Complete Citation: Lamont-Mills, Andrea (2004). Computer-aided qualitative research: A NUD*IST 6 approach. In Herbert Haag (Ed.), *Research methodology for sport and exercise science: a comprehensive introduction for study and research* (pp. 288-298). Reinhein, Germany: Hofmann. ISBN 3-7780-3418-9.

Accessed from USQ ePrints http://eprints.usq.edu.au

Computer aided qualitative research - A NUD*IST 6 approach.

Andrea Lamont-Mills, Department of Psychology, University of Southern Queensland, Toowoomba, Australia

Over the past decade there has been a rapid growth in the development and use of computer programs to assist in the analysis of qualitative data. Computer programs can be used by qualitative researchers to perform a variety of analytical tasks, from simple editing functions through to complex theory development (Fielding, 1998; Gibbs, 2002; Miles & Huberman, 1994; Weitzman & Miles, 1995). Given this popularity and the diverse usage of computer programs, it is helpful to conceptualise programs as belonging to one of five broad categories based on analytical function. These categories listed in increasing levels of sophistication are text retrievers, textbase managers, code-and-retrieve programs, code-based theory-builders, and conceptual network-builders[1]. Thus each of these programs differ in the amount of analytical assistance it renders to the researcher.

At the lowest level of analytical assistance sits **text retrievers**. These programs search and retrieve words, phrases, and other string characters that have been determined by the researcher to be of analytic interest, be this inductively or deductively determined. To illustrate using my research interests in gender and discourse, I could search and retrieve

all instances of the word *feminine* across each interview text file. In most text retrieval programs the raw data (e.g., interview file) is not entered into the program but stands outside the program where it is searched by the program. **Textbase managers** organise the raw data within the textbase manager program. That is, these programs store raw data and can systematically sort the database into meaningful subsets for comparison and contrast by the researcher. For instance, I could search and retrieve all instances of the word *feminine* across male athlete responses and across female athlete responses. **Code-and-retrieve programs** are an advance on the previous two categories in that they enable the researcher to retrieve and code the data. That is, lines, sentences or paragraphs can be coded on the basis of keywords. For example, I could search and retrieve all instances of the keywords *feminine, girly girl, girl stuff, wearing make-up,* and label all located items with the code feminine. The previous two program categories did not allow coding to be conducted at the same time as searching and retrieving nor did they allow for coding to be stored within the program itself.

Code-based theory-builders not only retrieve-and-code but also assist the researcher to develop and test theory. Here categories can be developed from the assigned codes, memo's can be written and linked to these codes and categories, and hypotheses that have been induced from the data can be formulated and tested. For example, I could search and retrieve all instances for keywords *feminine, girly girl, girl stuff, wearing make-up,* and label all located items with the code feminine. In subsequent analysis the code feminine could be grouped with the code masculine into the category "gender stereotypes". Further analysis of the gender stereotypes category could result in this being subsumed under the higher order theme "gender". The development and defining qualities of the two codes

and the two categories would be reported in a memo within the program that is attached to the category “gender stereotypes”. Indeed memos can be written for and stored for each search and each code. From this I may then hypothesis that the data appears to suggest that male athletes construct gender stereotypes differently than female athletes. This hypothesis could be tested via a search-retrieve-and-comparison of male and female responses. The last category is **conceptual network-builders.** These programs use semantically meaningful networks to build and test theory. Further, the researcher’s thinking and conceptualisation of the data can be represented graphically in these programs.

The aim of this chapter is to demonstrate the logic underpinning one particular code-based theory-builder, NUD*IST 6 which is produced by QSR International Pty Ltd. NUD*IST stands for Non-numerical Unstructured Data Indexing Searching and Theorising. The choice to discuss NUD*IST 6 is driven by my familiarity with the program. I used NUD*IST 6 in my Doctoral dissertation and extracts from this research will be used to illustrate points made during this chapter. The chapter will begin with a brief discussion regarding computer-aided analysis in qualitative research. A brief outline of what this chapter will not cover will then be presented. The main body of the chapter illustrates the logic and flexibility of the NUD*IST 6 program by contrasting the two interlocking data systems associated with the program, the document system and the index system. The chapter concludes with some limitations of the NUD*IST 6 program.

