Choosing a CAQDAS Package

A working paper by Ann Lewins & Christina Silver

5th edition July 2006

Drawn from forthcoming Sage Publication "Using Software for Qualitative Data Analysis : A step-by-step Guide" 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
auto code 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

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

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.

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; if the user is specifically concerned with the efficacy of this tool for particular purposes, attention should be paid to the units of text that can be returned or *auto coded'* around the actual finds or hits (providing more context), i.e. number of characters either side, sentences, paragraphs, whole document; note that some software provide all options, others only some and this may matter.

Awareness that you can also later search for relationships between codes as they have been applied to data (for example, co-occurrence, proximity etc.) enables the researcher to be conscious that the results of such searches will produce a secondary level of analytic coding. 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?

Included below...ATLAS.ti 5, HyperRESEARCH 2.6, MAXqda2 (MAXdictio & MAXmaps), N6, NVivo2, NVivo7, QDA Miner, Qualrus and Transana
…also look out for summaries of InfoRapid,
…and non code-and-retrieve based software – Storyspace

Summaries of some Theory-Building CAQDAS Software Packages

ATLAS.ti V5, HyperRESEARCH V.2.06, MAXqda V2, N6, NVivo 2, NVivo7 QDAMiner, Qualrus, Transana

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 Technical University of Berlin as a interdisciplinary project between the Psychology department, computer scientists, linguists and future users (1989-1992). The prototype was then further developed by Thomas Muhr and the company ATLAS.ti Scientific Software Development GmbH, formed in 2004, continues to develop and support the software. http://www.atlasti.de

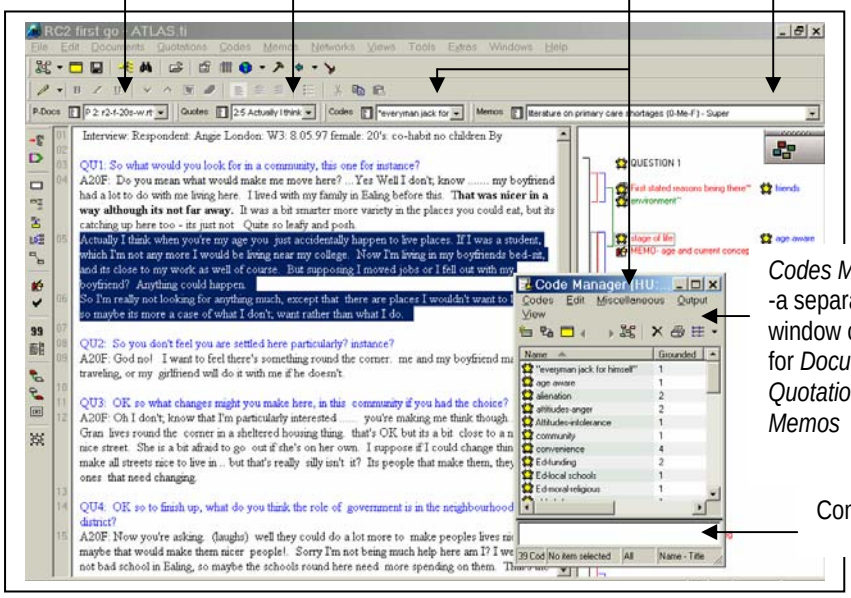
Minimum System Specifications (recommended by developer)

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 what we call 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 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*

Comment bar

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 *Superfamile*s 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 from several different places, including the Codes Manager window and 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 auto coding of finds or hits can be controlled as the search is done; the user can interactively code more text as the software finds and displays each hit in its full context (KWIC). Or the auto coding search can be done routinely without confirmation, just coding the hit, the whole word or 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: This tool allows the user to select a code (in the co-occurrence explorer, the Query Tool or 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.
The Object Crawler : Providing a powerful way to search for text and patterns across the whole HU (i.e. any object)., although searches can be easily restricted (scoped) to particular aspects of the project, for example, name, comment, content etc . Results are interactively linked.

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. Users can work on the same HU but only one user has the right to save changes under the original project name. 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.

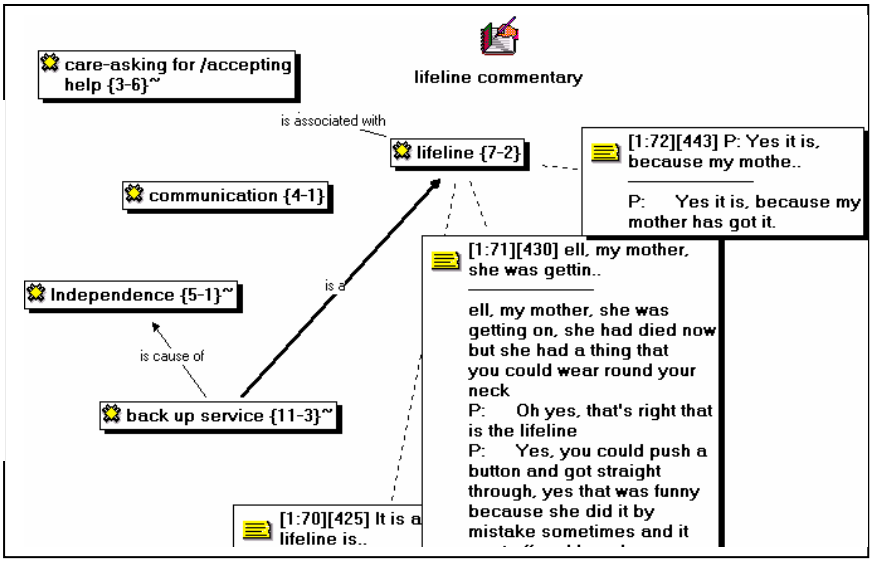
Hyperlinks between quotes – alternative to coding

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

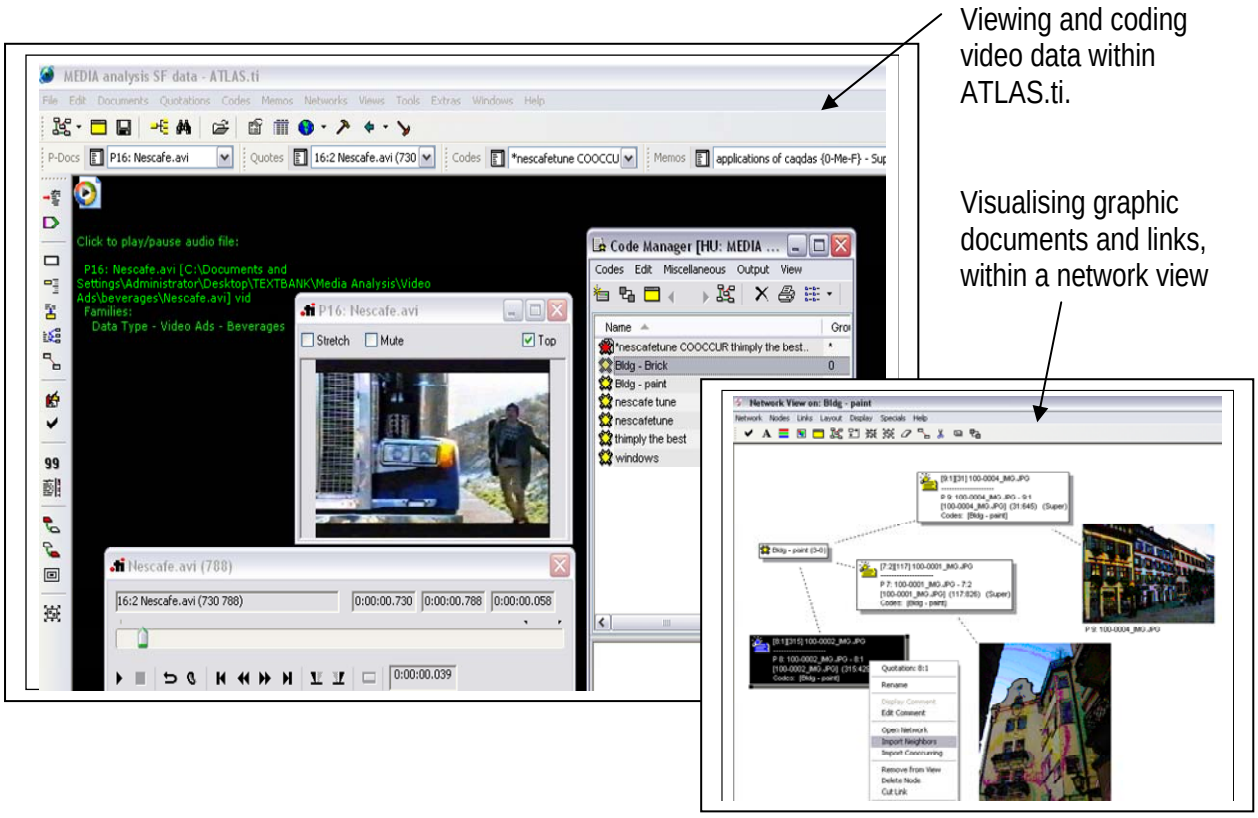
This tool counts the occurrence of words across the whole dataset, or a single file and the results can be saved into an exported spread sheet file or into a memo. The list of words is not interactive – i.e. no KWIK (but see later Searching tools for KWIC)

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.

Multi-media data in ATLAS.ti

Non-textual data such as graphics, audio and video data can be directly assigned to ATLAS.ti and treated in similar ways as written documents. This allows for full integration within one HU of data derived from different sources which contribute to the same research project. Quotations can be easily created out of multi-media documents and treated at the same level as textual quotations. Hyperlinking between quotations provides a useful, if step-by-step, way of linking parts of written transcripts to corresponding audio/video segments.



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).*

- *The fully interactive margin display shows all or selected codes, and documents can be printed showing all the codes.*

- *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 ability to assign non-textual data files to an HU allows a broad range of information and data corresponding to the research to be incorporated into the same 'work space'. Tools for analysing audio/video data are not as sophisticated or slick as in programs primarily designed for this purpose, but ATLAS.ti provides better ways of handling multi-media and textual data together than most of the packages reviewed here.*

- *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. Some search operators have very precise parameters and the user must be aware of these to read results and their implications reliably. It lacks the ability to combine searches for text in coded data.*

- *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 can be useful in many circumstances, for instance in recovering 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

