Choosing a CAQDAS Package
A working paper by Ann Lewins & Christina Silver

Glossary:

CAQDAS - Computer Assisted Qualitative Data Analysis
Qualitative data analysis – see below
Content analysis – quantitative analysis of the occurrence of words, language
Interactive – good, instant hyper linking (usually one click or double click) between an object e.g. a code in one pane and e.g. and highlighted source context in another (*KWIC* is an aspect of interactivity)
Invivo Coding – in the context of software, a short-cut tool for using a word or short phrase in the data, as a code label
Autocoding or Text search - a search for words or phrases in the data files
Hits – the initial finds in the data which result from the above
KWIC - Retrieval of key words in context

It is not always easy to visualise exactly what a CAQDAS package offers when exploring it for the first time yourself. Equally, when asking someone else for their opinion, it is not always easy to know which questions you should be asking. Most of the software packages we are aware of and discuss regularly are excellent products in one way or several! Sometimes you choose the package that is already *in situ* and make good use of it – but if you have a choice about which software to purchase for your research project, you may be in some uncertainty about how to proceed. You may have basic understanding about what a CAQDAS software package will do, but the differences between the tools offered in each package are subtly but significantly different.

What does this paper do…?

- This paper offers a **summary of types of software for managing textual or qualitative data as a starting point for thinking about which one may be most suited to the type of project and data** you are working with and the way you like to work.
- It provides more detailed information focused mainly on those traditionally categorised as "Code-based Theory Building" software packages (see below). We cannot provide an exhaustive description or comparison of all available CAQDAS software here, but we aim to highlight some of their key distinguishing elements in order to provide you with an early impression of each package.
    - Firstly we provide a description of the tools that these software packages have in common.
    - Secondly we provide information of some of the distinctive features of as many packages as we can. (This will be added to as we complete more software reviews and hear more opinions)
- It aims to address the most frequently asked questions that we at the CAQDAS Networking Project receive.
- It aims to assist you in your search for more information by **strongly recommending that you also visit the Software Developer websites as part of your decision-making process**, where you can access and download demonstration versions.
- It aims to review both commercially available software and open access or free software.

In order to help you with making this decision the CAQDAS Networking Project provide a series of (FREE) Software Planning Seminars [http://caqdas.soc.surrey.ac.uk/softwareplanning.htm] where we raise some of the issues you may need to consider, and discuss how some of the software programs differ in the way they handle such aspects. We also provide an email and telephone helpline [caqdas@soc.surrey.ac.uk +44 (0)1483 689455] if you require more specific advice in choosing and using CAQDAS software.

What types of software do we categorise as *CAQDAS?*

Software which falls under the CAQDAS 'umbrella' includes a wide range of packages but its general principles are concerned with taking a qualitative approach to qualitative data. A qualitative approach is one where there is a need to interpret data through the identification and possibly coding of themes, concepts, processes, contexts, etc., in order to build explanations or theories or to test or enlarge on a theory. Many approaches (including, for example, participant observation, action research, grounded theory, conversation analysis etc.) broadly make use of qualitative perspectives, though each may emphasise particular processes or differ in their sequence. Qualitative researchers may also use quantitative data, for instance, relating to the descriptive features of research participants – sex, age and so on – to help in the organisation of qualitative data. This assists in cross-referencing and comparing qualitative ideas across groups and subsets. This approach remains distinct from 'content analysis' methodology, in which it is the statistics of word or phrase frequencies and their occurrence relative to other words or phrases across a textual dataset that are the basis of the analytic work. Although you will see later in this page that we refer to packages that deal with the content analysis approach in varying degrees, we only describe such packages in detail where they also incorporate a strong qualitative dimension.

What ranges and types of software support work with qualitative data?

We refer in part to a typology suggested by Miles and Weitzman (1995)[1] concerning the handling and management of qualitative (and at that time, mostly textual) data. We make some reference to this typology, because it is still current in some respects, but increasingly with the advances of individual software described in this paper, the distinctions for instance between software formerly labelled Code and Retrieve and the more extensive functionality in Code-based Theory Builders have become blurred. Now very few software programs remain in the Code and Retrieve category. Similarly, some of the Code-based Theory Building software have taken on features more traditionally featured in Text retrievers, or Text based managers – e.g. *Content analysis* tools, Word frequencies, word indexing with *Key Word in Context retrieval* (KWIC) and complex text based searching tools. Conversely, by expanding to provide different *add-on* software to enable thematic coding, Text retrievers and Text based managers are beginning to enable tasks originally only possible in Code-based Theory Building software.

Code-based Theory building software

Both these and the earlier Code and Retrieve packages assist the researcher in managing the analysis of qualitative data, to apply thematic coding to chunks of data, thereby enabling the reduction of data along thematic lines (retrieval), limited searching tools and probably good memoing facilities. Code-based theory building software packages build on those tools and extend the collection of *search* tools allowing the researcher to test relationships between issues, concepts, themes, to e.g. develop broader or higher order categories, or at the other extreme, to develop more detailed specific codes where certain conditions combine in the data. Some of the programs enable the graphic visualisation of connections and processes using mapping tools.

Text Retrievers, Textbase Managers

Tools provided in both these categories often provide more quantitative and 'actual content' based analysis of textual data. There are complex and sophisticated ways to search for text and language including the use of thesaurus tools to find words with similar meaning, to index all words contained in the text, to provide word frequency tables, to create active word lists, to provide easy key word/phrase in context retrieval (KWIC). However, broadly summarising them in this fashion masks the huge variety of differing ways in which they each handle the quantitative analysis of content, and the importance given to particular aspects in each software. Generally speaking, some automatic 'content' analysis often happens just as part of the process of importing data.

Some Textbase Managers have very sophisticated 'content analysis' functions; creation of keyword co-occurrence matrices across cases, creation of proximity plots for identification of related keywords, charting and graph building facilities etc. Textbase Managers tend to offer more functionality than Text Retrievers and more possibilities to manage huge datasets in varied ways, but it must be said again, that all these categories are becoming blurred, and we link to developer sites and software within these two categories without making any distinction between them.

[1] Weitzman, E. & Miles, M.; (1995) **A Software Source Book: Computer Programs for Qualitative Data Analysis** Thousand Oaks, Sage Publications.

The theory building category itself is blurred by the increasing addition of some quantitative language based content analysis features (MAXqda, with add-on software MAXdictio with good KWIC), and to a lesser extent ATLAS.ti.

One or two of the Textbase Manager packages now incorporate 'thematic' qualitative coding functions which can be integrated with the range of content analysis tools e.g. the language based quantitative functions. So they offer a comprehensive range of both qualitative and quantitative approaches to data within one software, e.g. QDA Miner with the add-on Wordstat module, and CISAID). Where this is the case, we will try to provide information on the particular attributes of such software program's tools (from a qualitative standpoint) and will continue to add more comment and comparative information to this online resource as we undertake a review each software.

Which is the 'best' CAQDAS Package…?

This is perhaps the most frequently asked question we receive – however, it is impossible to answer! All the packages we work with and teach have their own range of tools to support you in the various stages of the analytic process. As such, they each have their own advantages and disadvantages.

Whilst we would argue that **some software packages may be more suited to certain types of approach, their purpose is not to provide you with a methodological or analytic framework**. The tools available may support certain tasks differently and there is some debate as to whether a particular package may 'steer' the way you do the analysis . **However, as the researcher you should remain in control of the interpretive process and decide which of the available tools within a software can facilitate *your approach* to analysis most effectively**. Whichever package you choose, you will usually be able to utilise a selection of tools which will help significantly with data management and analytic tasks .

Thinking about and using CAQDAS software should not necessarily be any different from other types of software package – just because a tool or function is available to you does not mean you will need or have to use it. You therefore need to think clearly about what it is you are looking to the software to help you with and we would caution against choosing a package simply because it seems the most sophisticated. Conversely, and this may have something to do with your longer term plans, you may feel a more ambitious selection of tools will serve you better over time.

The Basic Functionality of CAQDAS Software :
Key Similarities between Packages

Whilst there are many differences between CAQDAS packages, the key principles behind each of them in terms of facilitating the qualitative research process are similar in many ways; (the ways in which they differ are described in the software specific sections later on).

Structure of work

The 'project' that the user creates in the software acts as a container or a connector to all the different data files within the current working project. Internal databases, *contain* the individual data files – i.e. when you *import* data it is a *copy* process. External databases *connect to* the data files – i.e. you *assign* data to the project, but the data remains in its original location. In either case, the opening of one project file enables immediate access to all components of the dataset. There are varying benefits and disadvantages to either system. Internal databases are easier to move around from computer to computer, external databases may cope better with very large datasets and are more likely to directly handle a range of multimedia files.

'Closeness to data' and interactivity

At the most basic level, the packages discussed here provide almost instantaneous access to all source data files once introduced into the 'project'. Whatever tools you subsequently use, 'live' contact to source data is always easy, which increases the researcher's closeness to data.

Explore the data

Text search tools offer ways to search for one word or phrase, or even a collection of words around the same topic area. Such searches usually provide good access to those parts of the documents where those keywords appear – allowing a fairly instant retrieval of topic related material – sometimes abbreviated as KWIC (Key Words in Context).

Code and Retrieve Functionality

They all offer code and retrieve functionalities. User-defined key-words and/or conceptual categories (codes) can be applied to selections of (overlapping, embedded) text and as many codes as required can be applied to the same text selection. Usually, the user has complete freedom concerning the structure of the coding schema and the coding strategies employed – e.g. whether inductive, deductive or a combination of both.

In general terms code generation is both easy and flexible and the researcher is free to use a combination of strategies if desired and to refine coding if necessary. In all but one software reviewed here, the responsibility for thinking about each coding action rests entirely on the user. In Qualrus, however, the software (using *Artifical Intelligence*) learns from previous actions and makes suggestions.

In all packages, coded data can be retrieved, re-coded and outputted with ease. Software differ in the way coded information is provided and visible in the data file itself, for instance in the margin area. While this can be produced by most of the software reviewed here, it may be the ease with which such a view can be generated, the flexibility of what can be shown and the interactivity of the margin area which present key differences.

Project Management and Data Organisation

All these packages also offer powerful means by which to manage the research project as a whole and to organise data according to known 'facts', descriptive features and data types. Any files which can be converted into the format(s) supported by the given software constitute data as far as the software is concerned.

The project management elements mean that **these packages are not simply tools to facilitate the analytic stage of the research process**. For example, much work can be done before data is introduced to the software and as such, *theory-building* CAQDAS packages both reflect and significantly facilitate the 'cyclical' nature which is characteristic of many qualitative research and analysis processes. Data organisation enables the researcher to focus on (combinations of) sub-sets of data, thereby facilitating comparison. Even when used in the most basic way, therefore, CAQDAS software packages all significantly increase the researcher's access to the different elements of their project and sub-sets of data.

Searching and interrogating the database

At any stage of the process all the packages offer means by which to interrogate the dataset. This includes searching the content of data – based on language used – as well as searching for relationships between codes as they have been applied to data (for example, co-occurrence, proximity etc.). Search tools also allow you to combine the coding (interpretive or conceptual) and organisational (descriptive) dimensions of your work.

Writing tools

The process of qualitative data analysis is rarely linear and the various writing tools (for example memoing, commenting, annotating etc..) offered by CAQDAS packages provide ways to increase the researcher's contact with his/her thoughts and processes – provided, of course, they are used in a systematic way. Add notes from paper

Output

All reviewed software packages have a fairly standard selection of reports (output) which allow the user to view material in hard copy or integrate into other applications, e.g. Word, Excel, SPSS. Those standard reports will usually include coded segments either from one code or a selection of codes. The latter option is not always as straightforward as it seems but when it happens, it is accompanied by varying amounts of identifying information (source document identifier and the coded segments themselves is usually the minimum, and sometimes but not always, the relevant code label is included above the segment).