A word of warning – Computers and qualitative data analysis

The phrase "computer assisted qualitative data analysis" is somewhat of a misnomer in that qualitative analysis computer programs do not follow the same analytical principles as quantitative analysis programs (e.g., SPSS, SAS, BMDP, etc.). This is an important point for the reader to understand. Those readers who are familiar with quantitative analysis programs would understand that after raw data is entered into a program such as SPSS, the researcher proceeds with a particular analysis of interest where the program performs the actual analysis.

In this sense quantitative analytical programs compute the variance in the data and then determine whether this variance is of some importance. That is, whether there are differences in the variance between groups, between pre and post treatment conditions, and so forth. Quantitative data analysis programs tell the researcher whether the measured variance is important or of some value based on probability levels, effect size, and power. For example, I am able to say with 95% confidence that elite male athletes and elite female athletes do not differ in their responses to the Personal Attributes Questionnaire (PAQ: Spence & Helmreich, 1978) because the F value was not significant. Probability levels, effects size, power, and so forth are well-established parameters or standards of importance in quantitative statistics[2].

Computer programs for qualitative analysis, and in particular code-based theory-builders, do not analyse the data in the same manner as quantitative programs. Indeed qualitative programs do not analyse per se. Rather qualitative programs **assist** analysis by making information more accessible to the researcher. Thus they **do not** perform the analysis itself. Qualitative programs can systematically and logically search and organise

the data, and this searching and organisation may lead to the highlighting of an area of potential importance for the researcher. However programs will not tell the researcher what to search nor will they tell the researcher whether this is important or of some value. Qualitative programs potentially enhance the researcher's thinking about complex data by logically and systematically structuring and storing information in ways that facilitate the discovery and exploration of meaning in unstructured data. Here the use of the word potentially is a deliberate choice. Qualitative analysis programs only potentially enhance the researcher's thinking and conceptualisation of the data. Any program is only as good as the researcher using it, as is true of quantitative programs. The qualitative researcher, therefore, still needs to engage in reflexivity, critical thinking about the data, rigorous methodological practices, and so forth. Qualitative computer programs may make the data more manageable but they **do not** automatically make sense of the data, they **do not** automatically make the research more rigorous, and thus more methodologically and theoretically sound.

What this chapter will not cover

This chapter is not a users guide to the NUD*IST 6 program nor will it go into any detailed technical description of the program. For those readers interested in these aspects the QSR web site http://www.qsr.com.au/ and the User guide that accompanies NUD*IST 6 are the best places to find this information. Further, this chapter will not debate the advantages of the NUD*IST 6 program over other qualitative data analysis computer programs. For this the reader is directed to the Weitzman and Miles (1995) book. This chapter, therefore, focuses on the logical structure of the NUD*IST 6 program

and how this structure can potentially enhance analysis of qualitative data in the sport domain.

NUD*IST 6: The logic and the structure

As mentioned previously, NUD*IST 6 is a code-based theory-builder program that stores, searches, retrieves, codes, and aids in the analysis of qualitative data. The program allows the researcher to manage and explore the data in two interlocking systems, the document system and the index system. The **document system** (called document explorer in Figure 1) is the holding system for the raw data. Raw data or documents can be explored and coded within this system. By doing this, the documents are linked to codes or categories that reside in the index system. Thus no codes are stored in the document system. The **index system** (called the node explorer in Figure 1) is, therefore, where the codes, categories or themes are organised. Note that NUD*IST 6 refers to nodes not codes. Nodes contain the coded data as well as the researcher's thinking about the data as a memo attached to the node. These nodes can be explored and further coding done on them. By doing this, nodes in the index system are automatically linked to the documents that reside in the document system (see Figure 1 for the document system and index system as they appear in NUD*IST 6.).