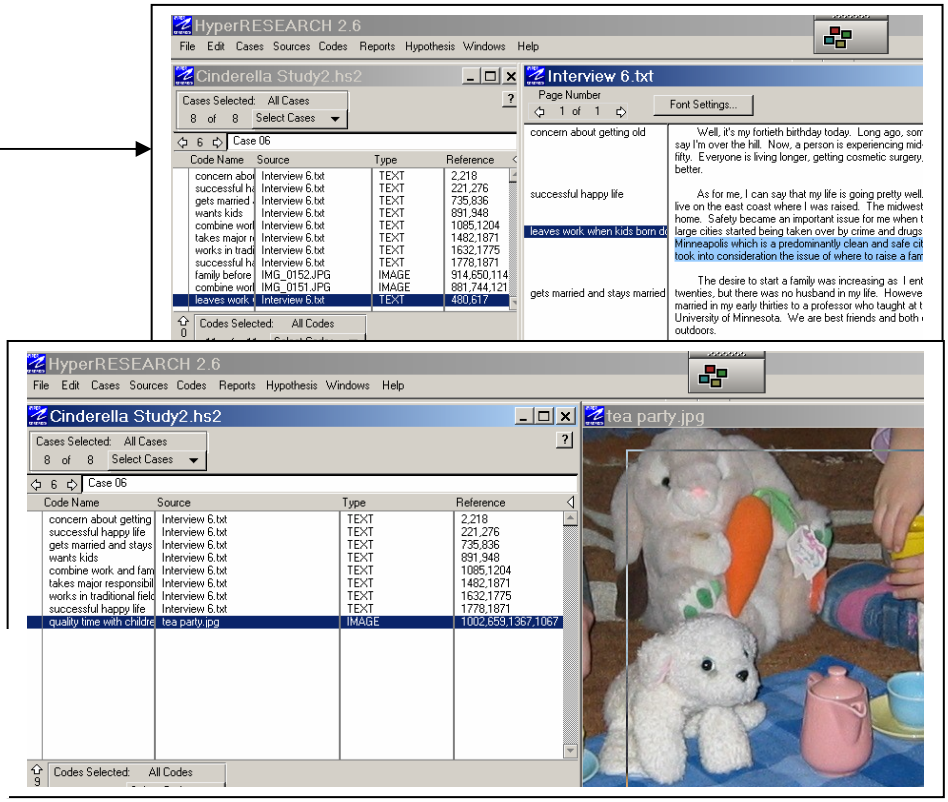
The project as a whole, is called a Study. The database is external – so the source files are accessed by the Study, not contained within it. The Study is comprised of Cases, and Cases can have references to one source data file or many. The Coded references are listed in the Case card for each case,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

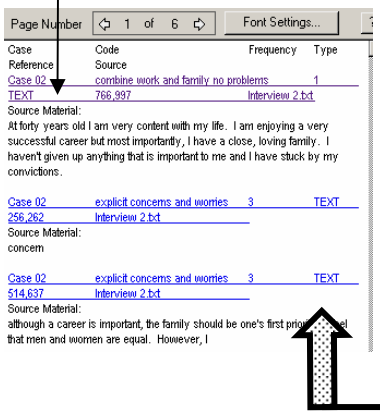
Coding processes in HyperRESEARCH

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
Double clicking on the code from the codes list assigns code to selected data. 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.

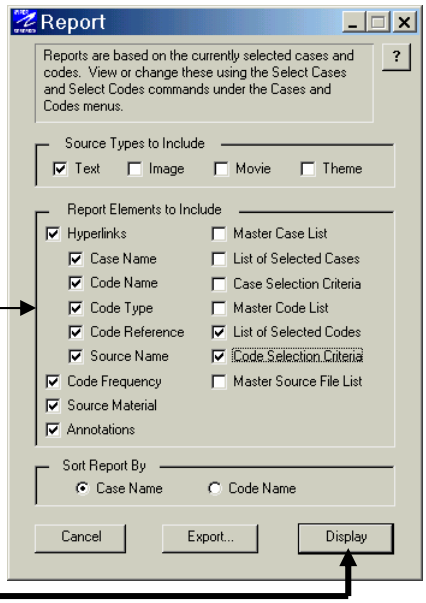
Basic Retrieval of coded data in HyperRESEARCH

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.

Auto coding: Searching tool allows user to search and e 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, or the whole documents. Sentences are not recognised. The user must proactively tell software which files, within which cases, to search, settings can be saved and reloaded (and amended) for future 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, but very step by step.*

- *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 report builder and its output, reflects whatever state of filtration in Case or Code selection is currently in place. The beginner could do with a reminder of this, in the report builder itself.*

- *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 if you wish to move the files to another computer.*

MAXqda2 – distinguishing features and functions with add-on MAXdictio & MAXmaps

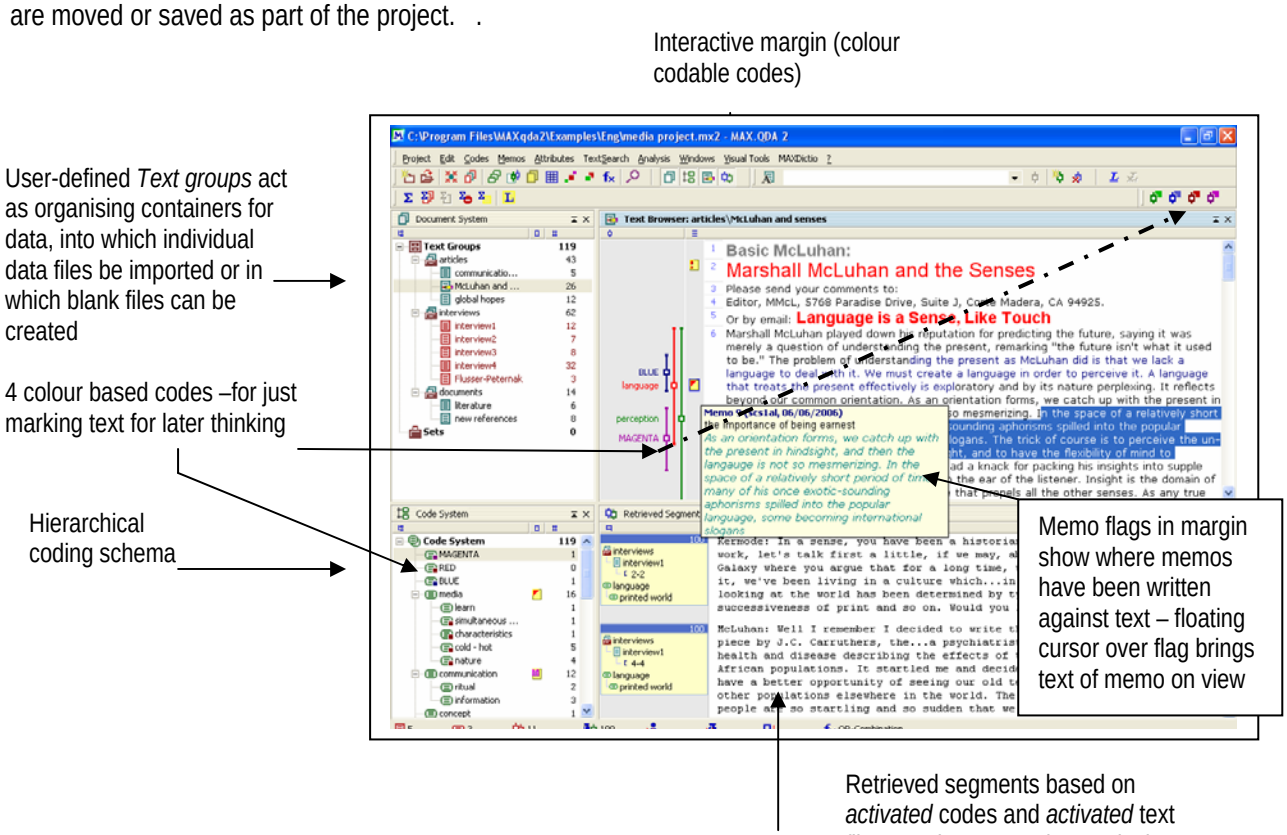
MAX, then WinMAX Pro, MAXqda and most recently MAXqda2 are part of a software stream originally authored by Udo Kuckartz in order to cope with the management and analysis of political discourse data on the environment. Its application has widened to multiple academic disciplines and applied research 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. MAXmaps (extra cost) ads the ability to draw graphic maps e.g. of the connections between codes etc. Contact for information: info@maxqda.de www.maxqda.de

Minimum System Specifications (recommended by developer)

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

The Structure of work in MAXQDA V2

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. .



Data types and format in MAXqda/MAXqda2

Textual data: files have to be saved in *Rich Text Format* –no other format of data is allowed in MAXqda, but new in MAXqda2, you can now have gaphics, text boxes, tables etc inserted into rich text format files.
Rich Text Format allows full editing rights on the data once in MAXqda/MAXqda2. Any amount of text can be selected for coding purposes, but only the whole of the inserted graphic or multimedia file can be coded in MAXqda2.

Closeness to data - Interactivity in MAXqda/MAXqda2

Interactivity from Coded segments in the Retrieved segments pane back to source data file (one click), and between codes in margin and source data, mean that the user is 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.

Text Links: New to MAXqda2 2 text passages can be linked together by means of hyperlinks. These are underlined in the same way as a link in an HTML text. Internal links (to other text passages in the MAXQDA project) and external links (to websites) are visually and functionally the same.

Coding schema in MAXqda / MAXqda2

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 / MAXqda2

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. New to MAXqda2 there are increased colours to choose from.
Apply 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 frequencies can be converted into attributes New to MAXqda2 - by selecting the option "Transform into Attribute" in the context menu for the selected code. A (numerical) attribute with the given code as heading is then entered in the attribute matrix.

.

Basic Retrieval of coded data in MAXqda / MAXqda2

A*ctivation*: 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 *Sets* 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 / MAXqda2

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 / MAXqda2

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 in MAXqda / MAXqda2

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.
Interactive Code to Code Matrix Browser: New to MAXqda2 a visualisation, in the form of a kind of matrix, of the text segments that have been assigned a code. The matrix shows a little square, or knot, for each text segment that has

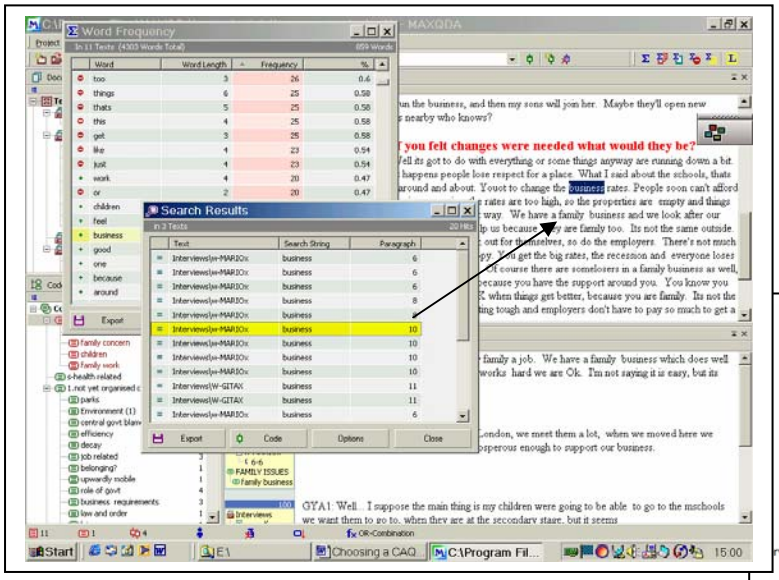
been assigned a code. The colour and size of the knot show how often the code was applied. Clicking on the knot interactively connects with the corresponding coded text segment in the RETRIEVED SEGMENTS window.
Interactive Code to data (*Code Relation*) Matrix Browser - New to MAXqda2 as above but provides matrix of codes by data.
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.
Auto code / 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.
Combining text searching with coded data - search expressions can integrate/combine the searching of words or phrases, in code

Output in MAXqda / MAXqda2

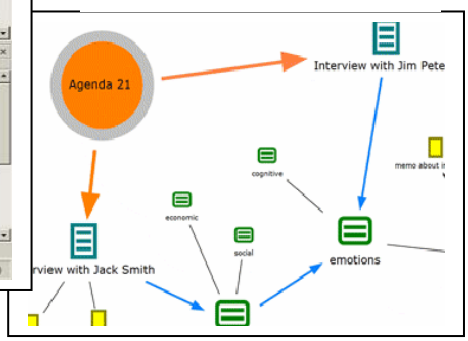
Any of the 4 main panes of the user interface van (and their contents) can be printed or exported into file format (Text manager pane, Text Browser pane (with codes in margin), Codes list or Retrieved segments pane.
Memos can be outputted in one file to be opened in Wordpad or Word (either all memos, 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
Exporting in HTML format: New to MAXqda2 it is now possible to export MAXQDA memos, attributes and coded text segments in HTML format. The data can be displayed in tables by any web browser (e.g. Internet Explorer).