Tabular output: usually simple tabular output is available providing breakdown of code frequencies which can be exported to Word, Excel or SPSS. Programs vary in the types of tables which can be generated. Output also varies in

terms of the level of its interactivity with live source data. Results of searches can be viewed often in both output format or inside the software, integrated into the working project.

When the software supports the use of mapping nor more graphic representations of coding schema etc., these can usually be exported and pasted or inserted in Word files.

NOTE: The fine distinctions in the ways that output can be varied by the choices the user makes in each software, are not reviewed comprehensively, though we point out particularly distinctive or interactive forms of output.

Summary

The combination of these key aspects of CAQDAS packages mean that the researcher is able to keep in reliable and continuous contact with the different elements of the project and the analytic processes. When used systematically and within the individual researchers' 'comfort zone', CAQDAS packages can aid continuity, and increase 'transparency' and methodological rigour.

Deciding on which is the 'best' CAQDAS software package is necessarily a subjective judgement and will probably be based on reaching a balance between a number of factors.

Some general questions to ask when choosing a CAQDAS package

- What kind(s) and amount of data do you have, and how do you want to handle it?
- What is your preferred style of working?
- What is your theoretical approach to analysis and how well developed is it at the outset?
- Do you have a well defined methodology?
- Do you want a simple to use software which will mainly help you manage *your* thinking and *thematic* coding?
- Are you more concerned with the *language*, the terminology used in the data, the comparison and occurrence of words and phrases across cases or between different variables ?
- Do you wish to consider tools which offer suggestions for coding, using *Artificial Intelligence* devices?
- Do you want both thematic and quantitative content information from the data?
- Do you want a multiplicity of tools (not quite so simple) enabling many ways of handling and interrogating data?
- How much time do you have to 'learn' the software? …
- ….How much analysis time has been built into the project?
- Are you working individually on the project or as part of a team?
- Is this just one phase of a larger project – do you already have quantitative data?
- Is there a package – and peer support – already available at your institution or place of work?

Coming soon : summaries of selected *caqdas packages*
see http://caqdas.soc.surrey.ac.uk/

Coming first… Atlas.ti, HyperRESEARCH, MAXqda (& MAXdictio), N6, NVivo, Qualrus, QDA Miner

…also look out for summaries of freely available software – AnSWR, InfoRapid, Transana
…and non code-and-retrieve based software – Storyspace

Summaries of some Theory-Building CAQDAS Software Packages – in alphabetical order

ATLAS.ti V5, HyperRESEARCH V.2.06, MAXqda, N6, NVivo, Qualrus

This section summarises some CAQDAS packages, but its only focus is the distinctive tools provided by each software program in the above or additional categories. Note that it does not seek to summarise the entire functionality of each software. We remind you that basic functionality available in all quoted software is listed in the above sections. For each software we have divided our work into main headings for ease of comparison:

- Minimum Specifications
- Structure of work – how your work and data is managed by the software
- Data types and format
- Closeness to data and interactivity – (e.g. hyper connectivity between coding & source text)
- Coding schema – coding structures and organisation
- Coding Processes
- Basic Retrieval of coded data
- Searching and Interrogating the database – distinctive tools
- Teamwork
- Going beyond code and retrieve – various functions enabling other ways to work

We strongly recommend that you visit individual software sites and experiment and explore demonstration versions of the software before making a final decision.

ATLAS.ti (Version 5): distinguishing features and functions

ATLAS.ti was initially developed at the Free University, Berlin as a collaborative project between the Psychology department and Thomas Muhr (a computer scientist), as an exercise to support Grounded Theory. Subsequently Thomas Muhr's independent company, Scientific Software Development, continues to develop and support the software. http://www.atlasti.de

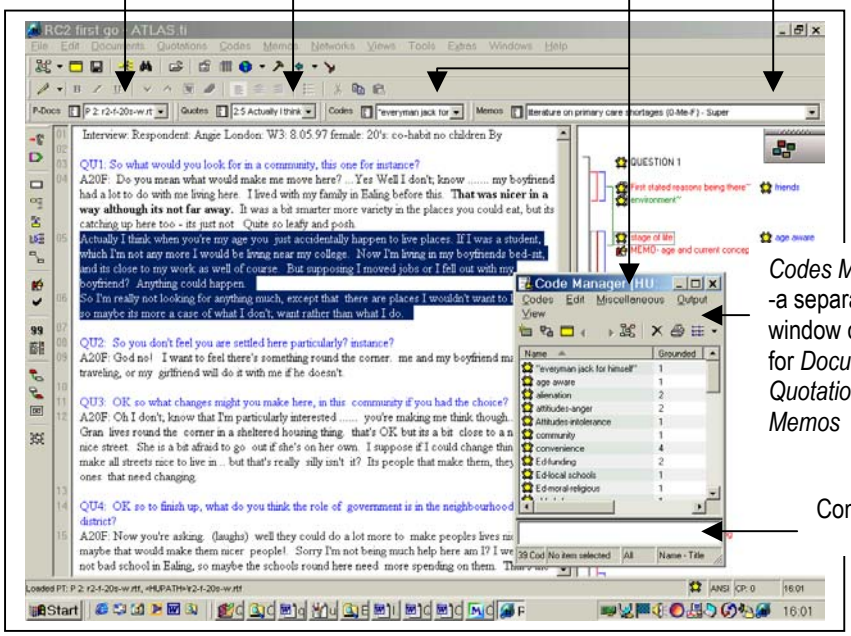
Minimum System Specifications (recommended by developer)

MS Windows 98 or later (W2000 or XP recommended)
RAM 64Mb (minimum), 256Mb (recommended)
25 Mb free disk space (minimum), 45Mb (recommended)

Structure of work in ATLAS.ti V5

ATLAS.ti functions using an external database structure which means data files are not copied into the 'project', but are assigned and then referred to by the software when the project is open.
Two main elements of the researchers work in ATLAS.ti consist of:
1. The Hermeneutic Unit (HU) in which only the records of assigned documents, quotation/segment positions, codes and memos are contained, and because of the external database, separately held are….
2. …the data files, e.g. transcripts, multimedia data etc

The main functions of the software operate from (always open in a typical view) main menus, a large selection of icons, and the main PD (Primary Document) pane, in which the qualitative data file currently selected is on display. In addition are four main 'manager' panes – which allow Documents, Quotations, Codes, and Memos to be created, assigned, and managed.



Codes Manager window - -a separate *Manager* window can be opened for *Documents, Quotations Codes* and *Memos*

Data types and format V5

Textual data: Version 5 of ATLAS.ti handles either rich text format (.rtf) data files (which can be marked, edited and annotated at any stage and other objects (such as graphics, tables etc.) can be embedded into them), or Word files (which cannot be edited after assignment to the HU). Coding can be assigned to any chunk of text however small.
Multimedia data: digital video, sound and graphic files can be directly assigned to the project and coded using similar processes as are used to code text. Some of the multimedia formats include: .jpg, jpeg, .gif, .bmp .wav, AVI Video, .mpg .mpeg, .mp3. See developer information for complete list.

Closeness to data – interactivity in ATLAS.ti V5

The central functionality of ATLAS.ti is connected to the independence of *quotations* (or interesting segments of text) that can just be selected and listed in the quotations list. They do not have to be coded. You can navigate around the list of significant quotations (*free quotations*) that you make, and the quotations are then highlighted within their source context, or, in the case of multimedia, clips of the sound or video file are played back. In their listing, individual quotations can be renamed to be more descriptive of each video clip. Similarly the researcher is kept 'close' to data throughout the analytic process – for example, when navigating around coded quotations in the data (whether across the whole database or within sub-sets of it) the original source context is always immediately visible around the coded segment. The margin view allows the researcher to gain an overview of how a data file has been coded, but it is also fully interactive – for example, highlighting a code in the margin will display the exact segment of text which has been coded; codes and memos can be edited from the margin. Similar interactivity exists within the network (map) views.

Coding schema in ATLAS.ti V5

In the first instance, the coding schema is, in its main listing and structure, non-hierarchical; this is intended to free the user who wants instead to be able to create semantic or meaningful links between codes, going beyond any structural hierarchy. Codes can be initially created in network (map) view e.g. to display a theoretical model.
Hierarchical or semantic links can be created to express relationships between concepts, themes etc. These links can be seen in Network views or the Object Explorer – but not in the main codes list/manager windows.
Families of codes: any code can be put in any number of families/collections, allowing the user to *filter* to view or get output on just a family – or to create *Superfamiles* from combinations of families. Families of codes can represent small or large collections of codes representing theory or process or similar types of theme, (*Attitudes, Outcomes*, etc). A code can belong in any number of families.
Margin display: fully interactive with edit functions. The user can decide which objects to have on display in the margin, i.e. Codes or Memos or Hyperlinks or all three.
Colour of codes in margin display: the user has no control over colours used in the margin – and these will vary according to size of segment or its relationship to large/smaller segments at that point.

Coding Processes in ATLAS.ti V5

Codes can remain abstract/free of text or can be dragged and dropped on to highlighted text or marked clips/quotations in the multimedia file either from the Codes Manager window or from the codes area in the Object Explorer.
Several iconised shortcuts can be utilised; coding to multiple codes at a time, *in vivo* coding etc
Codes and memos can be edited from the margin, i.e. change boundaries of quotation (selection of text) coded, name of code, merge codes, unlink codes from quotations.

Basic Retrieval of coded data in ATLAS.ti V5

In ATLAS.ti there are several ways to retrieve data. Selecting a code and either using navigation keys or double clicking allows the user to see in turn each coded segment highlighted (or played back) within the whole source context, whether the segment is textual or multimedia. Alternatively coded data can be lifted out of the context by asking for output e.g. on a selected code or a *family* or collection of codes or all codes. Segments are clearly labelled with source file name, paragraph number, code label.

Organisation of data in ATLAS.ti V5

Documents can be organised into subsets by putting them into *Document Families.*
Tabular data in the correct format can be imported to quickly organise whole files into *Families* e.g. based on Socio demographic characteristics or the user can do so in a step-by-step way in one user friendly window.

Writing tools in ATLAS.ti V5

ATLAS.ti allows the user to write in many different places.
Memo's: Memos are a main *object* / tool in ATLAS.ti and form the central place to write, and have listed memo topics. Memos can be *free* or linked to points in the text. They can be categorised or put in collections, and filtered or exported. Memo's can be linked to each other in the network / mapping tools and they can be linked to codes.
Comments: In addition each object manager in the software, Documents, Quotations, Codes, Networks, Families, all have *Comment* bars in which the user can write ad hoc notes to describe an item.

Searching and Interrogating the database in ATLAS.ti V5

The *Query tool*: The Query Tool allows the searching of coded data based on a number of standard Boolean, semantic and proximity based operators.
The Supercode tool: this is particularly useful allowing the researcher to save the way a search/query has been built in order to re-run later in the process. Having been created, the Supercode in ATLAS.ti is listed as a special code in the code list, enabling *up to date* navigation around the coded results of complex searches in the same way as happens for normal codes, thus enabling the testing of an early test or hypothesis later in the analytic process. Super codes can be also combined with other codes in the formulation of an even more complex 'query'.
Auto-coding - Text searching: The usual tools are available but autocoding of finds or hits can be controlled as the search is done; the user can enlarge each find to include sentences or more text as the software finds and displays each hit in its full context (KWIC). Or the autocoding search can be done routinely without confirmation, enlarging each coded passage to the sentence, the lower level paragraph (separated by one hard return), or collections of these paragraphs, finally ended by two or more hard returns, or the whole file.
Co-occurring codes in network view: This is so far, an exclusive tool to ATLAS.ti 5: it allows the user to select a code in a network and see *co-occurring codes*. This will put any other code which overlaps or coincides in the text, with the original selected code.

