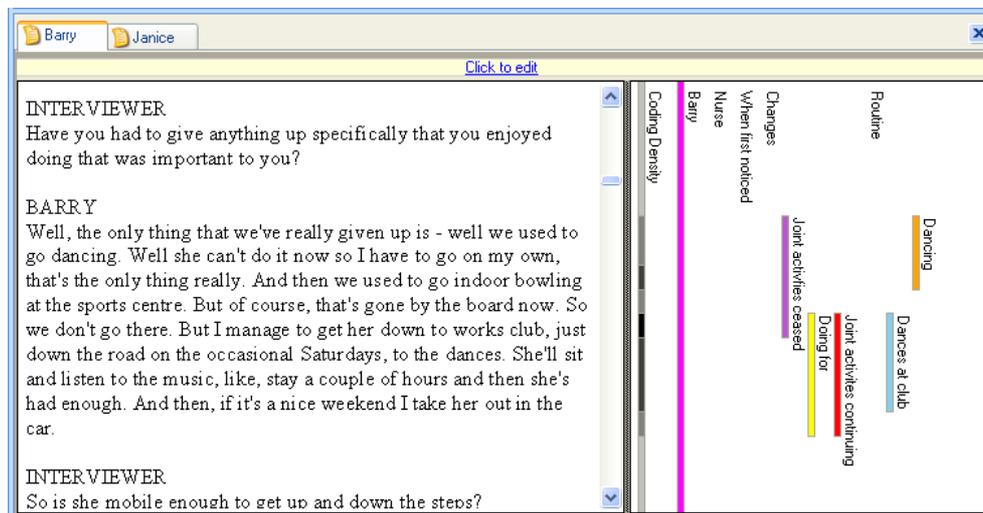


Figure 2 Document showing coding stripes

This code and retrieve activity is a central one in the analytic process. First, it enables you to check that the theme you have identified makes sense, is well evidenced by the sources you have coded and that the coding has been done consistently. It is thus a way of checking the quality of your analysis. Second, you can begin to look for patterns within the data sources coded to the same theme. For example, you can compare the results across different cases to see whether there are differences in what has been coded between groups of cases – perhaps all the older cases talk about this thematic issue in a different way from the younger ones. In this way you can build up a more sophisticated account of what is happening amongst the cases in your study.

Coding Crisis

A common problem experienced by many researchers doing qualitative analysis and especially those using software to help them is to end up with too many codes and/or disorganised codes. This is not necessarily a bad thing. It may simply reflect the heterogeneity of the data you are analysing and the complexity of your analysis. However, it can be a barrier to further analytic

work and especially to developing a clear understanding and explanation of your data. There are several things you can do. For example, you could print out all the codes you have (possibly with definitions and even short samples of the coded text) cut them up and then try rearranging them. Alternatively you might cut and paste such details into a spreadsheet where you can use rows for each code and columns for things like definitions and examples and for other thoughts about the code. In both cases you are looking to re-arrange and sort the codes or sometimes to re-express them. You should look for patterns and categories amongst the codes. One reason some researchers like to move away from the software to do this is that you are attempting to rethink the analysis you are doing and moving away from the existing project stops it interfering in the rethink. Other things that might help with this rethink are re-examining the literature for ideas or talking about your research to colleagues or your supervisor.

Some heuristics that are useful here have been suggested by supporters of grounded theory: for example, the idea of dimensions of codes (Corbin and Strauss, 2008). This is the notion that codes can be of different kinds of thing or refer to different contexts etc. of the same thing. It is quite common in the early, open stages of coding to note down all the different varieties of things without noticing that they have anything in common. Thus you might have codes for note taking, summarizing, writing prompt cards, re-reading and quizzing as ways of revising for exams. In which case there can be an overall code called 'Revising' and the others can be simply ways of doing that. In NVivo this can be done easily using the hierarchical coding system (the node tree) to make 'Revising' a parent node and 'note taking', 'summarizing' etc. its children nodes (NVUS, pp. 145-52).