Ability to go beyond Code and Retrieve in MAXqda / MAXqda2

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



MAXmaps



MAXmaps
Allows user to create maps based on objects already in the project, like codes, memo's or create free objects representing any new e.g. theory or link or idea. Appearance and properties can be changed, and objects in the map are interactively linked to text, memos, codes etc. Any number of maps can be made, and links made within a map are

only relevant in the map. Thus the map provides a scribble pad which does not affect structures in the rest of the software.
Layers can be made based on exclusive collections of things within one map.

Teamworking in MAXqda / MAXqda2

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. In version 2, colour coding in margin can be used to indicate different team member's work , e.g. on same data files. Also user ID facility with new log-in facility for teams.

CAQDAS Networking Project Comment on MAXqda / MAXqda2:

- *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. The software has some very simple yet appealing features which users often request:-*
    - *The ability to colour-code codes (as they appear in the margin.. more colours with MAXdqa2)*
    - *Simple to achieve multiple code retrieval, clearly labelled, unfussy report output.*

- *The user interface would be rather cramped with larger datasets, at least 1024-760 pixels resolution to achieve the space needed (most computers currently operate at this or a higher resolution)*

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

- *Excellent memoing and writing tools – easy, varied and systematic memo retrieval options which may be particularly useful in any situation and especially 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 coloured margin display of coding appears continuously – the margin can be resized to accommodate more codes and prints out easily and satisfactorily.*

- *For single or multiple code retrieval. Just activate the codes wanted, and get clear code & document identified retrieval accordingly.*

- *The auto-coding tools are perhaps less flexible than in other software because the only units of context to which you can 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 most other code based packages(apart from QDA Miner) – because they provide interactive(KWIC) connections to the source context.*

QSR N6 – some distinguishing features and functions

NOTE: N6 has now been superceded by NVivo7 which combines the functionality of N6 and Nvivo2. However as researchers still use the software we include the review.

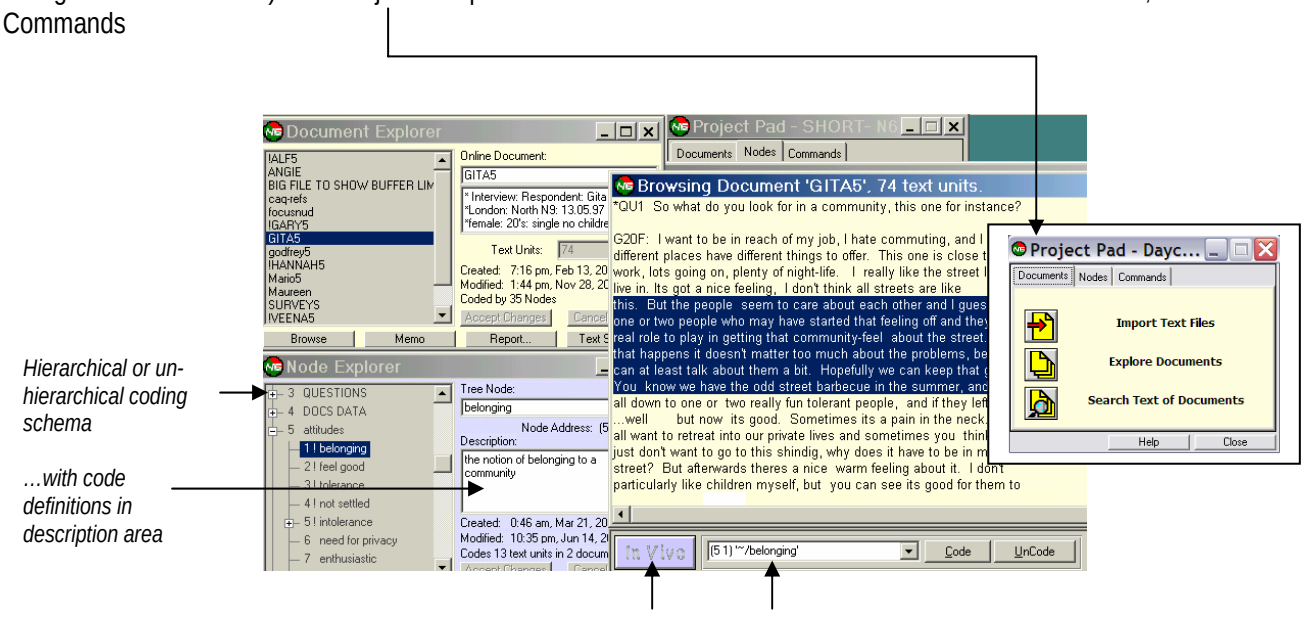
N6 was created by Qualitative Solutions Research (QSR) in the NUD*IST range of software, (Non-numerical Unstructured Data : Indexing, Searching and Theorizing). It was originally developed at La Trobe University, Melbourne, Australia by Tom Richards to support social research by Lyn Richards. Subsequently 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

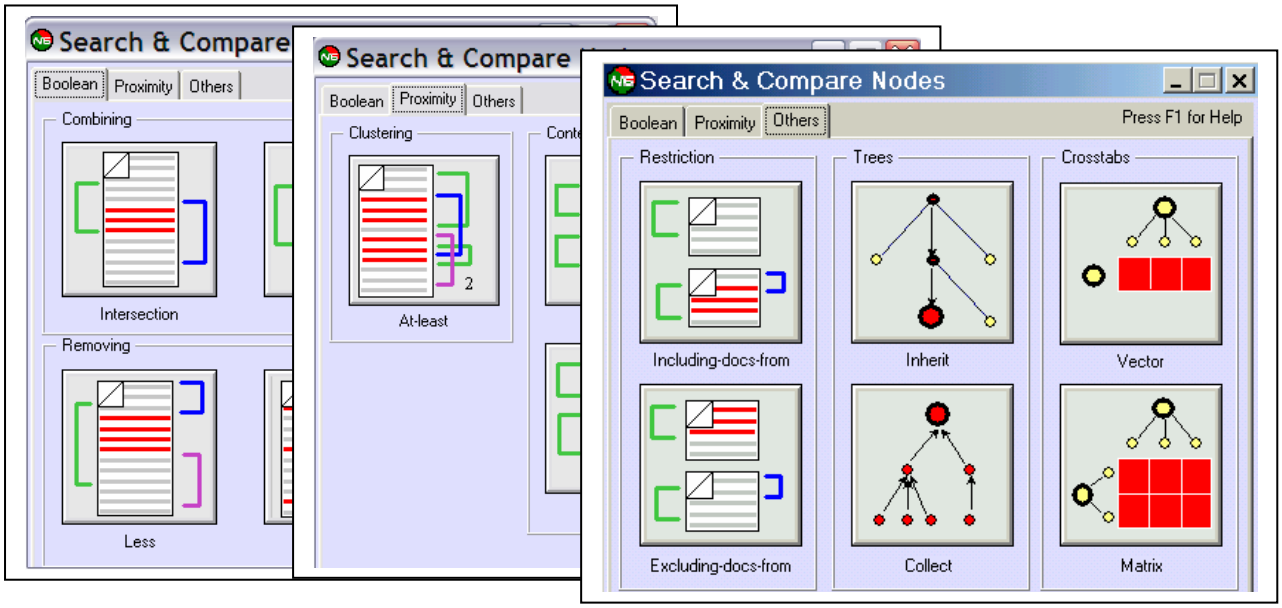
One memo can be created for each document, and for each node/code. *Annotations* which can be coded can also be inserted in the text occupying a new text unit, but this alters text unit numbering. All annotations can then be *browsed,* or combined in searches with thematic codes. Text units can also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

Searching and Interrogating the dataset in N6

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 / auto coding 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

NOTE: NVivo2 has now been superceded by NVivo7 which combines the functionality of N6 and Nvivo2. However as researchers still use the software we include the review.

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)

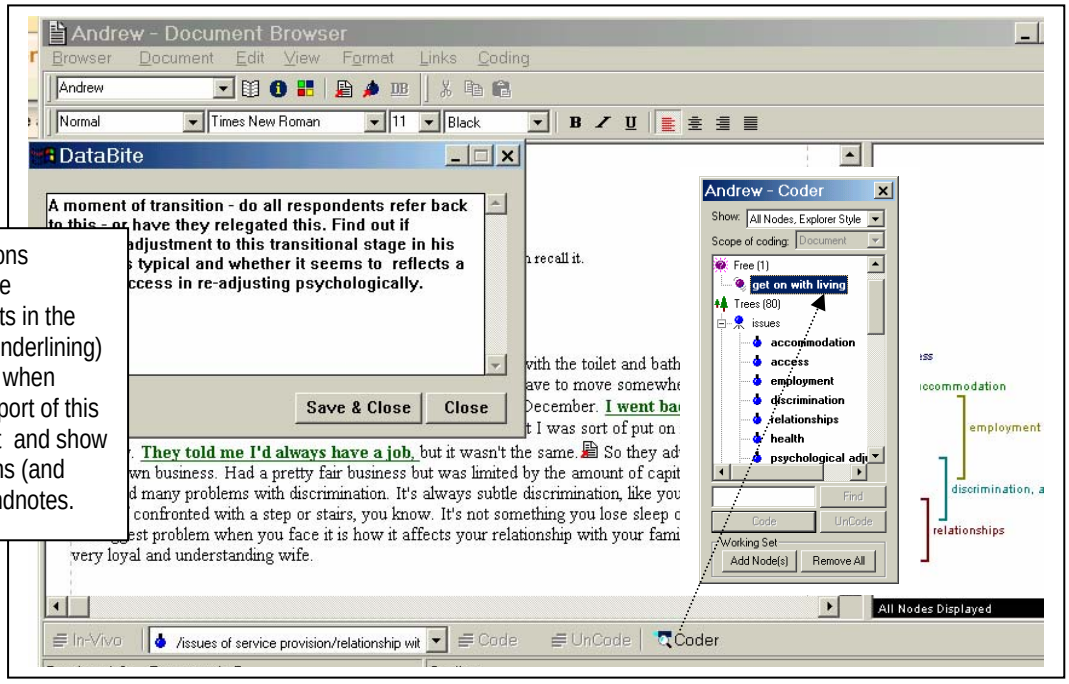
Structure of work in NVivo 2

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 auto coding 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.
Combining text searching with coded data - search expressions can integrate/combine the searching of words or phrases, in coded data
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.
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), or on several focus groups at the same time. Auto coding exercises which use the text search tool allow the inclusion of additional context around the finds/hits of up to 100 characters either side of the hit, the paragraph or the section. The sentence is not recognised by the software.
Auto coding via the Section coding tool provides very quick organisational coding of sections (which use heading levels consistently applied in the data).
Auto coding via the text search tool allows the user to coded up to 100 characters either side of the hits or a paragraph or a section. It does not recognise sentences.