Output in ATLAS.ti V5

Normal coded output distinguished by labelling of each segment with code, source document, paragraph numbers. Output can be varied to produce summaries; lists of codes and how they are connected to other codes.
Tables which display frequency of codes by documents can be filtered to show comparisons between document subsets and coding subsets (families). Both varieties can be exported e.g. into Word, or Excel or printed. Multiple codes and their quotations/segments can be exported into one file in a clearly labelled format . Networks (maps) can be exported to Word, either via clipboard or via generation into graphic file formats. The whole HU can be exported into an SPSS file. The export of the whole HU to an HTML file allows navigation around the project by the non expert user of ATLAS.ti .
Word frequency tables can be exported to Excel or run in the software (not interactfively connected to source context)
XML export: A Hermeneutic Unit exported to XML representation can be converted into a wide variety of other representations by using stylesheets. Individual reports, conversions into other programs' input formats or creating 'clickable' viewer versions of your HUs are among the options available. This relates to teamwork in that the XML export allows team members or interested non users of ATLAS.ti to navigate around XML generated reports from a web page.

Team-working in ATLAS.ti V5

Teams working separately can now use shared data files from one shared drive, although they cannot yet work on the same HU at the same time. If the HUs/projects themselves need to be merged a flexible Merge tool allows different models of merging HU's.. The Merge too allows the user to select from several options to create a tailor made Merge – e.g. to merge 'same codes or different codes', 'same data or different data files'.

Ability to go beyond Code and Retrieve in ATLAS.ti V5

Quotations

You do not need to code when looking at interesting data in ATLAS.ti because a segment of text or *quotation* can be selected and viewed in the list of quotations just because it is significant. You do not have to decide to *code* the data in order to get back to or retrieve the significant quotations. Quotations are a central and independent feature of the architecture of ATLAS.ti software - a distinctive feature compared with other code-based software.

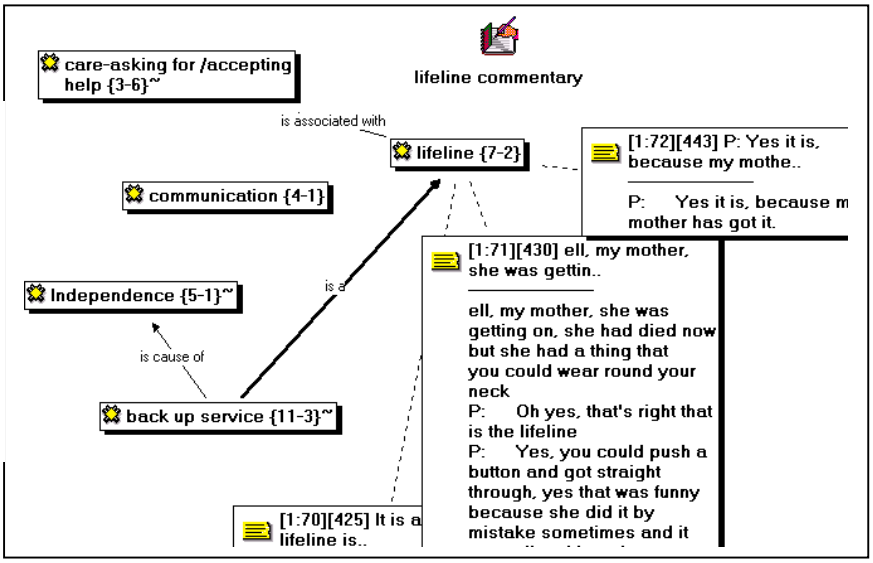
Hyperlinks between quotes – alternative to coding

The centrality of the quotation structure enables effective hyperlinking between quotations/passages of text – rather than being dependent on coding as the mechanism for following a theme. This allows you to track (and subsequently jump between) places in the data in the sequence you want in order to tell or explain e.g. a story or a process more effectively. You can additionally express relationships between quotes (software defined relationships e.g *explains, justifies, contradicts* or user defined connections e.g. *leads to, denies, reacts)*.

Mapping using the *Networking* tool

The Networking tool is flexible, allowing graphic, meaningful links to be created between quotations, codes, documents and memos. *Hyperlinks* between quotes can be created in Networks (as above). Codes can be linked together to express connections in a visual way, whilst certain relational links created between codes function as semantic search operators in the *Query* tool, retrieving e.g. everything connected to a code via a series of uni-directional transitive links. Any number of networks can be created, but links between any codes are meaningful, in that wherever two or more of the same linked codes appear in any network view they will be linked in the same way. Links can be reversed, broken, and changed in appearance.

Network containing codes, quotations and memos



Content analysis - word frequency

This tool counts the occurrence of words across the whole dataset, or a single file and the results can be saved into an exported spread sheet file or into a memo. The list of words is not interactive – i.e. no KWIK (but see later Searching tools for KWIC)

Object crawler

The object crawler allows the user to search for the use of strings, (keywords?) phrase etc., in the entire HU/project – in data, codes, memo's networks, comments.

CAQDAS Networking Project Comment on ATLAS.ti:

- *Atlas.ti V 5 offers great flexibility and provides several different ways of working to suit different purposes – the software goes far beyond code and retrieve as a method of analysis, yet if that is all the user wants it is very easy to get to that point. The flexibility provided by the quotation structure and the ability to hyperlink between places in the data is useful but if used extensively the user has to come up with an effective way of managing these linkages, and finding start points of such trails in the data.*

- *The external database makes the process of saving and moving the HU and data slightly more difficult to manage for users. The single most asked about issue concerns users who have moved their HU file without also moving their data. Version 5 and its altered Copy Bundle function in the Tools menu makes this process easier.*

- *Related to this issue, users who edit data files within the HU, (now possible in ATLAS.ti 5), will need to take care in the management and diligent saving of changes to the individual data file. This issue has improved but difficulties if encountered may relate to the synchronisation of changes between data files and the HU (the project).*

- *Though it is possible to create code connections and collections of a hierarchical nature, the main working code list (Codes manager) does not have a functioning hierarchical structure to choose from. To some users this is a main attraction, to others it lacks choice since hierarchical code structures often provide an easy way to systematically 'tidy up' a codes' list.. Of course once the user is much more familiar with the software, making hierarchies or collections of codes or linked codes are possible, using various tools, e.g. Families, Networks and Supercodes.*

- *The Supercodes function in the Query tool could be an excellent a way to place marker questions, or hypotheses, in order to test ideas and connections between themes giving the easiest way possible to re-run simple to complex searches. The presence of all supercodes in the codes list is a constant reminder of previously interesting ideas – with the ability to double click on them in the codes list –re-running the original query on updated data.*

- *The Query tool itself is easy to use for simple searches but can be daunting because codes have to be chosen before the search operator; some search operators have very precise parameters and the user must be aware of these to read results and their implications reliably.*

- *The Co-occurring codes in the network function (see above) is unique to ATLAS.ti and one its best improvements in version 5. No other software provides a simple tool which allows the user to choose any code and find out all other codes which overlap or co-occur with it in any way (in the data). This tool is particularly useful for identifying other more complex questions or trends in the data.*

- *The Network tool is flexible but if the user makes connections between codes too early in the analytic process it may be difficult to stay aware of links that have been made which are becoming less generalisable as more data becomes coded (unless for instance the links have been codes made for an abstract a priori theoretical model). Links between quotations on the other hand are made in situ for a specific situation or interaction.*

- *The Object crawler provides an overview of the project as a whole and would help for instance in the recovery of notes that the user knows he has made but cannot remember where (within an HU)*

- *Export HU (the project) to html is the simplest way to allow the non-user in a team to navigate and 'click" around a project and see its 'workings'. This facility is extremely useful in team situations and the presentation is user-friendly.*

HyperRESEARCH: distinguishing features and functions

HyperResearch is a software program developed by ResearchWare Inc.
Contact address for information is support@researchware.com . Web site: http://www.researchware.com
In its latest version 2.6 it is both available for MAC format and Windows. It supports a full range of coding, annotation and searching tools for textual and multimedia data. It has a fairly unique structure compared to other software since its main focus and default unit of analysis is the Case.

Minimum System Specifications (recommended by developer)

MS Windows 95 or later, Mac OS8 or later
RAM 64Mb
5 Mb disk space

Structure of work in HyperRESEARCH

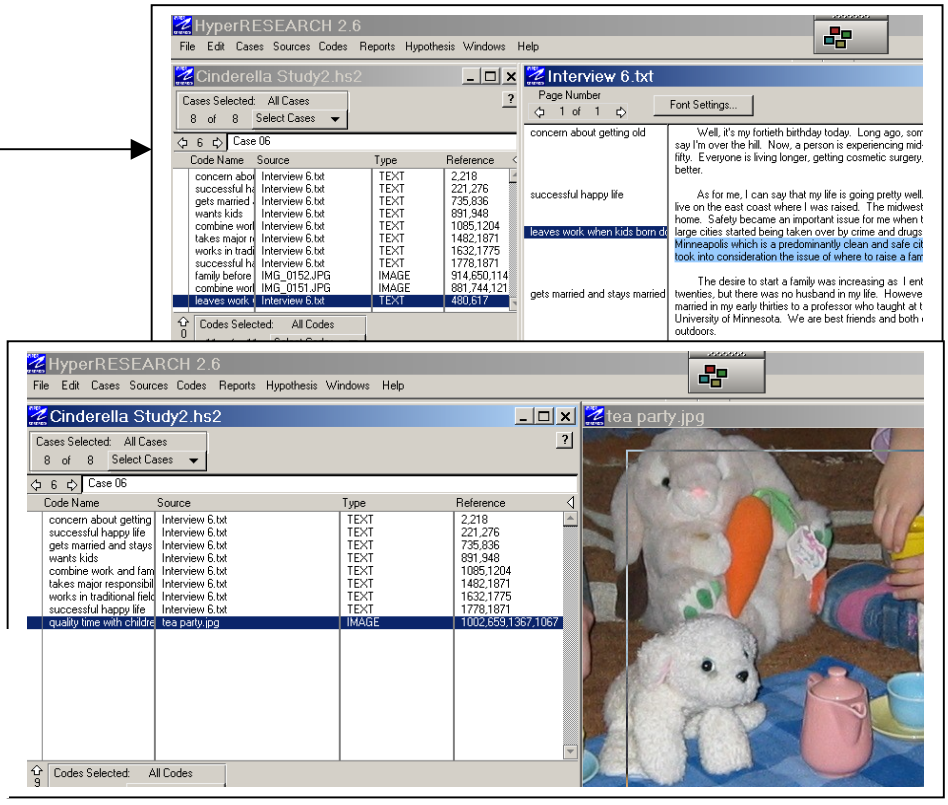
The project as a whole, is called a Study. The database is external – so the source files are accessed by the Study, not contained within it. The Study is comprised of Cases, and Cases can have references to one source data file or many. The Coded references are listed in the Case card for each case, and the coded references are hyper-linked to the source file. Case cards are interactively linked to their relevant references in the source files. See below.

1. The user moves between Cases, not files

2. Coding is listed in *case cards*

4. Retrieval and navigation is by Cases

5. A case and its case card can refer to one file or parts of many files



A coded reference interactively connected to the coded part of a jpeg image file

Data types and formats in HyperRESEARCH

Both textual and many forms of multimedia data can be directly coded in HyperRESEARCH
Textual data can be transcribed in any Word processing application, but needs to be saved as 'Text only' before it is opened as a Source file in HyperResearch. Coding can be assigned to any chunk of text however small.
In the text source file window customize the font settings (typeface and size)
Multimedia files are in format necessary to save or condense them in Windows / MAC ,
e.g. .jpg, . jpeg / JPEG, JPG, .gif / GIFf , . bmp / BMP, .wav / WAVE, .avi / .mov / MooV. .mpg .mpeg / MPEG, .mp3 / MPG3. See developer information for complete list.