In other cases you may find that nodes are actually about the same thing but you didn't notice you had two of them when you were coding (perhaps they relate to different cases). In which case they can simply be combined. On the other hand when looking carefully at both the definition of a node and the text that has been coded at it you might realise that there are two (or more) distinct analytic things represented by the coded items. In which case you can split the node by creating a new one and moving some of the coded items to the new node (and removing them from the old node).

Searching

CAQDAS programs, NVivo included, support two kinds of searching, searching for text, also called lexical searching, and searching for codes. Both can be used as ways of advancing the analytic process. In NVivo they are both types of query. Lexical searching is rather like the word searching facility in a word processor, only more powerful. In NVivo you can search for a number of different terms (perhaps synonyms) at the same time, search for word roots (and find all the words with different endings) and even search for words spelt like the terms you are using (NVUS, pp. 242-250). After the search NVivo codes all the terms it has found and you can then display these finds and look at each, in turn, in its context.

Lexical searching can help the analysis in a number of ways. First, it can be used a way of getting familiar with the text (in addition to reading the text, of course). Search for terms that are connected with your theoretical hunches and then inspect the passages where the terms are found in the original documents. This might produce new ideas or candidates for new nodes. Second, such searching can be used as a way of looking for passages similar to those you have already coded. Passages already coded will contain terms, words or phrases that might occur elsewhere and indicate similar topic matter. Put these terms and others you can think of into the text search tool to find all the further occurrences. Of course, this won't guarantee that you find all relevant passages but it can complement your reading of the data. Beware, sometimes the search will find passages that contain the term but just aren't relevant. You need to read each in turn and decide about their relevance. Third, you can use this approach as a way of checking the validity of your analysis and in particular you can check for the occurrence of negative cases – that is, instances that are inconsistent with your explanations. You may have missed these because you just weren't expecting them in the context they appear. But if they use the same terms as other instances then lexical searching will find them. However, beware, this approach is not infallible. Relevant passages of text might just not use the terms you are searching for and so you won't find them. In the end you still need to read the text and inspect the other sources in a comprehensive way.

Searching for coded text and attributes

It is quite common, even in published work, for researchers just to summarise the major thematic codes in the report on their study. This expresses what they have found and, naturally tends to be quite descriptive. Sometimes that is interesting, but qualitative studies can go a lot further and offer accounts of the patterns of the occurrence of such themes and, perhaps, suggest causes for those patterns. In CAQDAS programs it is searching for coded text and searching for attributes (often combined together) that supports that. In this case what is compared in the search is the actual text coded at or linked to the node or attribute. Thus in the simplest case, if you search for one node or another, what is compared is the text coded with these nodes. The search will find all the text coded at either node, if any (including that coded at both nodes, if any).

NVivo allows two or more nodes (and sometimes attributes too) to be searched for in combination (NVUS, pp. 233-7, 250-8). Such combination is divided into two kinds, Boolean and proximity. Boolean searches combine codes using the logical terms like 'and', 'or' and 'not'. Proximity searches rely on the coded text being near, after or perhaps overlapping some other coded text. Commonly used proximity searches are 'followed by' (also referred to as 'sequence' or 'preceding') and 'near' (also referred to as 'co-occurrence') (NVUS, pp. 263-6). Boolean searches are most useful in examining hypotheses or ideas about the data and rely on consistent and accurate coding, whereas proximity searches can be used more speculatively and to explore the data, often at an early stage of coding.

For instance in a study of teachers' career development, you might wonder if male teachers had a different view from female teachers about career development courses. Assuming you had a node for 'career development courses' with lots of coding done and an attribute for gender, then you could search for text coded at the node 'career development courses' AND in cases with the attribute female, and then repeat it for the male cases and compare the two sets of data you have retrieved.

Teamwork

Using computers to assist with analysis is particularly useful when working in teams on projects. Partly this is because teams tend to work on larger projects, with more respondents and more settings so the database is simply larger. However, teams need to be properly coordinated and can undertake certain kinds of cross checking and the software can be of crucial help here.