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

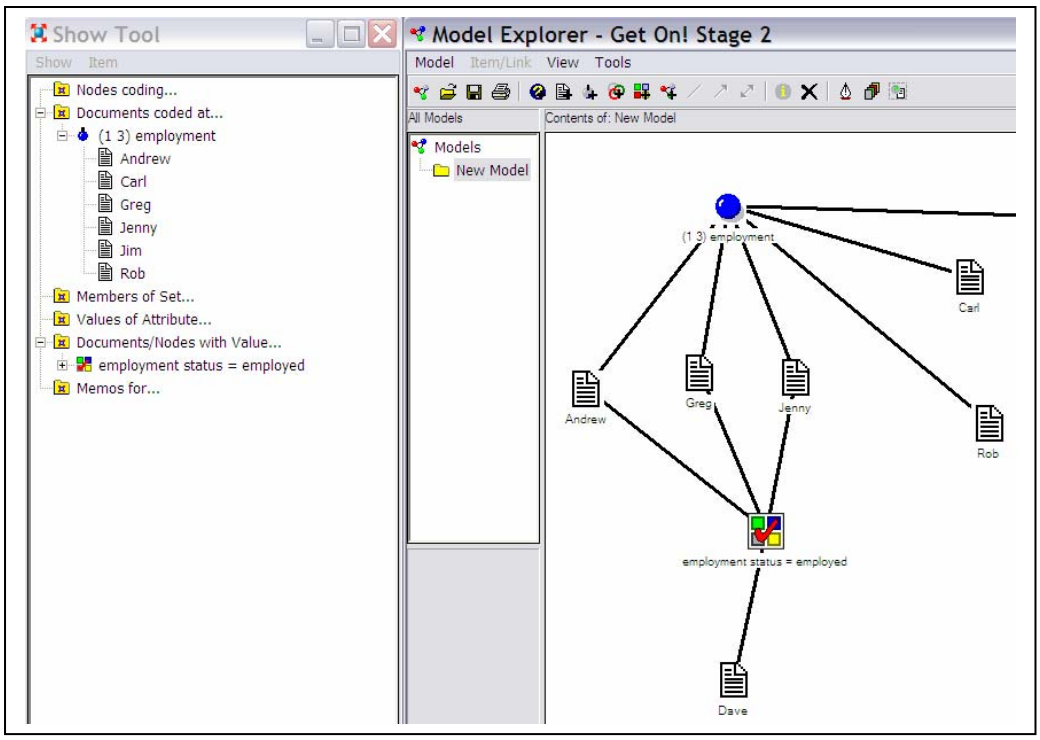
Linking tools

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. Thus 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.

Layers - any group of objects within one model can be grouped or put into layers to separate the model into different (conceptual?) facets



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.*

- *Some simple retrieval options are missing- e.g. ability to ask for multiple codes at the same time in clearly labelled output format (it is possible but only by very circuitous route not accessible to beginner) –a much requested option*

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

- *The margin display is useful compared to sister package N6, 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 a little 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.*

NVivo (Version 7) – some distinguishing features and functions

NVivo7 is the latest qualitative software package released by QSR International. It combines features from N6 and NVivo2 and includes some new functionality. The interface is designed using Microsoft Windows XP™ guidelines and looks similar to MS Outlook™. QSR International have stated that a new version of NVivo7 will be released on a yearly basis. http://www.qsrinternational.com

Minimum System Specifications (recommended by developer)

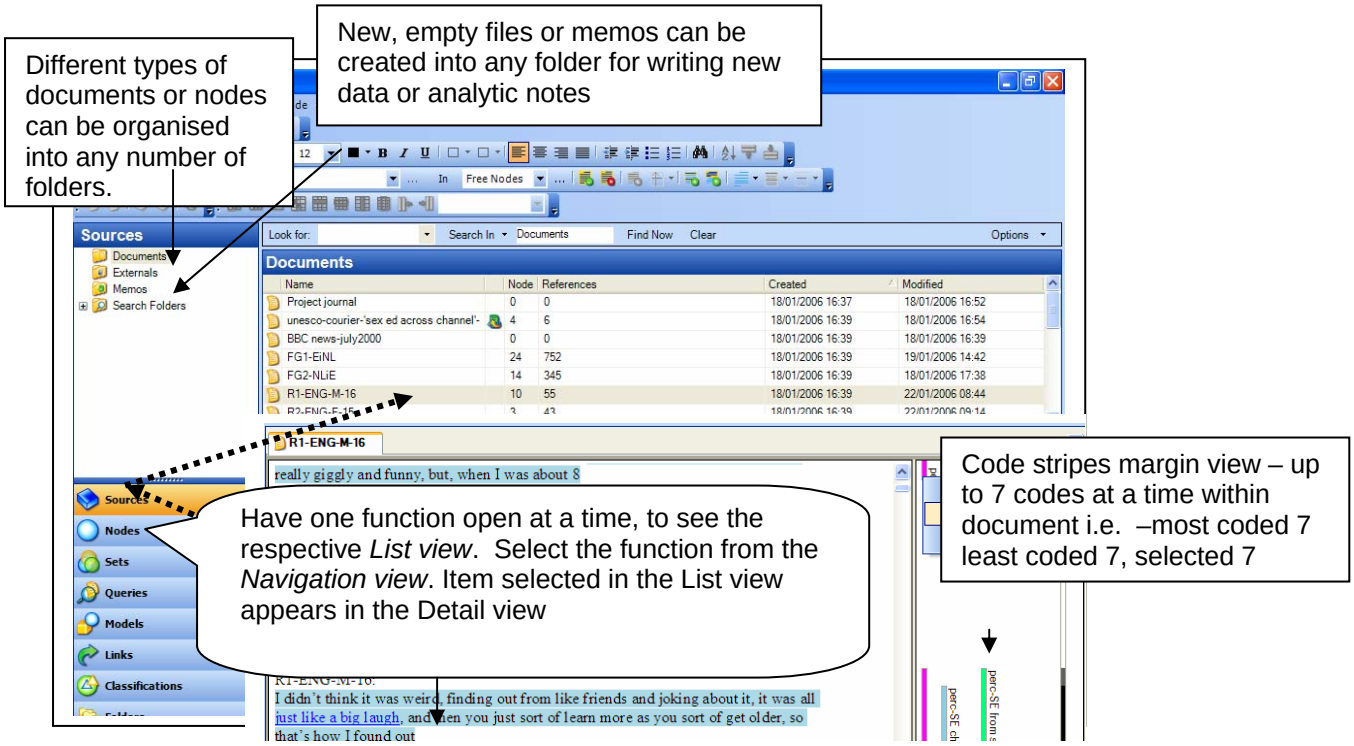
2GHz Pentium 4 Pentium processor or faster
512MB RAM or more
1024 x 768 screen resolution or higher
Windows XP™ Service Pack 2 or later
Approximately 1GB of available hard disk space
Internet connection
NVivo7 requires the MS SQL Server™ component in order to run (this comes with and is installed with the software).

Structure of work in NVivo7

NVivo7 functions using what we call an internal database system into which data files (sources) are imported. One project file holds all information making backing-up and moving the project easier than previous QSR software. The 'Workspace' provides access to the different elements of your work via Navigation buttons, and the interface can be customised in certain respects (e.g. where different panes appear etc.).

The main views in the workspace are the Navigation View (providing access to project folders e.g. sources, nodes, models, links etc); the List View (listing items within each folder); and the Detail View (where the content of an item is displayed) (see illustration below).

The user interface in NVivo7



Data types and format in NVivo7

NVivo7 can directly import and handle the following textual formats:
Text only, *Rich text format* (.rtf), or *MS Word™.*
Graphics can be embedded into *RTF* files.
There are 3 types of sources:
Documents : the source materials (e.g. transcripts).
Memos : to record thoughts
Externals : proxy files to represent files outside the NVivo7 database.
Sources can be given read only properties.

Closeness to data and interactivity in NVivo7

Sources are viewed in the Detail View and already opened sources are tabbed providing access without having to use the Navigation buttons.
Linking devices (see below) increase integration between project items. Sets are shortcuts to project items allowing any combination of items to be gathered together (e.g. individual documents, node, memos etc. can belong to the same set). Detail view panes can be undocked from the main interface to view alongside one another in separate windows.

Coding schema in NVivo7

The coding schema can be as hierarchical or un-hierarchical as required.
There are 5 types of node :
Free Nodes & Tree Nodes (for housing un-organised and hierarchical thematic codes)
Cases (for organising documents)
Relationships (for coding links between items)
Matrices (to save results of qualitative cross-tabulations – see below).

Nodes are visible in separate folders in the Nodes Folder View with the contents of each folder viewable in the corresponding List View. The All Nodes folder allows access to all current nodes, regardless of the type, but it is not possible to view the whole coding schema including hierarchical structure together in one place within the software.

Coding processes in NVivo7

Free and *Tree nodes* can be created up-front and can remain independent of text. Drag and Drop selected text onto existing nodes, but not vice versa. Short cut coding icons facilitate coding processes.

Basic retrieval of coded data in NVivo7

Retrieval by double clicking on a node to view coded data in the Detail View: lifts coded data out of context. Detail Views are live allowing recoding whilst in this view interactive connection to source context.

Coding Stripes allow up to 7 nodes to be viewed in a right hand margin. Choose according to for example, codes most coding item, codes least coding item, codes recently coding item, or manually choose codes. Coding stripes view also shows coding density – i.e. where the greatest number of nodes appear together at a point in the source. The coding stripes view has some interactivity : selecting a code highlights the relevant text, coded data can be accessed and segments uncoded.

Data organisation in NVivo7

The organisation of data happens by creating cases and applying attributes. Cases based on one document, several documents or parts of documents (e.g.focus group data) are all organized using the same case node structure (with attributes) seen in a unified case book table.
Known characteristics (attributes can only be applied to cases).

Case and attribute information can be imported from or exported to a spreadsheet application. Sources can be autocoded according to structure (headings or paragraphs).
Sets can be created at any time to group sources and any other project item in different ways.

Writing tools in NVivo7

A number writing tools allow writing in many different places.
Memo's: Blank documents can be created as memos at any time, or linked to an existing document or node. These are full ranking documents and therefore can also be coded. A memo can only be linked to one other source.
Annotations: comments can be embedded at any point in any document and re-opened when required. A list of annotations can be exported to a spreadsheet or word processing application.

Searching and interrogating the dataset in NVivo7

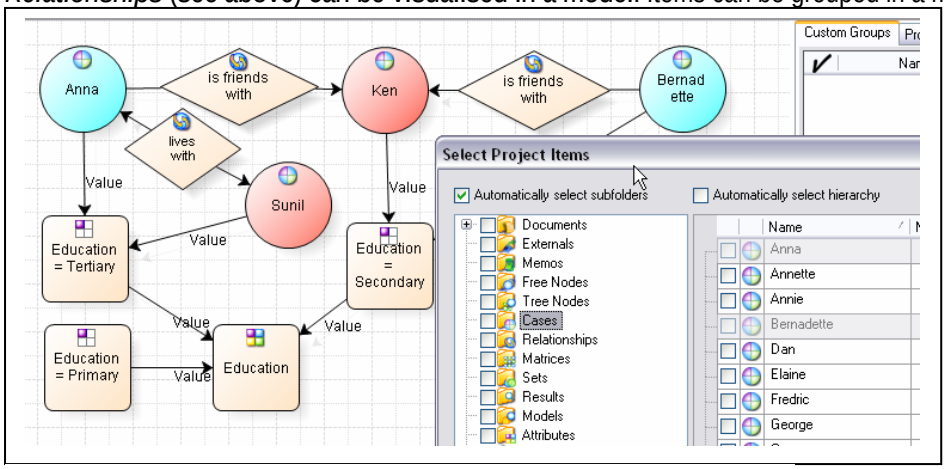
NVivo7 has four types of Query :
Text Search (for finding content in sources, nodes, sets or annotations).
Coding Query (simple queries find e.g. where cases with a particular attribute are coded at a particular node, advanced queries allow searching for position of nodes in the data (e.g. Boolean or Proximity operators).
Matrix Coding (compares pairs of project items and displays in a table).
Compound Query (search for specified text in or near coded data).
Queries can be saved to re-run later.