Closeness to data – interactivity in HyperRESEARCH

There is no contact with source files from within the software until the user starts to code data and therefore coded references to data files start appearing on the *Case cards*. After that there is good contact with the whole context of the individual data files because annotations, case card coded references, and coded segments in reports are all hyper-linked (one click) to respective source file position, whether the file is textual or multimedia (e.g. the coded video clip will replay as the reference is selected). The coded margin display is interactive, but since there are no brackets showing the approximate size of coded segments, the user selects the relevant code in the margin to see where exactly it has been applied. Reports containing references to coded passages are also hyperlinked back to source file positions.

Coding Schema in HyperRESEARCH

The coding schema is not hierarchical. The codes list is alphabetically organised. There are no easy ways to create hierarchical collections of codes except in the code map (one code map is enabled– see Searching and Interrogating for more information on how code map works)

Coding processes in HyperRESEARCH

Whilst the correct case card plus source file is selected (on view), codes are assigned to selected text (any number to same or overlapping chunks) by menu selection or by double clicking on the code from the codes list. The code appears in the margin of the file and is interactively (one click) connected to the passage at that point. One click on the code will highlight coded text. Codes can be applied to areas/parts within a graphic file, or clips of a sound/video file. The coding reference also appears in the *Case Card* of the case that is selected. See screen shot above.

Basic Retrieval of coded data in HyperRESEARCH

Since the case structure of HyperRESEARCH is so dominant you might choose to first navigate around cases based on the presence or absence of codes – so cases on view can be determined by selection of codes.

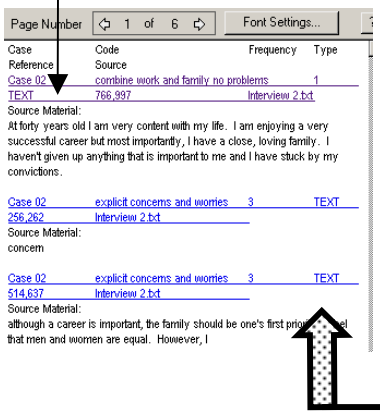
Codes in context: navigate between case cards based on:

Selection of codes: by name, or by criteria e.g. a code's presence or absence in a case, or its inclusion with other codes in a case. Hyperlink (one click) from case cards to coded references in textual or multimedia files.

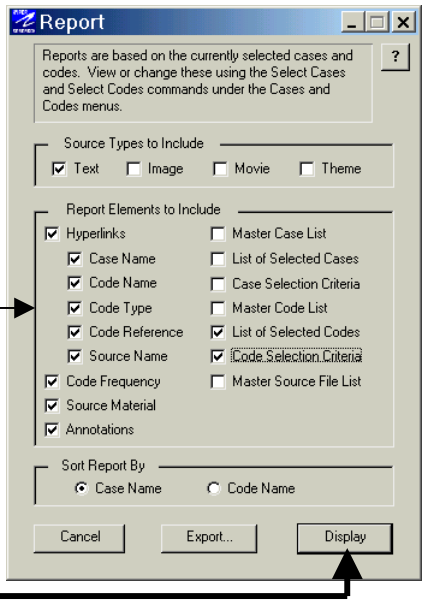
Selection of Cases on the basis of various simple to complex criteria (on basis of name or inclusion, exclusion, overlapping codes etc.).

Main output of coded retrieval using various criteria happens in reports (above):

Hyperlinks to source text included in Reports at each listed segment (one click) to source text.



Report building dialogue box – allows *user to select various aspects for inclusion.*



Organisation of data in HyperRESEARCH

Basic organisation can happen at the level of the case, but it is also possible to categorise whole or parts of files by coding.

Writing tools in HyperRESEARCH

Annotations: you can write comments attached to coded references on each case card – the annotations are flagged in a column next to the coded reference – and are hyper linked (one click) to both the text of the annotation and the source file position. These can be included in reports on coded data etc.

Searching /Interrogating the dataset in HyperRESEARCH

Selection of cases or codes: 'Searching' happens in the software just by a process of filtering or selecting of codes or cases. Every time the user selects codes – by name, by criteria (inclusion, overlaps etc) or by codemap etc – the case cards reflect the selection – since only those references asked for will appear. Every time the user selects cases, on the basis of name, code, or criteria, the cases which do not show the criteria are not present as the user browses the cases or makes a report. Searching is very useful and flexible when combined with *reporting* the results. See also *Reports* section, under *Coding and Retrieval* above.

Hypothesis Tool: Works similarly to the selection of codes or cases, but allows the user to formalise the searches. You can test the existence or co-existence of codes in cases to do one of two things:

1. To add higher concept *Themes* to whole cases
2. To build rules in order to progressively test elements of a whole hypothesis. Hypotheses can be saved and re-run on newly coded data – or used as the basis for new, edited versions of the test – or for completely different studies.

Themes applied to cases do not act like new codes in terms of hyperlinkages to source text, but they e.g. act as filters to allow navigation through Cases in which they DO appear or DO NOT appear and can act as criteria for building retrieval, reports, or hypotheses.

Searching by *applying* code map: the code map view allows the user to create connections between codes – you can then select one code and instruct the software to select all codes within 1 or 2 etc connectivity (i.e.1= immediately connected to selected code or 2= connected at one remove). This then shows user which cases, if any, have any of the codes defined by the *Apply* operation.

Autocoding: Searching tool allows user to search for words or phrase and code results –with added context coded if wished, to include required numbers of words or characters before and after hits. User must proactively tell software which files, within which cases, to search, settings can be saved and reloaded (and amended) for future autocode searches.

Output in HyperRESEARCH

See reports above (in Coded retrieval section) –

Report parameters can be saved and loaded again. The information saved includes the original code and case selections, as well as the information you wished included in the report

Code Matrix: *Code frequency across Cases can be exported to e.g. Excel*

Code Map can be exported into Word

Teamworking in HyperRESEARCH

There is no straightforward way to merge different projects, though the developers can help with this if it is required.

Ability to go beyond code and retrieve in HyperRESEARCH

Code-mapping tool

Use the Code Map Window to graphically represent relationships between your codes. Any operation in map has to be preceded by selection of appropriate function button (see also Searching and Interrogating data section)

CAQDAS Networking Project Comment on HyperRESEARCH:

- *The software is cross platform, for both MAC and Microsoft Windows/XP users and therefore belongs to a very small body of software for qualitative data analysis specifically written and kept up to date for the MAC user.*

- *The software is simple to use.*

- *The unit of the analysis is the 'case' not the File, this may appeal to some users. Any number of files (or just one) can be referenced to a case (or to any number of cases. This means files are not hard wired to cases so that when in coding mode the user must always be aware what case is open so that the coding for a file goes to the correct case card.*

- *The hypothesis tester and the 'Themes' which can then be assigned to cases as a result of testing, provide an understandable and clearly visible way of categorising cases by higher concepts. The fact that further hypothesis tests can include these 'themes' as criteria for selection underlines the importance of understanding how at this stage, the case increasingly becomes the unit of analysis (though of course you may only have one file in each case)*

- *Interactivity and hyperlinks between case cards, reports, annotations is very good. One click or at most a double click provides access to the whole file with relevant data highlighted.*

- *The report builder – and the subsequent reports are unusual and very useful. The user has complete control about how much information is included in the report. In the report itself, the coded segments are clearly labelled and hyper- linked to source text (one click) whether the actual text of the segment is included in the report or whether its just a listed reference to the segment.*

- *The code map is rather cumbersome to use, but acts for a different purpose to other mapping tools mentioned. It mainly acts as a filtering /searching device. Its utility to graphically describe a multiplicity of relationships is limited since the user must make changes to the one code map permitted in a study*

- *The software has no backdrop/background so the interface does not obscure other non minimised applications. This can be a nuisance.*

- *There is no simple way to 'pack up' all the elements of the 'Study' (i.e. the Study file and also all the source files)if for instance you wish to move the files to another computer.*

MAXqda – distinguishing features and functions and MAXdictio (add-on module)

MAX, then WinMAX Pro, and now MAXqda www.maxqda.de are part of a software stream originally authored by Udo Kuckartz in order to cope with the management and analysis of political discourse data on the environment. Its application has widened to multiple academic disciplines and applied research areas. Optional add-on software MAXdictio (extra cost) expands its functionality to provide basic content analysis style indexing and counting of all words in the dataset. Contact for information: info@maxqda.de

Minimum System Specifications (recommended by developer)

MS Windows 98 or higher
RAM: 64MB (minimum)

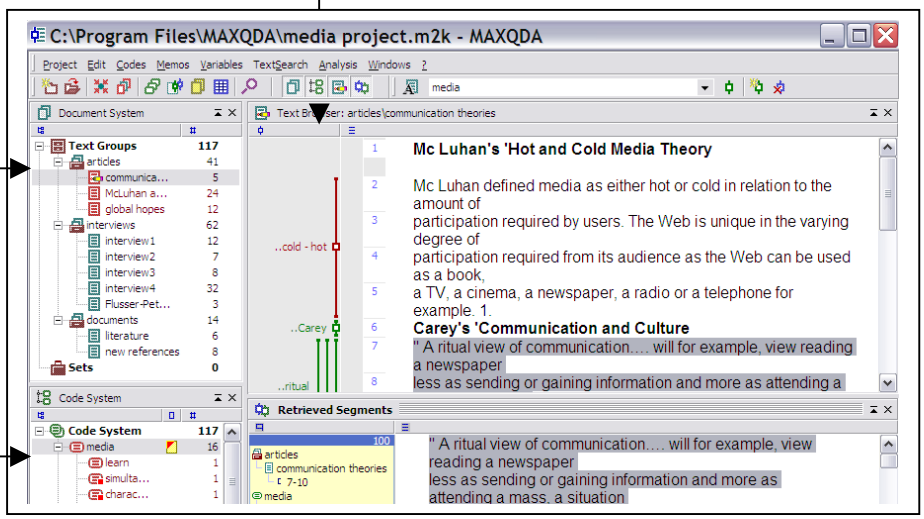
The Structure of work in MAXqda

MAXqda's internal database means that data files (documents), once imported are contained within the *Project*, and are moved or saved as part of the project. .

Interactive margin (colour codable codes)

User-defined *Text groups* act as organising containers for data, into which individual data files be imported or in which blank files can be created

Hierarchical coding schema



Retrieved segments based on *activated* codes and *activated* text files –each segment interactively linked to source file top right

Data types and format in MAXqda

Textual data: files have to be saved in *Rich Text Format* –no other format of data is allowed.
Rich Text Format allows full editing rights on the data once in MAXqda. Objects, text boxes and graphics should not be contained in the Rich Text file. Any amount of text can be selected for coding purposes.

Closeness to data - Interactivity in MAXqda

Interactivity from Coded segments in the Retrieved segments pane back to source data file, between codes in margin and source data, mean that the user is very close to whole context of the source files. Interactive frequency tables (varied by activation of different files), results of text searches are all interactively connected to source data – (one click). The compact user interface with 4 main windows enhances the users interactive contact with the whole dataset.

Coding schema in MAXqda

The **coding schema** can be as hierarchical or as un-hierarchical as required. **Drag and drop** allows easy re-organisation of coding schema and hierarchical structure.
Colour of codes in margin display: User can assign codes a specific colour (from a limited selection of colours) to emphasise particular aspects of coding better in the margin display
Activated coded segments can be printed or saved as a file for export into e.g. Word.