When working in a team, it makes sense, very early on in the project to create a model template for everyone to work on. This can be done in NVivo using the Project Properties (NVUS, pp. 51-6). Here you can set up all the users who are entitled to use and work on the project. You should password protect the project and decide whether each researcher has read only access to the data or whether they can edit the data as well. Of course it makes sense to have a convention for how you name sources (documents, videos etc.) and cases and how you organise them in the project. One person, perhaps the senior researcher (or the NVivo expert) can set up the cases and import the documents available so far and even establish some preliminary codes in the project along with associated definitions and linked memos about them. This project can then be shared with others in the team and those who you decide may do so can add more sources, coding and cases as appropriate.

When several people are working on the analysis then consistency in approach becomes an issue. This may be down to simple things like how large are the chunks of text that get coded. You might decide that a minimum size might be a sentence, or that where possible and appropriate several sentences or even whole paragraphs or speeches should be coded. In other cases, it is possible to assess the reliability of coding by comparing one coder's work with that of another (NVUS, pp. 267-8).

Resources

The **CAQDAS Networking Project** provides practical support, training and information in the use of a range of software programs designed to assist qualitative data analysis and has links to all the manufacturers' websites.

(caqdas.soc.surrey.ac.uk)

QSR is the publisher of NVivo and on their website you will find pdf versions of the help files. For version 8 these are: 'NVivo 8 Help – Using the software' and 'NVivo 8 Help – Working with your data' and they can be found by searching for the titles in the Resource Articles section. These documents are not available for version 9 yet, but you will find the video tutorials for version 9 very helpful. They can be found under Tutorial: NVivo 9.

www.qsrinternational.com

Suggested Further reading

N.B. All the books mentioned here cover older version of the software. However, their advice and much of the detailed instructions will still apply to the most recent versions.

Lewins, A. and Silver, C. (2007) *Using Software in Qualitative Research: A Step-by-Step Guide*. London: Sage.

Written by two experts from the CAQDAS Networking Project, this covers in detail the three most popular programs; NVivo, MAXQDA and Atlas.ti, as well as some discussion of other programs. There is good advice on how to choose which program is right for your analysis and how to set up and use the programs for your project. A new edition, covering the latest versions of the software is due out soon.

di Gregorio, S. and Davidson, J. (2008) *Qualitative Research Design for Software Users*. Maidenhead: Open University Press, McGraw-Hill.

Focuses on the issues around the ways that your research design will influence and be influenced by the use of software. In particular it contains sage advice about things to consider when first setting up your data in a new computer project.

Bazeley, P. (2007) *Qualitative Data Analysis with NVivo*. London: Sage.

This works through all the stages of undertaking an analysis using one program, NVivo. There are detailed instructions on how to use the software at each step of your research.

Gibbs, G.R. (2007) *Analyzing Qualitative Data*. London: Sage.

This is a more general book on qualitative analysis, but the last few chapter provided step-by-step instructions on how to get started in your analysis using either NVivo, MAXQDA or Atlas.ti.

References

Corbin, J.M. and Strauss, A.L. (2008) *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Thousand Oaks, CA: Sage.

Fielding, N.G. and Lee, R.M. (1998) *Computer Analysis and Qualitative Research*. London: Sage.

Miles, M.B. and Huberman, A.M. (1994) *Qualitative data analysis: a sourcebook of new methods*. Beverly Hills, CA: Sage.

Questions for further investigation

1. Can software help with your project's analysis? Which software will support what you need to do and which do you have access to?
2. How will you set up your data set in the software program you are using? What are the main groups of documents? What are the cases? What initial, a priori codes will you use?
3. How can you use the software to ensure that your analysis is of high quality and any conclusions you draw from it are justified?
4. If you are working in a team, how will you organise the analytic work? Will you share the coding and, if so, how will you compare and combine the coding you have done?