Linking devices in NVivo7

NVivo 7 includes five types of linking device:
Memo Links : A memo can be linked either a node, document or an external (but not to several).
See Also Links : text segments linked to other project items (including passages of text in other documents).
Annotations : Write comments about text segments.
Hyperlinks : link text segments to files or websites held outside NVivo7.
Relationships: a type of node which defines a connection between two project items. Different relationship types can be created, and the data which supports that relationship coded accordingly.
All linkages can be shown in the modeler tool.

Modelling in NVivo7

Models in NVivo7 can be either *static* or dynamic.
Items in *Static* models are not connected to any other project item.
Dynamic models provide *live* connection between project existing within the model, and the project items they represent.
Relationships (see above) can be visualised in a model. Items can be grouped in a model (acting as layers).



(Illustration taken from QSR tutorial material - Lyn Richards)

Output in NVivo7

There are several forms of output in NVivo7:
Export Project Items : sources, nodes, matrices and the casebook. It is not possible to export multiple items into the same file. It is not possible to print a document with the coding stripes on view.
Tabular Output any table can be exported to a spreadsheet application.
Create Summaries on individual aspects of the (Source summary, Node summary, Relationship summary, Attribute summary and Coding summary).
Coding Comparison report shows similarities and difference in how two sources have been coded.

Team-working in NVivo7

One or more projects can be imported into the 'master' project. All items or selected items can be imported.

CAQDAS Networking Project comment on NVivo7:

- *QSR's goal of normalisation is achieved to some extent with the use of the MS Outlook interface. As long as you are familiar with Outlook you should find it easy to find your way around the software.*
- *The Document folder structure provides a more obvious way to organise and list documents*
- *The best attempt of all the software in unifying and simplifying the organisation of data i.e. the application of attributes e.g. .socio-demographic variables/values to both whole documents and parts of documents e.g. different speakers in focus groups (providing formatting has been done correctly and the steps to coding cases are understood).*
- *There is not enough visual reassurance that a coding action has worked and this is a particularly important aspect for new users. This and other small aspects of usability will hopefully be improved.*
- *The interface means that you can only view one type of list at a time. i.e. either sources, or codes, or sets, or queries etc. Compared to other software packages reviewed here, this impacts on the sense of contact you have across different aspects of your work. The movement between different aspects of work is not as fluid as in other software. There are rather too many steps to achieve some simple and common functions.*
- *Text in the Detail view (i.e browsing pane) does not line wrap to fit size of pane –unusual in the current generation of software – this exacerbates the usual computer difficulty of being able to arrange windows and see enough of required material in each pane.*
- *The Codes stripes margin display is interactive. The restriction to viewing up to 7 nodes at a time in any one document limits the analytic use of the margin view and it is a significant downgrading of functionality compared to other software with an interactive margin. Even in NVivo 2, the margin display though static allowed the user to see all or selected sets of codes in the margin. Counteracting these limits, the additional coding density stripe is useful, particularly in the absence of being able to see how a document or node is coded in its entirety, since hovering over the coding density stripe will reveal all codes sitting at that approximate position in the data. An additional disadvantage is that the document with its margin display/coding density stripe cannot be printed out.. As yet users have been given no information about when this is likely to be fixed.*
- *Sets are flexible in NVvio7. The ability to add different project items to the same set is unique amongst software reviewed here. This provides useful ways to isolate parts of a project, or to group aspects together for many different reasons.*
- *The ability to create one new (e.g.broad brush code) out of selected codes just by an option on the right button menu is an excellent addition. This obviates the need for more difficult to do 'OR' types queries.*

However, simple to achieve output in one file of multiple codes with clearly labelled coded information above relevant data segments is missing and users frequently request this option in software.

- *The query tool is good but as with many complex dialogue boxes some steps are easy to miss. The range of outcomes from queries is very good. Matrix searches in particular offer flexible ways of conducting multiple searches at one time and accessing results. The ability to save the way queries are built, in a list, is useful.*

- *Linking tools are not as versatile as in other packages. While it is possible to hyperlink between points in the text, subsequently navigating around the dataset according to these links made is clunky.*

- *Currently users are experiencing some problems moving Nvivo2 and N6 projects into NVivo7*

QDA Miner V.1.3: distinguishing features and functions (and Wordstat: add-on content analysis module)

Normand Peladeau, a former program evaluator, believed that a software should be created to facilitate the analysis of both numerical and textual data, within one program. To that end he mixes graphical and statistical tools with the features characteristic of qualitative software. QDA Miner is the 'CAQDAS' qualitative tool, which can operate independently, but with the add-on Wordstat, to which we make reference (or Simstat –not included here), it extends the range of exploration and analytic techniques, to the more mathematical, *content analysis* of data. Potentially handling thousands of cases, the combined software spans the CAQDAS, Text–retriever and Text-based Manager categories of software. As such, the software is part of the new generation of qualitative software, crossing methodological boundaries and mixing methods. http://www.provalisresearch.com/

Minimum System Specification (recommended by developer)

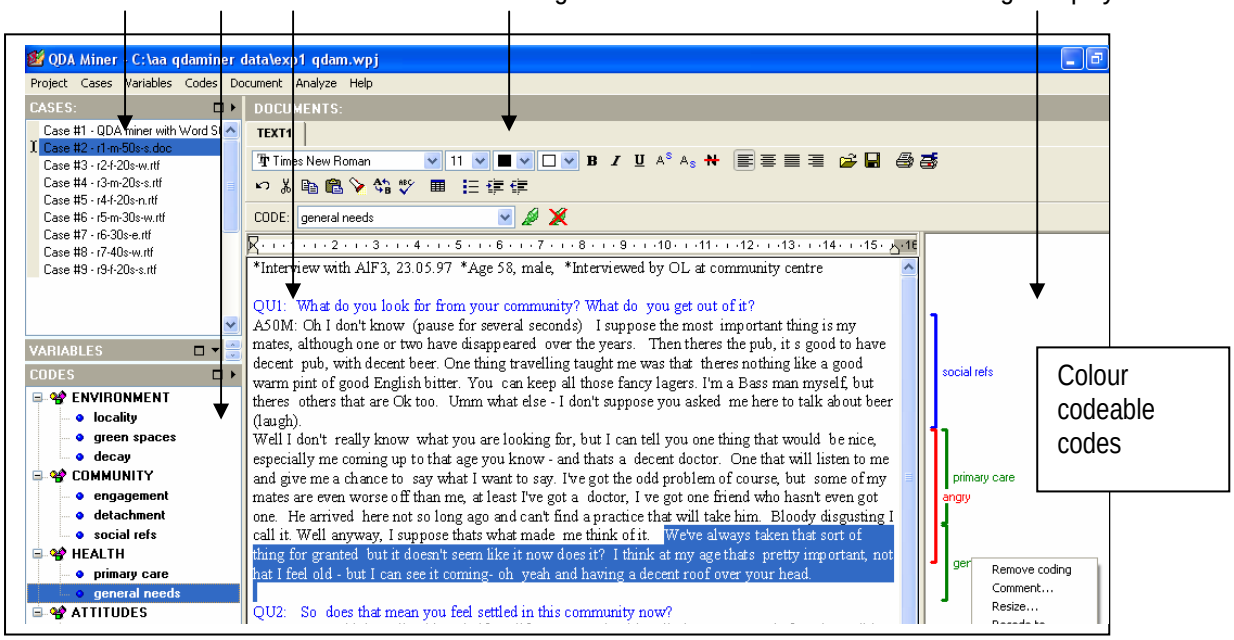
Windows 95, 98, 2000 or XP (or Mac with vitual PC)
128 Mb of memory: 9 Mb of hard disk space (36Mb when used with WordStat content analysis module)

Structure of work in QDA Miner 1.3.

QDA Miner uses an internal database structure (so documents are imported into a database). An archival feature, resulting in a compressed ZIP file, facilitates backing up and movement of projects (a project consists of many files, database, index, structure information files, project information files, etc). The compression is between 15 and 20% of the original file size.

Variables structure :The software is unusual amongst other CAQDAS packages in that it relies on the creation of variables at the outset, to handle both quantitative values about data, but also the types of data, and without them, textual documents cannot be assigned or created. However, using a *Document conversion wizard* to start a project off, automatically sets up the document type variables. Once the document variables have been created the qualitative parts of the dataset can then be transferred into the project. Thus the different document type 'variables' act like holders for different types of qualitative information; for instance the user can decide at any time to create a second document variable, 'notes'. This would automatically create e.g. an empty editable 'notes' file for each case. Other variables might be string, number, nominal, ordinal etc. These additional more quantitative variables may be used to hold respondent or case information and are also used for case filtering and conducting comparison. This information can be imported from a spreadsheet.

List of documents - Codes - Main text window - Editing bar Interactive margin display



Data types and format in QDA Miner 1.3.

QDA Miner stores documents using rich text format. It can import various file formats such as MS Word, WordPerfect, Rich Text, HTML, Adobe Acrobat, and plain text files. These files are converted by the *Document Conversion wizard* in the software, into the rich text format QDA Miner requires. The textual data is fully editable inside QDA Miner. The software will also import several database and spreadsheet file formats (MS Access, Excel, dBase, Paradox) and any data file with an ODBC driver (Oracle, MS SQL, etc.)
Multimedia support: A single document may include tables and graphic elements. Coding can be assigned to any chunk of text however small, to one or several table cells, or a whole graphic or another embedded object. However, coding of portion of images or segments of multimedia objects is not possible.

Closeness to data – Interactivity in QDA Miner 1.3.

Documents open with one click from listing in main window, see above screenshot.
The coded information in the margin area provides interactive contact with all codes applied to text and ability to remove codes from positions in the text.
One click on segment in coded retrieval window connects with source data and highlights segment within its source context.

Coding schema in QDA Miner 1.3.

Hierarchical: The coding schema consists of a two level hierarchy, a code has to be created under a top level category.
Drag and drop editing allow easy moving of codes within and between categories.
Colour of codes in margin display: User can assign a specific colour (any Windows colour) to any code. Alternatively, colours of codes in the margin may be associated with different coders, allowing easier comparison of codes assigned by different coders.

Coding processes in QDA Miner 1.3.

Double clicking on a code assigns it to the selected segment
Drag and drop: A code may be dragged and dropped to a selected text segment.
Paragraph coding without selection: Code a whole paragraph by dragging a code on to it (without the need to first select this paragraph).
Code margin display: Codes are displayed interactively in the right margin of the document. Clicking once highlights associated text. Clicking again on the same code or right clicking on it brings a pop-up menu to perform various operations – add or edit a comment, resize the selection, remove the coding, change the associated code to another one, etc.
Code List – listing codes, description and frequency across documents *and* cases

Basic Retrieval of coded data in QDA Miner 1.3.