Coding Processes in MAXqda

Any selection of text can be dragged on to a code (to assign code to text) or use coding bar above text window to assign recently used codes.
Undo recent coding actions at coding bar above text window.
Coding margin display: Codes then appear in the margin in *Text browser* window to allow user interactive clicking on code in margin to highlight relevant text. Text browser window with margin can also be printed. Colour attributes assigned to certain codes appear in margin display.
Weight: Coded segments (for certain types of codes) can be assigned weighting within a range of 1 to 100. A weight of 100 is assumed unless otherwise assigned.
Code frequency table interactively shows frequency of codes across whole data set, and also lists frequencies just *activated* files, to enable quick comparison across different subsets of data as they are activated.

.

Basic Retrieval of coded data in MAXqda

***Activation*:** this is the central retrieval principle of MAXqda software allowing the user to control the way retrieval of selected codes for selected files appears. Put simply, the user *activates* the code required, then activates the files on which retrieval is required. Segments can be activated by *Text groups* or S*ets* of files (see below). This principle of activation and its effect when combined with 4 main functioning panes below is what makes MAXqda easy to grasp and manipulate.
Activate by weight: activation(retrieval) of a e.g. a code based on a particular weight range which has been assigned to segments
Retrieved coded segments appear with interactive source data file information allowing (1 click) hyperlink connection from segment to its in-context position in data file. See lower right pane in above screen shot.

Organisation of data in MAXqda

Variables: The assignment of e.g. socio demographic variables and values to data files or parts of data files to allow the interrogation of subsets of the data is possible in a step-by-step or by importing tabular information.
Sets of documents: drag data files into new *Sets* (involving the creation of shortcuts to original files) means that the user can *activate* or switch on files on the basis of sets - useful for e.g. code retrieval, for frequency tables, or for retrieval of all memo's written about any of the Set's documents.

Writing Tools in MAXqda

Attach memos to areas in text – then flagged up and opened in margin display
Link memo's to topics (codes) to enable overview and easy collection, listing and export of all notes written any where in project about that topic. Print or export any memo and its content, or a collection of memos.
Retrieval of memos into collections: e.g. all memos linked to a particular code, all memo's within one document, all memos for a *Set* of documents, all memo's for a text group – can all be printed or saved as a file

Searching and interrogating the dataset in MAXqda

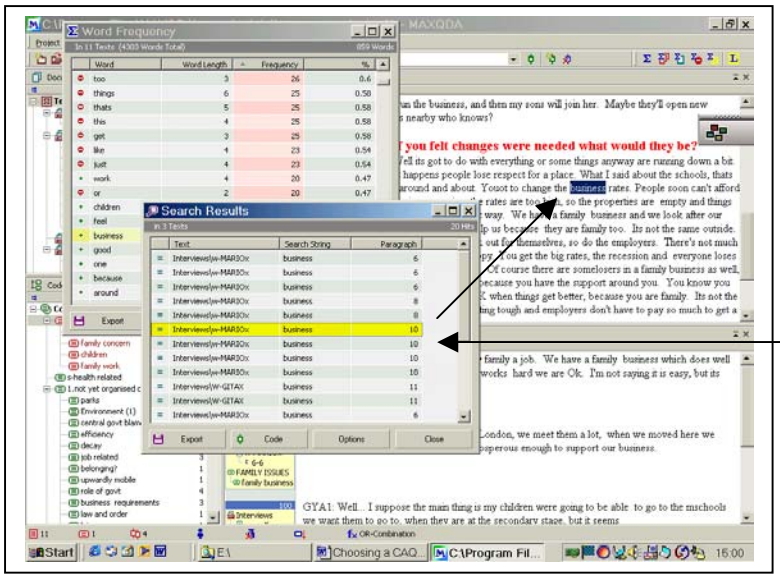
Interrogation of the coding schema happens by simple to complex states of activation and there are several more complex search operators or (*activation* using *'logic'*) e.g. *followed by, near.*
Using weight filter inserts an extra dimension on any search performed.
Autocoding /Text search: a range of words or expressions can be searched for. The list that is produced is interactively connected to source data. The minimum unit of text saved (if the finds are coded) is always the whole paragraph around each *hit.* The search expression can be saved and reloaded.

Output in MAXqda

Any of the 4 main panes of the user interface van (and their contents) can be printed or exported into file format (Text manager pane, Text Browser pane (with codes in margin), Codes list or Retrieved segments pane.
Memos can be outputted in one file to be opened in Wordpad or Word (either all memos, or all memos linked to one data file or a set of files, or all memos linked to one code).
Tabular data: any interactive table produced inside the software can also be exported into Excel or dbf. format

Ability to go beyond *Code and Retrieve*

MAXdictio (add on module to MAXqda – cost is extra)
Functions which are added:
Word Frequency across all files
Produces an interactive frequency index of all words. User can choose any word from this list to produce further index occurrences of the word, each hyper linked to positions in source data (0ne click) see screen shot below
Word frequency across 'activated' (or selected) files
Create 'stop' list: allows user to build list to exclude certain words in count – i.e. and, but, is, etc.
(or this can be done by double clicking on any word in the Frequency list)
Create dictionary: allows user to build list of active words s/he is interested in
- a further button allows use of this dictionary to govern functioning of other buttons



Teamworking in MAXqda

Maxqda comes with various tools allowing various types of merging. Individual documents can be imported together with their coding from other team members' projects, or a whole project's database can be imported into another.

CAQDAS Networking Project Comment on MAXqda:

- *The MAXqda user interface is very appealing and tidy with auto arranging but resizable windows allowing easy customisation and focusing on particular element(s) of work. The compact user interface with 4 main windows repeatedly enhances the user's interactive contact with the whole context of source data.*

- *Having said that, the user interface may feel rather cramped with larger datasets, especially for low resolution screens – e.g. 800x600 pixels would be too low*

- *It is an intuitive and simple software, and easy to get to grips with - perhaps particularly good to teach students*

- *Very good memoing and writing tools – easy systematic retrieval options which may be particularly useful in team situations*

- *In the team situation, users can be selective about which items they merge together into one larger project to be able to compare or continue working cumulatively.*

- *Interactive and colour margin display of coding, which prints out very easily and satisfactorily.*

- *The simple activation of single or multiple codes (OR) or intersecting (AND) multiple codes is easy but the more complex proximity search tools are more difficult to make use of*

- *The autocoding tools are perhaps less flexible than in other software because the only units of context to which you can auto code to are the search string itself or the paragraph*

- *With addition of Maxdictio the software has a small but useful range of content analysis, word frequency tools not currently available in other code based packages – because they provide KWIC retrieval.*

QSR N6 – some distinguishing features and functions

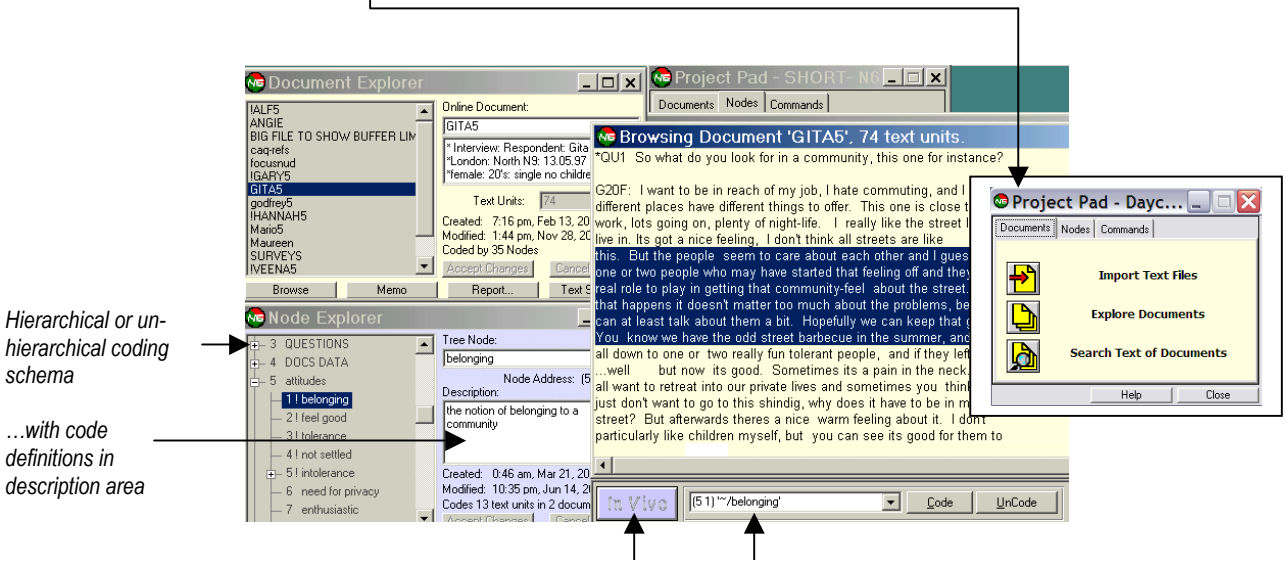
N6 is the latest product from Qualitative Solutions Research (QSR) in the NUD*IST range of software, (Non-numerical Unstructured Data : Indexing, Searching and Theorizing). It was originally developed at La Trobe University, Melbourne, Australia by Tom Richards to support social research by Lyn Richards. Subsequently QSR International was formed which now develops and supports all QSR software: N6, NVivo, Merge for NVivo and XSight. http://www.qsrinternational.com/

Minimum System Specifications (recommended by developer)

MS Windows Me, 2000, XP
RAM : for Me 64Mb, for 2000 and XP 128Mb
15 Mb disk space required (plus space for project data files)

Structure of Work in N6

N6 functions with an internal database which means that all individual data files are imported into the 'Project' and so wherever the Project is placed on your computer the data files will be inside it. There are two main elements to the database – the Document Explorer (where all the datafiles (called documents) are held) and the Node Explorer (where the coding schema is housed). The Project Pad provides access to main functions and has 3 tabs – Documents, Nodes and Commands



In Vivo coding and Coding Bar

Data types and format in N6

N6 directly handles plain text (.txt) data files only – although internal references to other external data formats are possible. The researcher has a choice of the minimum unit of text to which to apply codes – line, sentence, paragraph. This then becomes the *text unit.* Editing of text is possible but only on one text unit at a time.

Closeness to data – interactivity in N6

Documents can be viewed and annotated before any coding takes place. Viewing coded data lifts out coded segments from their source position, though the user can return to the relevant position in the source file (2 clicks away). There is no margin view showing how a whole document has been coded within the software (although an output report can be generated (see below)

Coding schema in N6

The N6 coding schema has a hierarchical and non-hierarchical area allowing the researcher to work in a combination of ways. However, although codes can be moved around the coding schema by *cutting* and *attaching*, there is no drag and drop option for the re-organisation of the coding schema.

Coding is facilitated by a coding bar at the base of the screen allowing the very quick creation and application of new codes to selected text units. In addition an ***In Vivo*** coding tool provides easy generation of new codes based on words or phrases in the data – although it is up to the user to note which document the code originated from.

***Quick coder* tool** supports the researcher who prefers to code on paper – and subsequently input codes within N6 without having to go through the source data again.

Basic Retrieval of coded data in N6

Making a ***code stripes report*:** allows user to select up to only 26 codes to view in a document's margin (or a code/node report)
Retrieval of coded data: Make a report or *browse* all data coded at a topic – by lifting it out of the original source data files, spread the context / jump to the source data (2 clicks) or continue to code (*code-on* / *re-code*) and create new categories.
Export coding schema to mapping package – Inspiration or Decision Explorer to manipulate connections between themes and issues.

Organisation of data in N6

The organisation of data happens at the coding level within N6 and can be achieved semi-automatically by importing tabular information. This can be useful where large amounts of data is being analysed and/or when quantitative (descriptive) information concerning interview respondents, for example, is already held in spreadsheet format.

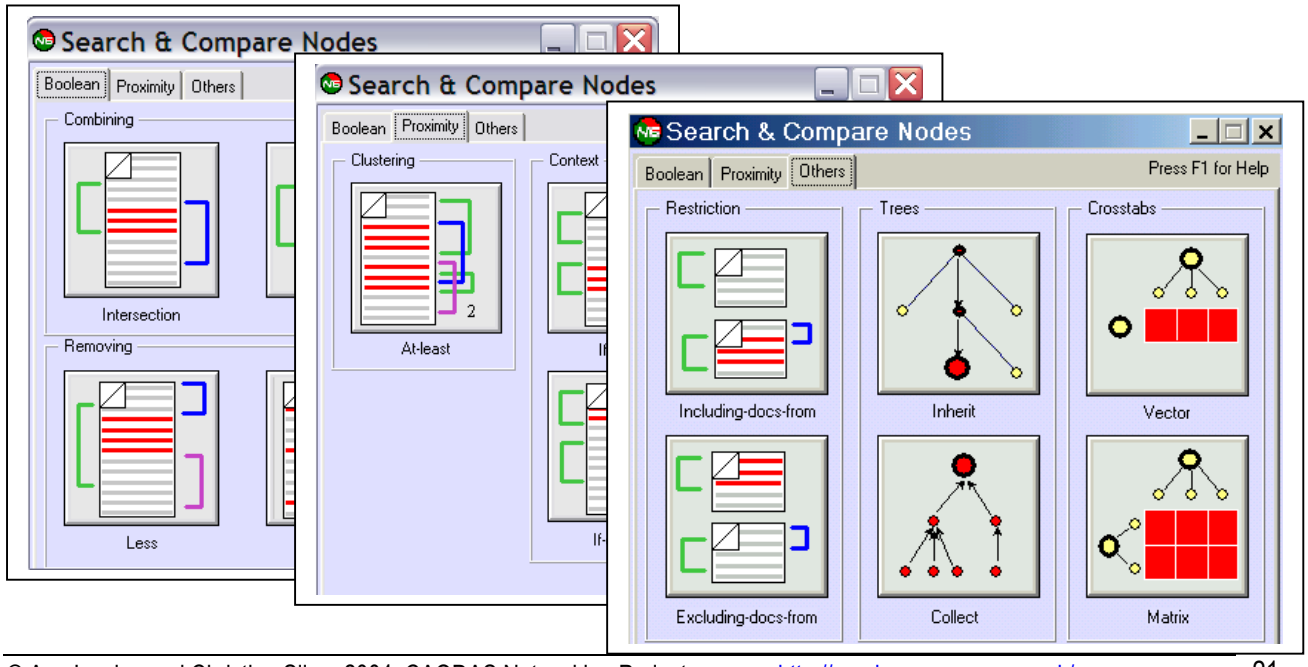
Writing Tools in N6

One memo can be created for each document, and for each node/code. ***Annotations*** which can be coded can also be inserted in the text occupying a new text unit, but this alters text unit numbering. All annotations can then be *browsed,* or combined in searches with thematic codes. Text units can also be edited (one at a time) so this also can be a method of embedding notes in the data.

Searching and Interrogating

N6 includes a range of sophisticated search tools; graphic descriptions of search operators, included in the user interface, are useful when learning the software. The results of any type of search are automatically saved as a new code – thereby easily facilitating the process of asking further questions based on the results of previous findings.

The way a search is performed can also be saved as a command file for re-running on different or accumulated data.
Qualitative cross tabulations – interactive tables: Unique to QSR software at present, is the ability to carryout qualitative cross-tabulations (matrix searches), which may be useful resources at the far-end of the research process. Interactive tables which summarise with a variable count of the results give access to the relevant qualitative data from each cell of the table.



Text search / autocoding tools can be flexibly applied e.g. to all datafiles or data coded in a particular way – or on those data NOT coded in that way. These can be performed individually or multiple searches can be combined in command tools. The hits from these searches can be saved with surrounding context – either the text unit (maybe a line / sentence /paragraph) or to a paragraph (whatever the text unit) or the section.

The ability to go beyond *Code and retrieve* in N6

Automation and Command files

A number of data processing tasks can be automated using command files, and an easy to use and intuitive Command Assistant facilitates the process of writing and building the command files. Command files can automate a range of (repetitive) clerical tasks and scripts written in the command file language can be saved in order, for example, to be re-run on new waves of data.

Teamworking in N6

N6 software is supplied with its own Merge software allowing different team members to work independently on their own parts of a project, with the aim of merging projects into one project at a later or successive points. The merge must be planned for, and coding schema's and datasets can be merged using various models described by the Merge help notes/ manual. Experimental merges should be performed to see the effects of such operations, and to see whether further preparations on individual projects are necessary before the merged information is used seriously.

CAQDAS Networking Project Comment on N6:

- *The decision you have to make about text unit type (or the minimum codeable chunk) means that you need a good awareness at an early stage of how you will use certain tools at a later stage. To start with you may not be thoroughly familiar yet with all the tools or what they will do for you, or how the ways that data are prepared can affect the efficiency of the software later; for instance text search tools, and some command files can make use of structures and syntax within the data that need to be in place before, or shortly after, the data is imported. This can make the initial familiarisation and/or teaching of the software a little problematic.*

- *The text unit structure also means the coding process may not be as flexible as in other packages, but*

- *Conversely, however, the text unit structure may be particularly useful for large or longitudinal projects as it allows coding (of large amounts of data) to be achieved quickly. Additionally, text only format for data files means that searching tasks and browsing codes may happen quicker than other software which use Rich Text Format.*

- *The automation (e.g. with command files) which it is possible to build in to a large N6 project exceeds what is possible in other packages. Requiring such automation would therefore be a good reason to choose N6.*

- *Excellent range of search tools in N6 with fairly user friendly dialogue boxes*

- *Although a report can be generated to show coding in margin, it is not instant and it is very restrictive and limited compared to other margin displays*

- *Data can be edited but only by opening up each text unit individually and changing and saving that text unit. Not to be recommended on a wide scale.*

- *The Quick coder tool new to N6 is really useful – allowing those who prefer to code in the more traditional way in hard copy to do so and then easily transfer coding to N6 in a easy and quick way.*

Nvivo (Version 2) – some distinguishing features and functions

NVivo is the sister package to N6 and is also developed by QSR. It is therefore similar to N6 in a number of ways – for example, in its database structure, coding schema and some of its functionality. However, both software have distinctive and separate strengths (see N6 above) and although NVivo offers a range of additional tools and flexible ways of working it should not simply be seen as the 'latest version' in the NUD*IST range of software packages. Rather it is a package in its own right and enables inquiry to go beyond coding and retrieval. http://www.qsrinternational.com/

Minimum System Specifications (recommended by developer)

MS Windows Me, 2000, XP
RAM : for Me 64Mb, for 2000 128Mb, for XP 256Mb
40 – 125 Mb disk space required (plus space for project data files)

Structure of work in NVivo 2

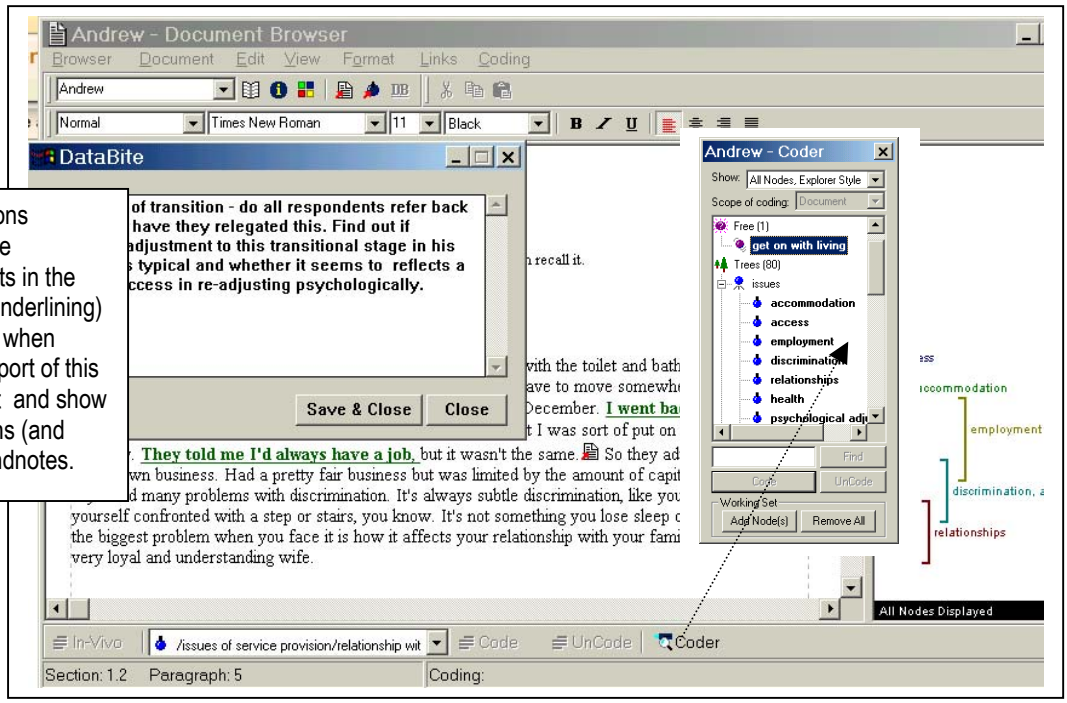
The user creates a Project and into that the individual data files are imported. The internal database means that when the user moves project the data files are moved within it. The project itself becomes a folder on your computer, within which various other folders sit.

The main ***Project pad*** provides initial access to the main functions and it has two foci, *Documents* and *Nodes.* Selecting either of these changes the buttons on the pad affording more or less symmetrical functions for handling the associated tools for documents (individual data files) or nodes (codes, or coding related tasks).

Data types and format in NVivo 2

Files which can be imported for direct handling in NVivo must be in *Text only* or *Rich text format* (.rtf).
Full editing rights on data files: they can be marked, edited or annotated at any stage. This also means the researcher's own writing in the form of a memo or annotation can be easily coded.
Blank files can be created within the software to contain analytic notes about documents, codes, or perhaps to transcribe summaries of multimedia files. Proxy files can represent other files outside the Nvivo data base.
Links can be inserted in NVivo textual documents, to external e.g. any multimedia file or file which is accessible on your computer. Multimedia files cannot be coded, although abstracts or proxy documents (which can have links to the multimedia file as a whole) can be coded normally.

Internal annotations (databites) can be anchored to points in the text (see green underlining) – and re-opened when required. Or a report of this document will list and show text of annotations (and other links) as endnotes.



Closeness to data and Interactivity in NVivo 2

Documents can be browsed and annotated before any coding takes place. Browsing coded data – lifts out the segments from their source position, though the user can return to the relevant position in source file (2 clicks away). The margin display showing coding is not interactive, though selecting any code in the Coder window will interactively colour all the data linked to that code in current document, blue. Versatile linking devices (databites, doclinks and nodelinks) increase the integration between memos and data, documents and other documents, documents and coded data.
Document Explorer lists and gives access to all documents and much functionality concerning data files occurs here
***Sets* of documents within Document Explorer** allow user to create collections of short cuts to documents for any reason e.g. *all coded documents, all interviews containing some aggressive responses…etc. etc.*
Coloured document icons allows user to visually differentiate between different features or types of data files to visually mark features of data.