Single to multiple coded retrieval: A window which provides options to choose by putting ticks against required codes, in order to get single or multiple coded retrieval.
Auto coding of the resultant segments can be built in to simple retrieval – (e.g. when multiple codes are requested) to attach new codes to existing text.
Code sequence retrieval tool may also be used to explore common code sequences. New codes can be assigned any group of retrieved codes and their coded segments, either selectively or globally to each segment in the result window.

Organisation of data in QDA Miner 1.3.

Documents are organized by cases. Cases are presented either as a simple list (showing documents associated with them,) or as a tree view where cases are grouped using values of up to two variables (for example, cases consisting of individuals may be grouped by gender and age). Cases can be filtered according to complex filtering conditions.
Socio-demographic variables or any other numerical, categorical, logical and date data that may be used to

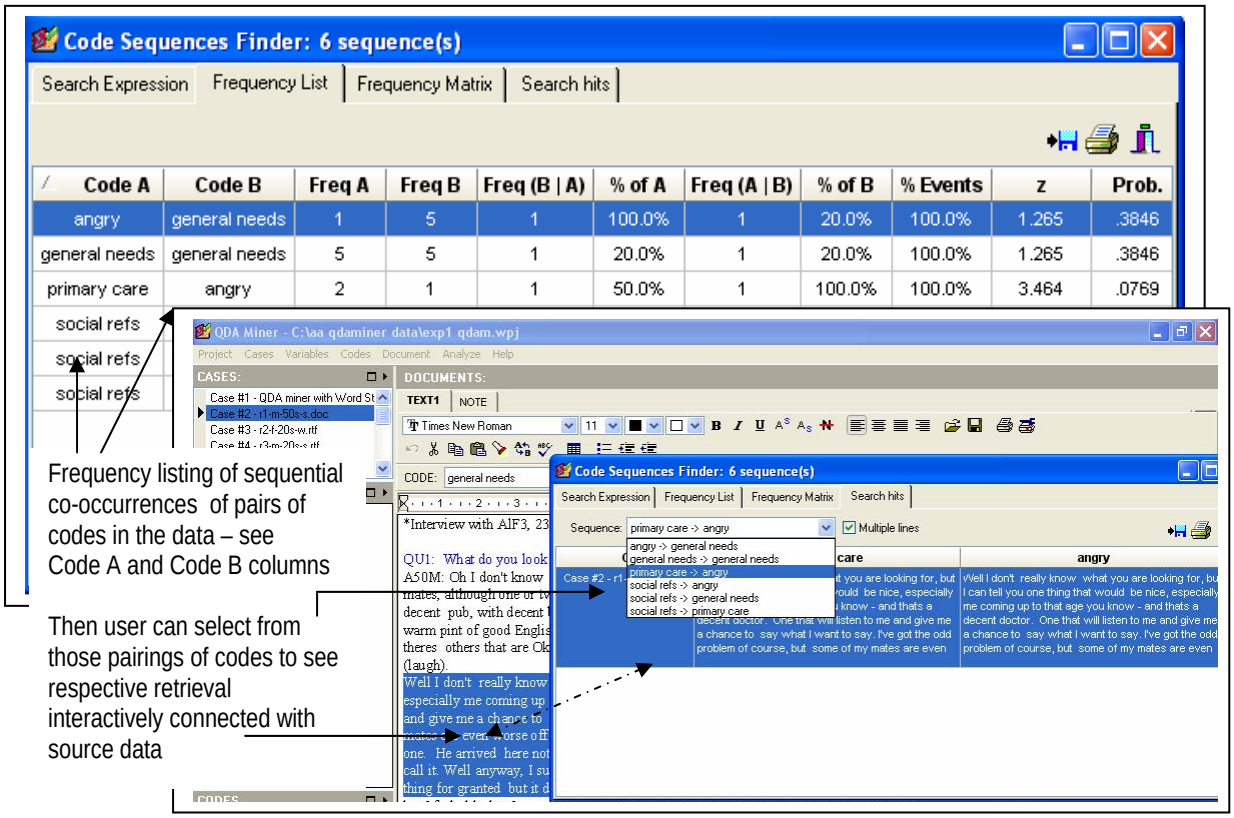
categorize cases are stored into variables and may be viewed or edited from the main window.
A project can contain up to 2030 variables, and theoretically could contain several million cases.

Writing tools in QDA Miner 1.3.

Annotations may be attached to codes in the code list, coded segments or at the code mark in the margin area. or to an entire project. To create case based memo files, one new *document* type variable may easily be added to the project. This creates an empty, editable memo file for each case. Such files are full rank documents which can be coded.

Searching and interrogating the database in QDA Miner 1.3.

Thesaurus facility a thesaurus search item may also be inserted, allowing the program to return a hit every time any one of the words or phrase associated with this thesaurus item is found.
Integrated text searching on coded data: Searching of text may be performed on different units of text -the whole document, a paragraph, a sentence or a previously coded text segment associated with a specific or one of several codes.
Searches include up to two logical expressions using the following operators (equal to, enclosing, included in, overlapping, followed by, preceded by, near) and join them by boolean operators (and, or, not). Include values of variables to compare finds across subsets
Code sequence retrieval tool may be used to explore common code sequences. User can ask for what codes follow one particular code or the search can be closely defined for both Codes A or B (or find out where *any* codes follow *any* codes) and how often (with distance between defined). A table listing the pairs of codes in the sequential co-occurrences provides frequency and location information. The user can choose whether to include overlaps of the pairs or not. Different views of the results allow you to see a matrix of sequential co-occurrences, or to choose interactive results (with retrieval) pane for a particular co-occurrence (auto coding possible from this), or a simple co-occurrence frequency table with case/document identification. See screen shot below



Section Retrieval features provides another method to search and retrieve documents, by looking for text delimiters and retrieving text between those delimiters. This feature is useful for automatically coding sections of structured documents, such as interviews (with symbols or strings to indicate the speaker) or fixed structure reports or text files (such as PsychInfo or MedLine database searches).
Auto coding: The list of hits is displayed in the text retrieval dialogue in a table format. Moving from one hit to the other automatically moves to the corresponding case and document and highlights the original text segment. (Note: the text retrieval form is a stay-on-top dialogue, allowing the user to move from the hit table to the main application. Manual coding may be done on selected hits or may be applied on the entire list of hits.)
Similar auto coding is also available for other types of searches (section retrieval, keyword retrieval & coding retrieval).
Auto coding of the hits or finds can also include the surrounding context defined by the user, e.g. the sentence, the paragraph, the coded segment or the whole document. A choice of icons give the user the chance to assign a new code to one selected segment or all the results in the window.

Output in QDA Miner in QDA Miner 1.3.

All tabular outputs may be printed or exported to Excel, HTML, Comma or tab-delimited files. Coding retrieval results may also be exported as a new QDA Miner project.
Documents and reports may be saved to disk in Rich Text, MS Word, ASCII or HTML format
Graphical displays in the software (cluster plots etc.,) may be copied to the clipboard and pasted into a word processor. They may also be exported to disk in BMP, WMF, PNG and JPG format
A whole project may be also be exported to several spreadsheet and database file formats such as Quattro Pro, Lotus, Excel, Paradox, dBase.

Team Working in QDA Miner 1.3.

Two strategies for team-working are available in QDA Miner.

Multi-user logon allows several coders to work on the same project (not concurrently, but at different times) with either full access to other users coding as well as QDA Miner editing and analysis features, or to only a limited number of features and no access to other coders work. Access to the program features is granted by an administrator.
OR
Several users may also work on different copies of a project (consisting of either identical copies of a project or different subsets of cases).
A merge feature can then be used to consolidate several project files into a master project, merging codebooks, new cases and new variables into a master project. Code assignments made to the same cases and documents are also merged into the proper document. The merge feature has an option to create a backup copy of the master project before proceeding.
A Coding Agreement dialogue allows a project manager to assess the reliability of codings made by several coders with the computation of agreement measures and the identification of common disagreements.

Ability to go beyond Code and Retrieve in QDA Miner 1.3.

Content analysis: The integration of the add-on module WordStat provides a comprehensive quantitative content analysis and text mining module with features such as (lemmatization, stemming, word frequency lists, stop lists, hierarchical categorization of words, word patterns and phrases, KWIC lists, links to lexical databases, vocabulary and phrase extractors, and many exploratory tools and graphics (hierarchical clustering, multidimensional scaling, proximity plots, heatmaps, correspondence analysis, etc.).

Comparison may also be made with any numerical or nominal variable. A similar type of quantitative content analysis may also be applied to coded segments, allowing the user to compare text associated with different codes. Application of various numerical and graphical tools to explore the co-occurrence of codes (hierarchical clustering, multidimensional scaling, proximity plots).
Analysis of case or document similarities. The same tools as those use for exploring code co-occurrences (see previous item) may be used to cluster cases based on their similarity of codings.

CAQDAS Networking Project Comment on QDA Miner 1.3.

- *There are some trade-offs between the sophisticated statistical functionality that started this stream of software and the qualitative coding module QDA Miner. For instance the coding schema has to be relatively simple with only two levels of hierarchy currently enabled (the number of levels to be increased shortly) Yet there are key, seemingly simple functions in QDA Miner which many people ask for, and do not get in some of the more sophisticated code based CAQDAS packages….*
  - *…user can change colours of codes as they appear in margin-based on meaning or based on differential colour coding for a team member*
  - *…one of the few software to allow double clicking to assign a code to selected text*
  - *…for single or multiple code retrieval. Just tick the codes wanted, and get clear code & document identified retrieval accordingly*

- *The in-house conversion of Adobe acrobat (pdf files), using the Document conversion wizard is a first amongst the here referenced CAQDAS software*

- *Thesaurus facility and the search category tool greatly extends the potential of the text search tool in catching meaningfully related words.*

- *The central 'variable-based' structure of the software makes this software difficult to get started. The next version to be released shortly makes this much easier. It may seem be intuitively built for a purely quantitative approach but can be unfamiliar terrain to users of other qualitative software.*

- *The level of interactivity between windows is very good.*

- *There are some dialogue boxes (e.g. in the otherwise excellent Document Conversion Wizard, and in some Content analysis functions) that need to give more direction and help. There are many different options and some functions remain rather hidden until you chance on them by selecting the right combination of options.*

- *The Code Sequence Analyser is an excellent tool –offering the user the ability to not only list co-occurring or overlapping codes in the data, but separated sequential paired occurrences of all codes with each other across the whole dataset.. With its interactive connections to source data, and its varied types of listing and tables containing this data, it is one of QDA Miner' s easiest tools, replacing what would be complicated searches in other software. It is impossible to overestimate the value of this tool.*

- *The export features are well organised and easy to create. When viewing the results table of a coded retrieval, the report is created at one click of an icon.*

- *The option to export coded segments, and in the process create a new project with them, is unusual and makes good data reduction sense when dealing with hundreds or thousands of documents or records. The exported code labels become case names, and the segments are contained within a document type variable for each case.*

- *As with all the developers whose software we review here, this developer is very receptive to constructive suggestions and is unusually quick at applying them in the software!*

- *The comments are based mainly on the interpretive-(QDA Miner) part of the software; please enquire further for other fuller, recent reviews of the content analysis features of Wordstat.*

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.