Coding schema in NVivo 2

The coding schema can be as hierarchical or un-hierarchical as required and is visible in the **Node Explorer** window – and also in small **coder pane** while in Coding mode. Reorganise coding schema with **Drag and drop**.
Codes can be applied to any (overlapping, embedded etc.) text selection
Drag and drop codes on to selected text to assign coding – **Uncode** text easily (break link between code and data)
In Vivo button allows the use of word or phrase in data to generate new codes. You are reminded of all codes created in this way by the creation of a *Set* of short cuts to these **In Vivo** nodes
Sets of Nodes can be created at any time - allowing user to depart from existing hierarchical structure and to create groups of short cuts to collections of nodes for different reasons, e.g. to search on, or to theorise with use of certain codes – without changing hierarchical structures.
Margin stripes view of the data files offers a (static) display of how documents have been coded this view can also be printed. Margin display – NOT interactive.
Colour of codes in margin display: user has no control over colours used in the margin – and these will vary according to size of segment at that point.

Basic Retrieval of coded data in NVivo 2

Basic retrieval of data coded by one code happens by *browsing* a node/code – which lifts out coded segments from their source position, though the user can return to the relevant position in the source file (2 clicks away). These *browse* windows are live – in that the user can recode whilst in this view.

Organisation of data in NVivo 2

The (descriptive) organisation of data happens at the level of a spread-sheet like system of 'attributes' and such information can be imported from, and exported to, database or statistical packages.
Document Attributes can be assigned to whole files/documents
Node attributes can be assigned to parts of files which have been coded e.g. to assign demographic characteristics to speakers in a focus-group (see also autocoding tools below).
Sets of Documents, Sets of Nodes/Codes
Sets are user-friendly ways of creating short cuts or aliases to groups of files or nodes which allow the user to focus on those sets in many ways and at different moments when using the software. Sets can be created in a number of different ways; for instance simply by dragging and dropping in the Document/Node Explorers, or more analytically, from the results a search, or using the Set Editors. Sets can be useful when *scoping* or directing a search on a part of the dataset, when *assaying*, when writing, when exporting frequency information, or as a subsidiary way to organise or visualise collections codes or documents; for instance, to create more theoretical collections of codes without upsetting original listing of documents or without re-arranging the hierarchies of codes/nodes.
It is also possible to use **case nodes/codes** to organise data.

Writing tools in NVivo 2

There are a number flexible writing tools in NVivo, allowing the user to write in many different places.
Memo's: Blank documents can be created as memos at any time, or linked to any existing document or node. These are full ranking documents and therefore can also be coded.
Internal annotations: comments can be embedded at any point in any document and re-opened when required. Or a report of the document will list and show the text of annotations (and other links) as endnotes.

Searching and Interrogating the dataset in NVivo 2

The Search Tool is similar to that in N6 in terms of some of the types of searches that are possible but has some additional more sophisticated functions, which allow the user to integrate different types of searches, to flexibly scope or direct a search and in different ways, save the results;
Qualitative matrices: Unique to QSR software at present, with increased flexibility in NVivo compared to N6, is the ability to build qualitative cross-tabulations (matrix searches), which as in N6 are interactive. The results are summarised with variable frequency display options in the table, giving access to the relevant qualitative data from each cell of the table.
The Assay Tool associated with the Search tool can for example, generate an overview of the presence or absence of requested items (codes, attributes) in a document, or set of documents is similar in principle to the graphic Show relations tool – see below. They both offer ways to look into a project *from above* after much of the coding and organisation of the dataset has been done.
Autocoding tools (connected with Text searching tools)
Automatic 'Section' coder makes use of Heading levels inserted at subtitles/questions/speaker Ids. Allows quick coding of e.g. different speakers in a focus group – (prior e.g. to assignation of Node Atttributes)

Output in NVivo 2

Normal coding output and flexible and varied tabular output. It is possible to print a document or node (coded data) with the coded margin view on display.
Models can be exported to Word, either via clipboard or via generation into graphic file formats.

Team-working in NVivo 2

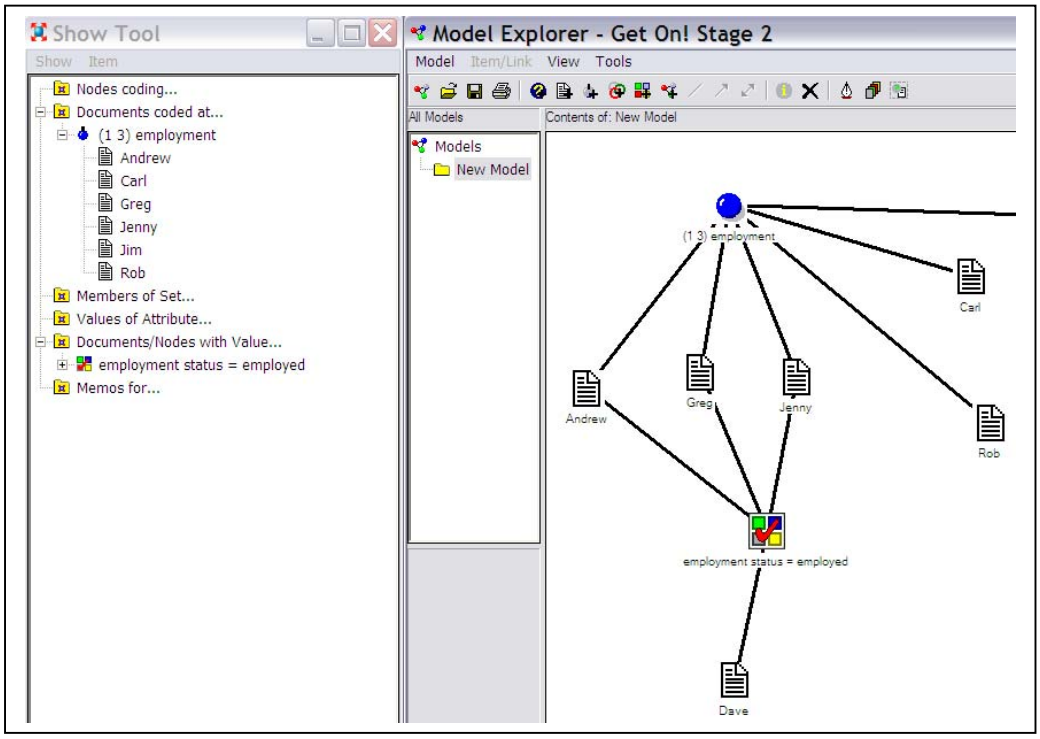
Merge for NVivo needs to be purchased by one member of the team, to enable merging of separately worked on projects. Experimental merges should be performed to see the effects of such operations, and to see whether further preparations on individual projects are necessary before the merged information is used seriously.

Ability to go beyond code and retrieve in NVivo 2

Linking tools
There are a number of different ways of linking data and ideas; – for example, linking to externally held multi-media data files and websites);
Databites create internal annotations anchored to a specific point in the data; or to external e.g. multimedia files (although these files cannot be directly coded).
Doc links linking across data files (2 clicks),
Node links create links to coded data from any point in any data file – for example you may be writing a memo, and in it you wish to link to a collection of coded segments of data. Making each of these linkages puts an icon representative of the type of linkage you have made *in* the data.

Modelling and mapping and Show Relations tool

The modelling tool offers a visual way to present ideas and theories which are linked to the data represented. It operates as a graphic scribbling pad in that links made do not have any implications beyond that particular model. As such, any number of models can be created and different types and styles of links between the same codes created in different models – without affecting the coding schema or the results generated from other software tools. In addition, the **Show Tool** – especially when used in conjunction with the modeller – offers a powerful way to (graphically) interrogate the project from the 'top down' – for example, integrating the searching of coding and attributes already applied with the ability to visualise the results in a graphic display.



Create relationships or express ideas in the modeller

…also there is good interactivity between elements in the model and the data itself.

CAQDAS Networking Project Comment on NVivo 2:

- *Very powerful software, especially in the range of ways given to handle data and to search later on. Like N6 has a large range of searching possibilities. Matrix searches in particular offer superb ways of doing multiple searches at a time and accessing results. Search tools allow the integration of different types of searching, and are particularly flexible.*

- *Creative and flexible functions enabling writing, analytic notes and integrating work and thoughts by using linking devices.*

- *The margin display is useful but due to its lack of interactivity can be a little misleading at times. The print out of the page which includes the margin display makes the text difficult to read.*

- *Very strong in systematic support given to user, but multiplicity of ways to organise data is both a strength and a difficulty. The different uses of Sets, Cases, and Attributes are initially difficult to work out, and some beginners may need advice. This should not detract from the flexibility these tools add to the basic handling of data, and the powerful ways they enable search tools to be used efficiently later.*

- *The software uses up RAM, and the rather untidy user interface (its possible to have several versions of the same pane open, but buried under other panes) exacerbates this problem, especially for users who are as yet unfamiliar with more efficient ways of moving around software.*

- *The internal database and the Rich Text format of data means that some browsing tasks and searching exercises can take a little while to execute. This is a difficult thing to predict and you are generally advised by the developers that this is not a software for very large datasets. Check this out with the developer.*

- *The merge tool is flexible and easy to use, however because it is not integrated into the main software, it is important to budget at an early stage if you need this facility.*

QUALRUS (version 2.0.4.0.) – some distinguishing features and functions

QUALRUS was developed by Idea Works, Inc. with support from qualitative researchers including Howard Becker. It differs significantly from other *theory building* CAQDAS software packages because it uses 'intelligent computational strategies' at various stages of the analytic process to assist the researcher with certain tasks. It may therefore be seen as the first in a new generation of qualitative data analysis software packages; indeed it is intended for application beyond qualitative research. http://www.qualrus.com

Minimum System Specifications (recommended by developer)

MS Windows 95 or higher
48Mb RAM (required), 128Mb (recommended)

Structure of work in Qualrus

Qualrus functions using an external database structure, whereby the software refers to source files located elsewhere on the computer rather than physically copying them into the project.

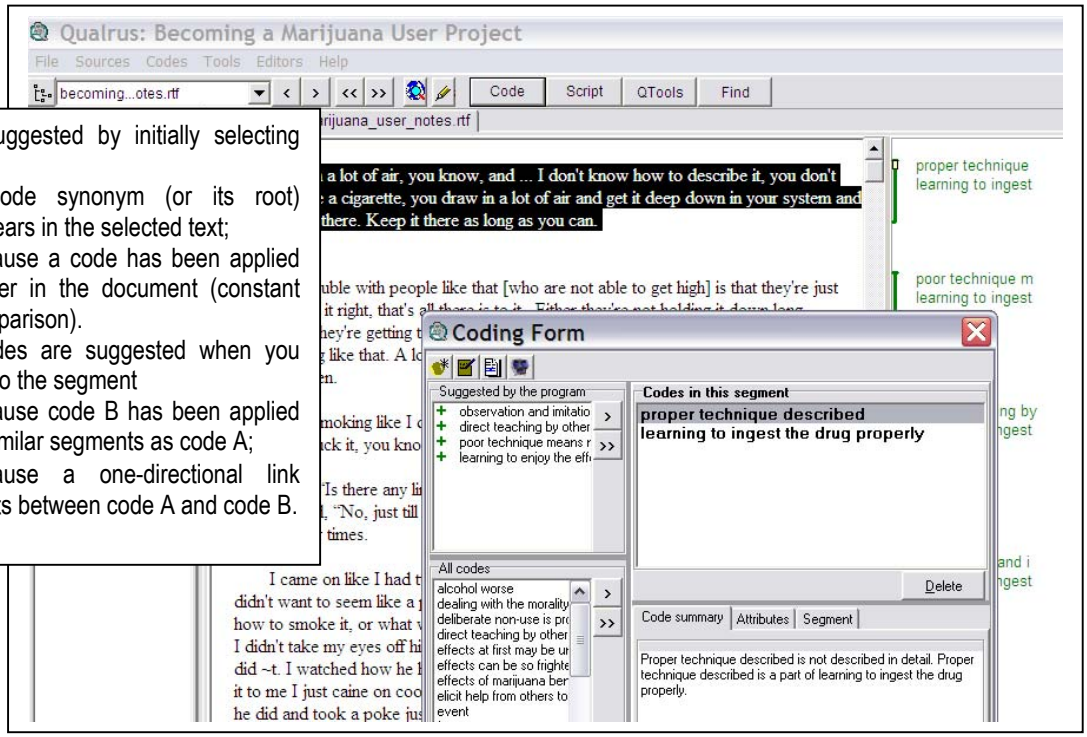
The segment is the basic unit of analysis in Qualrus which affects the way certain search tools function. Codes can only be applied to segments. There is complete flexibility concerning the size and number of segments created - segments can overlap, be embedded within one another etc., and as many codes can apply to one segment as is required. However, it is not possible to apply codes to text which is not already a segment – which may affect the utility of the auto-coding functions (see below).