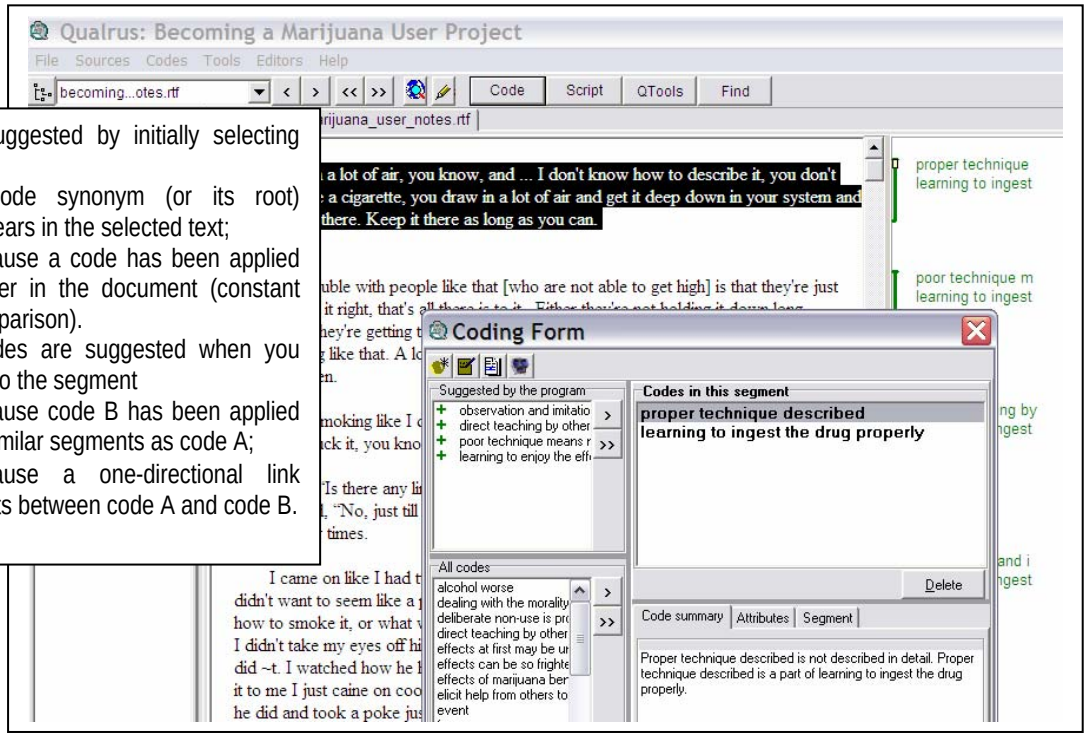
The segment is the basic unit of analysis in Qualrus which affects the way certain search tools function. Codes can only be applied to segments. There is complete flexibility concerning the size and number of segments created - segments can overlap, be embedded within one another etc., and as many codes 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.

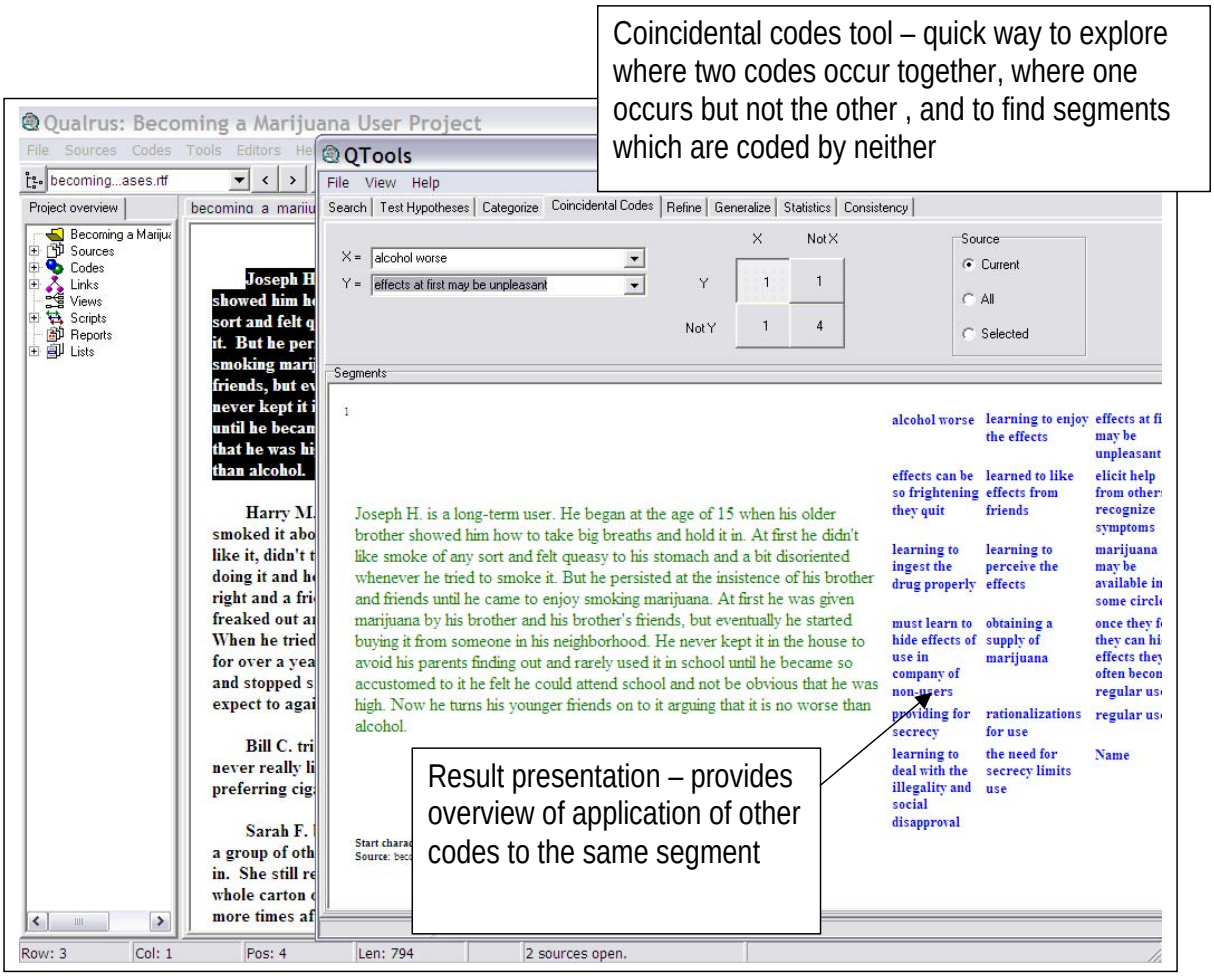
Searching and Interrogating the dataset in Qualrus

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 auto coding 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 and management 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.*

TRANSANA 2 - distinguishing features and functions

Transana is a FREE and OPENSOURCE software package designed to facilitate the transcription, management and analysis of digital **video or audio data.** It was originally created by Chris Fassnacht, and is now developed and maintained by David K. Woods at the Wisconsin Center for Education Research, University of Wisconsin-Madison, USA. www.transana.org

Note: Transana is different from the other software packages reviewed here because of its primary focus on audio/video analysis rather than text. For ease of comparison we have maintained the same structure in writing this review, but in places it has been necessary to add extra sections to reflect the nature and focus of Transana.

Minimum System Specifications (recommended by developer)

For Windows OS Versions 1.24 & lower and 2.02 and higher - MS Windows 98 or higher
Versions 2.00 & 2.01 – MS Windows 2000 or XP

For Macintosh OS Versions 2.00 and higher - MAC OS/X 10.3 or higher

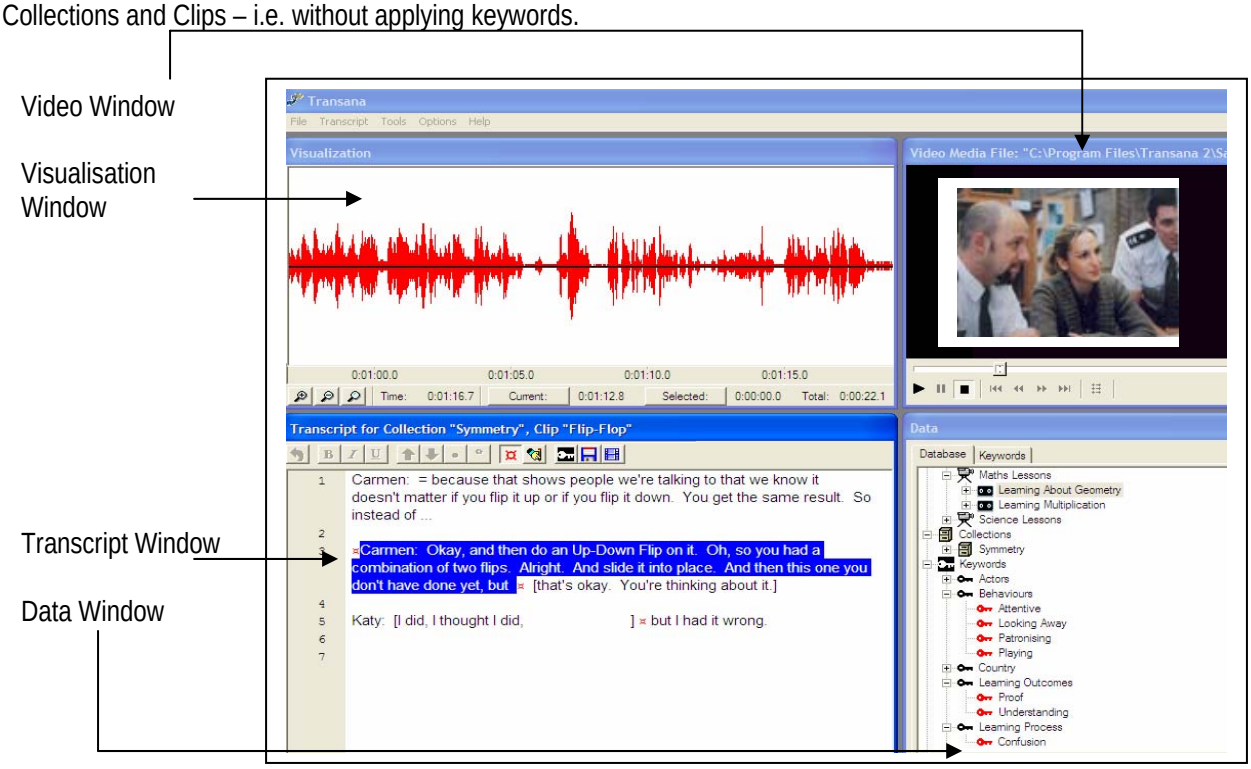
RAM 64Mb (minimum)
Minimum Screen Resolution of 1024x768 pixels
60-100 MB disk space for Program, 10MB space for Database – the amount and quality of video used is the critical factor relating to required (Transana will not work with video stored on CD or DVD)

Structure of work in Transana 2

Transana functions using an external database structure comprised of 3 components: the application itself, the audio/video files being worked with and the database which contains the transcripts, clips, keywords, collections etc. The database is therefore "the *project".* The audio/video files are therefore external to the program; it does not matter where the files are stored for Transana to refer to them. Audio/video files are not altered in any way during the analytic process.

The Transana user interface is comprised of 5 elements: the menu bar (which houses the main program controls); the Video window (displaying the video to be analysed); the Visualisation window (displaying the waveform of the video); the Transcript window (providing tools for transcribing the video); and the Data window (providing an overview of the way you have structured the data (termed the 'database tree').