Codes are suggested by initially selecting text because:

1) a code synonym (or its root) appears in the selected text;
2) because a code has been applied earlier in the document (constant comparison).

Additional codes are suggested when you apply code A to the segment

3) because code B has been applied to similar segments as code A;
4) because a one-directional link exists between code A and code B.



Data types and format in Qualrus

Qualrus directly handles multi-media data in a number of formats including plain text (*.txt), rich text (*.rtf), video (*.avi), audio (*.wav, *mp3) and graphics (*.bmp, *.jpg). Version 2 enables the downloading, saving and coding of *html sources. It is also possible to navigate the internet from within Qualrus. See developer information for complete list.

Closesness to data – Interactivity in Qualrus

The researcher is kept close to the source data throughout the analytic process – for example, you are usually just 1 click away from the source data and context when , for example, viewing the results of a search (see below). The margin view provides an overview of how a data file has been coded and is also interactive – i.e. clicking on a code in the margin will highlight the segment to which it is applied.

Coding schema in Qualrus

In the first instance the coding schema in Qualrus in un-hierarchical. The advice of the software developers is to develop the coding framework as far as possible before beginning the process of assigning codes to segments of data – which (in order to make the most of the software's capabilities) involves creating a semantic network representing relational links between codes, specifying code synonyms and defining and applying attribute-value pairs to codes.

Margin display

The margin view is interactive but the user has no control over colours used in the margin – which will vary according to the size of segment at that point. When code labels are more than about 18 characters long, they will be truncated – however wide the margin space is.

Coding processes in Qualrus

Qualrus can take an active role in the process of coding qualitative data as it is able to 'learn' as the researcher proceeds, offering suggestions as to how to code text segments based on natural language, code synonyms and (user-defined) links between codes. The process of coding takes place in the Coding Form and whilst it is easy to apply codes to individual segments within this window, the Coding Form has to be closed to select a new segment – and then re-opened to continue coding. There is no drag-and-drop coding or short-cut coding icons.

Basic Retrieval of coded data in Qualrus

Retrieval of coded data can only be performed by using the search tools (QTools), but retrieval is always interactive – whilst in the first instance it is 'lifted-out' of context (enabling easy outputting), clicking on a segment in the results pane shows it in its original context.

Organisation of Data in Qualrus

Organisation of data, if required, happens at the coding level within Qualrus if the user wants to make use of such organisation within subsequent searches.

Writing tools in Qualrus

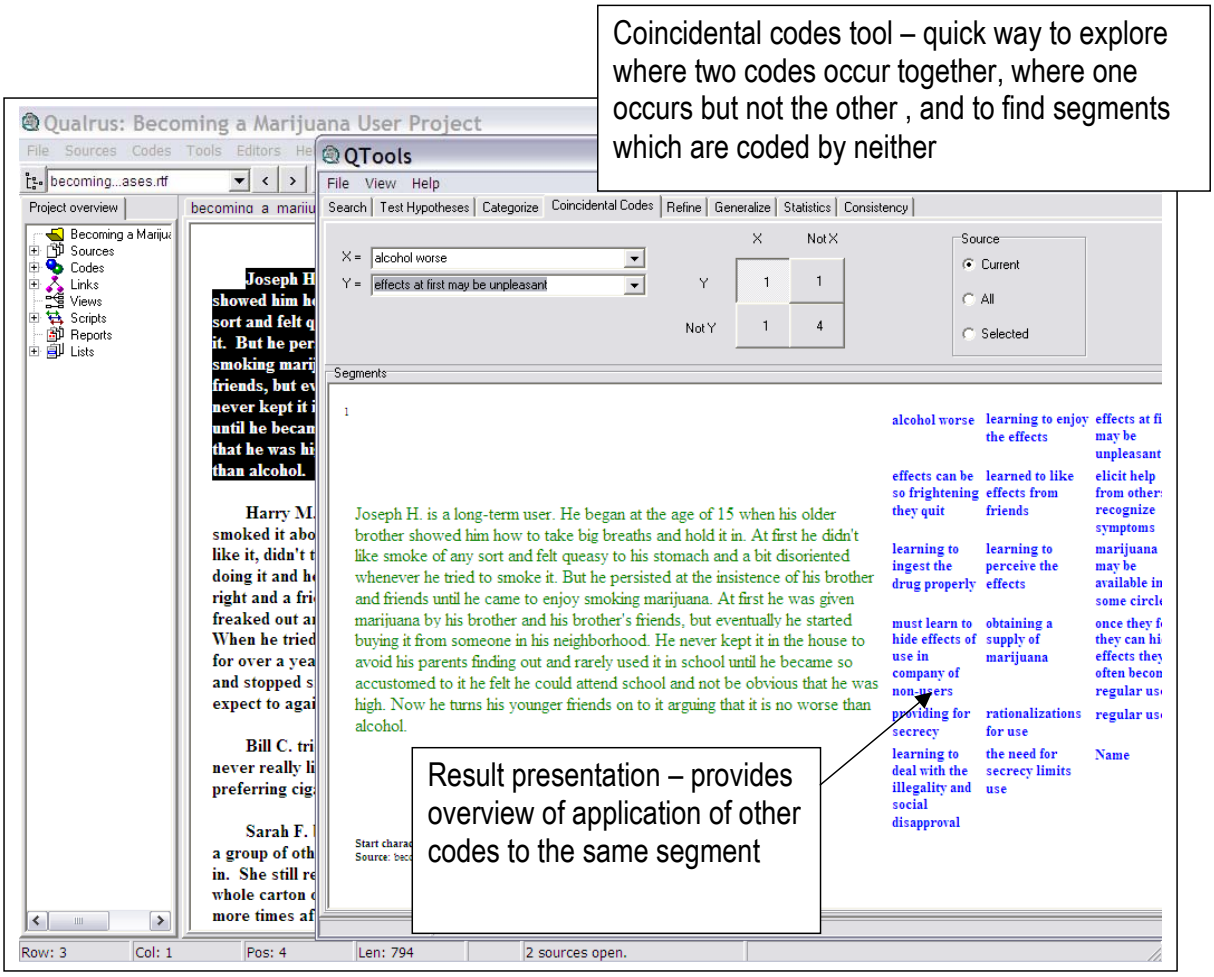
It is possible to edit data within Qualrus, however, doing so once coding begins runs the risk of messing up the precise segment to which a code applies. Memos can be attached to sources, codes and segments, and it is possible to search for segments which have a memo attached to them for outputting purposes.

Searching and Interrogating the dataset in Qualrus

As well as being able to search coded data based on Boolean operators e.g. AND and OR, a number of additional QTools are available which help to interrogate the dataset and your thinking in different ways, these include the following:

The **statistics** tool – designed to assist with theoretical sampling by providing basic statistics summarizing, for example, the number of times a particular code is found in a source or how many times two codes occur at the same

segment.
The **coincidental codes** tool – enables the examination of the co-occurrence of each possible pair of codes in segments, ordering pairs by how frequently they co-occur. This can facilitate assessing the relationship between codes – to quickly see how often two codes co-occur and compare and contrast segments where both codes occur, only one occurs, or neither occur to help assess any possible relationship.
The **categorizing** tool – offers an additional way to think about and code data, facilitating the process of identifying evolving categories and generating higher concepts as coded text segments can be sorted into different stacks which themselves can be assigned codes.



The **hypothesis testing** tool allows the researcher to examine all segments to test specific hypotheses. Researchers can see a summary of how well the hypothesis accounts for the data as well as examine the segments that support the hypothesis and those for which it is not true.
The **refine** tool helps researchers examine either codes or relationships among codes to determine whether those need to be further refined, such as when two codes lump together too many diverse items that should be refined or separated into different categories.
The **generalization** tool helps researchers assess whether two or more concepts are really measuring the same thing and should be combined into a single broader concept, or perhaps are each special cases of that single broader concept.

Auto-coding – text searching

Because of the segment structure underlying Qualrus the autocoding tools are not as flexible as in other packages – whilst it is possible to search for words or phrases (in one source, selected sources or the whole dataset) and return the sentence or paragraph around the find – if these units of text are not already identified as segments, it is not possible to automatically apply codes to the results.

Output in Qualrus

There are two types of output possible : Lists and Reports
Lists remain internal to Qualrus and are useful ways to save results from which to build further questions upon.
Reports are ways to save output for viewing outside of Qualrus and you have the choice to save reports in either basic or dynamic html format – providing easy sharing of output. Dynamic html format is particularly flexible – allowing non-expert Qualrus users to navigate around your work.

Ability to go beyond Code and Retrieve in Qualrus

Extensibility
One of the key distinguishing features of Qualrus is that it is extensible in that the researcher is able to modify the program to fit their specific and evolving analytic needs. Using an object-oriented scripting language (similar to C++ or Java) and the script editor, researchers can write scripts to perform tasks in order to address specific problems. The researcher is thereby able to extend and tailor Qualrus according to theoretical perspectives, substantive problems or practical applications.

Team working in Qualrus

Qualrus includes a merge facility (at no extra cost) so that different researchers can work on the project and merge their changes. The coder consistency tool may facilitate team-working by providing statistical data showing how consistent each coder is with the recommendations of the program.

CAQDAS Networking Project Comment on Qualrus :

- *Qualrus is a highly sophisticated software with a number of tools which are not currently available in any other package.*
- *It can be used very simply without having to do anything very complicated – and from that basis there is a lot of functionality waiting for you to make use of – sometimes by merely clicking on a button. Several of these functions provide different ways to, for example, compare codes, refine your coding schema, look at the statistics of coding, the coincidence of your codes together.*
- *In nearly every window there is very good interactivity between that window and other aspects of your work.*
- *The segment structure is a little restrictive in comparison to other software packages – specifically the current inability to search for where codes overlap in the data but are not applied to exactly the same segment.*
- *There may be some debate about the appeal of Qualrus' Artificially Intelligent aspects and of its use as a generic qualitative data management and analysis tool – particularly for novice qualitative researchers and/or software users who may feel under confident about scripting. Making the most use of the intelligent coding strategies may be partly dependent on having a fairly mature coding schema at the outset and may therefore be particularly appropriate to more deductive coding approaches.*
- *There is as yet no straightforward way to compare codes etc. across subsets of your data – e.g. comparing how the men and women talk about an issue*
- *Qualrus has a very different feel from the other software packages reviewed here and is not as flexible in the straight-forward handling of data – however, its extensibility does mean that for the confident computer user who is willing to learn the scripting language its potential flexibility is almost unlimited.*

Coming next…

…QDA Miner
…some free software – Transana and InfoRapid
…non code-and-retrieve based software – Storyspace