The basic unit of analysis in Transana is the Clip – a portion of an Episode (a video or audio file) identified by the user as analytically interesting – which can be grouped into (thematic) Collections. Any Clip can belong to any number of Collections. Clips are independent segments of Episodes thus making it possible to work exclusively at the level of Collections and Clips – i.e. without applying keywords.



Data types and format in Transana 2

Transana can handle MPEG-1, MPEG-2 (which are the recommended video formats), most AVI video, MP3 and WAV audio. It can also work with some additional video formats which do not have a waveform diagram (see developer site for more details on different formats supported for Windows and Macintosh OS/X.) It includes database and file manipulation tools which facilitate the organization and storage of large collections of digitized video. It is not possible to create or import text documents which are not linked to a video or audio file.

Transcribing & Synchronising Audio/Video with written transcripts in Transana 2

The Transcription Mode facilitates the process of creating written transcripts for video or sound data and the auto-rewind function and play-back speed can be altered to suit individual users' requirements.
The waveform is the visual representation of the intensity of the source data's sound which, if required, Transana generates automatically upon creating an Episode.

Once the researcher has created the written transcript, the process of inserting time codes enables a particular point in the transcript to be linked to a frame in the corresponding video or audio file –subsequently allowing synchronised playback of video, sound and transcript, based, for example, on turn-taking in conversation. This allows working with audio/video, waveform and written transcript simultaneously. The position of Time Codes are user-defined and inserted manually (which can be done at any point).

Closeness to data and interactivity in Transana 2

The synchronisation of audio/video, waveform and written transcript results in the user being extremely close to all three forms of data simultaneously throughout the analytic process.
This interactivity between the transcript, video and waveform is superb and it is possible to control the video playback from any of the three windows: clicking on a particular point in one window takes you to the corresponding point in all 3 windows, enabling very precise pinpointing of aspects of data.
The windows are resizable to enable greater focus on transcript / video / audio / database tree depending on what is required at a particular point of the analytic process.
Having split audio/video into collections and clips it is always possible to view clips individually or collectively, or to view the clip in its Episode context. To alter the size of the clip however, requires deleting and recreating it.

Coding Schema – coding structures and organisation in Transana 2

As well as being organised into Collections, Clips can also be assigned (multiple) Keywords, which are grouped into Keyword Groups. The process and functionality of Keywords in Transana is essentially the same as coding in the other software reviewed here.
The keyword database tree is hierarchical in that keywords can only be created within an already existent Keyword Group. Keyword Groups can only have one level of sub-hierarchy (although a Keyword can belong to multiple Keyword Groups).

Coding Processes in Transana 2

A segment of audio/video/transcript can only be assigned a Keyword once it has been created as a clip.
Drag and drop text which comprises a clip onto Keyword(s) - or drag keywords onto clips in the database tree
Keywords can be applied to whole episodes, collections or individual clips.
A Clip can be assigned multiple Keywords, and appear in multiple collections.

Basic Retrieval of (Coded) Data in Transana 2

There are three ways in which to retrieve clips: either individually, by the Collection within which they are located, or by using the Search tool to find clips (regardless of collection) to which Keyword(s) have been applied. The first two ways are independent of the way the data has been "coded" (application of keywords).
Retrieval by Collection is a very quick and easy way to view all clips which have been grouped together sequentially

Summary Information is very easy to access, providing an overview of which Keywords have been applied to the clip currently in the transcription window. This view also allows you to edit or delete keywords from a clip or episode.

Instances of keywords applied to Episodes or Clips can be searched for based on the Boolean operators (AND, OR, NOT). The audio/video clips to which they have been applied can be reviewed and the results converted into collections. As such, video clips can be combined and recombined into new collections based on the results of keyword searches.

Organisation of data in Transana 2

Upon assignation to Transana, audio/video files and their associated transcripts are organised into Series and Episodes. A Series may have multiple Episodes (which are likely to be separate audio/video files)
Socio-demographic characteristics are handled at the Keyword level, and can be assigned to Series, Episodes and Clips – either upon their creation or subsequently.

Writing Tools in Transana 2

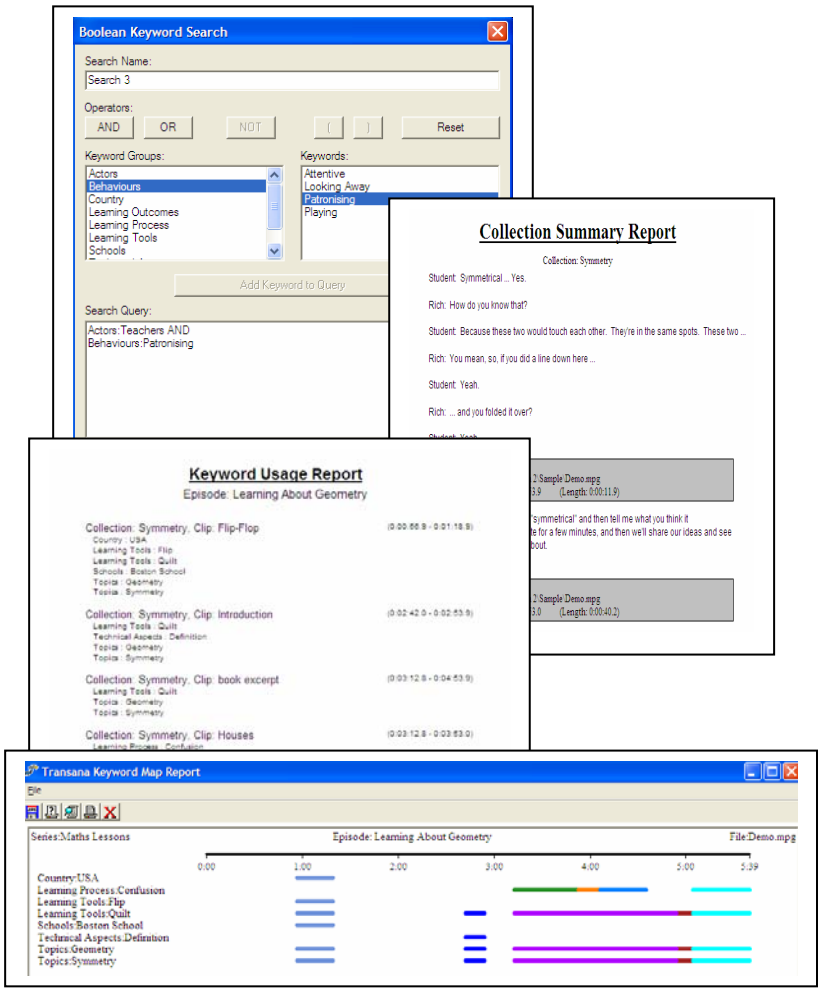
Transana's integrated writing tools consist of creating Notes which can be attached to Series, Episodes, Transcripts, Collections and Clips (but not to Keyword Groups or Keywords – to which definitions can be specified).

Searching / Interrogating the database in Transana 2

In addition to the basic Boolean search operators, is the **Keyword Map** which visually represents the sequential location of keywords as they have been applied to Clips within Episodes. Clips are colour coded in this view.

Output in Transana 2

Individual Clips, Collections or the results of searches can be played-back and reviewed within the software (not externally as Transana is not a video editing package). However, for demonstrational purposes different Presentation Modes are available to choose from which vary the relative size of the four windows (the video window, the transcript window, the data window, and the sound window).
Keyword Usage Reports can be executed on Series, Episodes and Collections, giving an overview of the occurrence of Clips (including timecode) and the keywords which are assigned to them.
Collection Summary Reports contain the written content of all the clips belonging to a Collection Reports can only be printed, not saved.



Teamworking in Transana 2

Transana 2.10-MU is a separate multi-user version of Transana. It has been developed specifically for collaborative projects. It allows multiple users to communicate and make changes to the Database (e.g. transcripts, clips, keywords etc.) *at the same time,* therefore facilitating real-time analytic collaboration between distributed teams working on the same network and between networks.

It is worth noting that it is possible to have both the single version and multi-user versions on the same computer. This is extremely useful if working on a single user project and a collaborative project.

Each project can be worked on independently or in real-time therefore there is no need to merge projects and the potential problems which can occur with this procedure. When working in real-time team members are able to communicate via the Transana Chat Window. In comparison to the single user version the chat facility is the only perceivable difference to the interface when using Transana 2.10-MU. The Transana Chat Window is a synchronous text-based tool. It can facilitate communication during analytic sessions and also as a writing tool, as it can provide a record of analytic thinking and process in a similar to memo keeping, for example it can be used as a log of the session, of discussions related to the analytic procedures, specific episodes, clips, collections or keywords. Video and/ or audio communication is not currently supported within Transana 2.10-MU itself, however if this is seen as necessary in addition to the text based tool, teams can choose from a number of products which can be used in conjunction with Transana 2.10-MU.

Users make changes to the database in the same way. One notable feature of Transana 2.10-MU is that changes made are not automatically tagged according to the user who made the change. However, teams may find ways of overcoming this by determining protocols for teamworking such as creating collections with their initials as the first letters of the title.

It should be noted that setting up Transana 2.10-MU is quite intensive and may seem a little daunting and time consuming. However, the website gives detailed instructions and more information may be found from local technical support teams and from the Transana forum.

Going beyond code and retrieve

Presentation Mode provides different options as to how video is presented during play-back. These include 'Video only', which displays the video window in full screen and 'Video and Transcript only' which displays the video and scrolling highlighting of synchronised transcript (if auto word-tracking is enabled).

CAQDAS Networking Project Comment on Transana 2 :

- *Comments about Transana must be viewed within the context of the fact that it is a FREE and OPEN SOURCE software, developed for video analysis rather than text. Researchers primarily using video data are highly recommended to consider this software carefully.*

- *It is a very easy to use tool for facilitating the transcription of video data or audio files. The ability to synchronise playback of video, sound and transcript and to control playback from any view (e.g. by selection in the waveform of particularly loud or soft periods) is very slick and analytically helpful.*

- *Interactivity varies in other respects, for example it is sometimes necessary to close one dialogue box before options in the main interface are available again, whereas in other instances there is good access from one dialogue box to a related tool.*

- *Transana currently does not easily enable the management and analysis of projects where textual data other than video transcripts are being utilised as a primary source of data.*

- *The limited range of integrated writing tools provided by Transana is probably its greatest weakness; in the sense that it is not possible to link a particular note to multiple clips, for example, or to pull out all notes, wherever they are located in the database.*

- *The structure of the database tree system is a little confusing at the outset as the organisation of Clips into Collections can play a very similar function as the organisation of keywords into Keyword Groups. In addition, each clip, collection, keyword group etc can have its own ID – so you need to be clear about the difference in the way you will use IDs and groups of other objects (collections, keywords etc.) in order not to become confused.*

- *In comparison to the other software packages reviewed here the Search Tool is not very sophisticated.*

- *As all the developers of the software reviewed here, this developer is very responsive to the feedback and needs of Transana users. The website provides access to the development team's log where they post the latest information concerning new features and versions. In addition, the open source nature of Transana means that not only is the software freely distributed, but the source code (and relevant documentation) can be obtained by anyone. Therefore Transana can potentially be customized to suit the particular needs of a given project.*

- *The ability of the multi-user version to allow geographically dispersed researchers to work on one database simultaneously is unique amongst the software reviewed here and can significantly facilitate collaborative teamworking.*

Coming next…

…..InfoRapid…..
…non code-and-retrieve based software – Storyspace