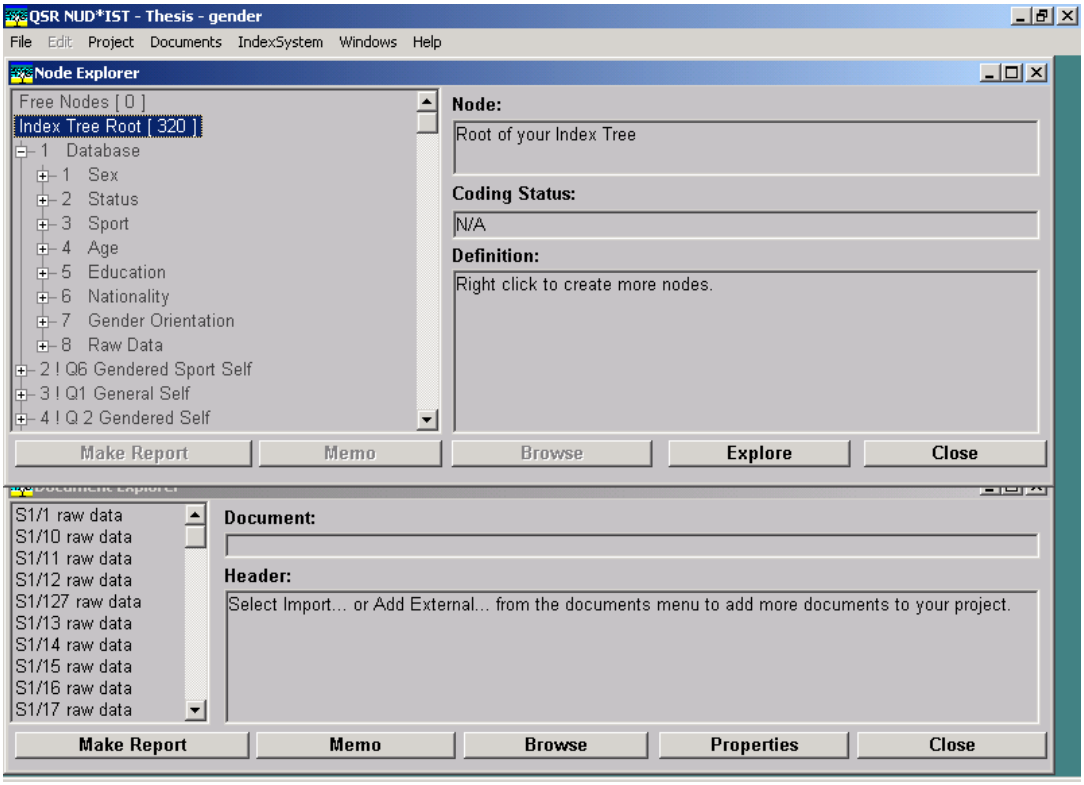

Figure1. Index and document systems in NUD*IST 6.

The document system: How to manage mountains of raw data

The document system allows the researcher to collect and organise the raw data, study and explore the raw data, develop ideas about it, edit and annotate the raw data, make notes and memos about raw data, and search and retrieve words, phrases from the raw data. It is the researcher's entry point to all the raw data in the research project. For example, in my research the 75 transcribed interviews conducted with elite athletes and coaches were held in the document system. The document system thus allows the researcher to work on the raw data, code it, and think about it. Exploration in the document system organises the documents first, nodes second. The advantage of the

document system is that it codes, retrieves, and browses raw data more thoroughly, more rigorously, and faster than can be done manually. Thus repetitious, factual, and descriptive coding can be efficiently handled by the document system.

As previously mentioned, coding results are not stored in the document system. They are stored in the index system. Through interlocking with the index system, exploration and interpretation of the data becomes a continuous process. Coding that is both inductive (codes that emerge from the data) and deductive (codes that are imposed on the data) can be efficiently handled by the system.

Different researchers use the document system in various ways. A researcher can work predominantly within the document system where individual documents are explored, and the text contained in each document coded with the results of each code being placed in the index system. Here the researcher works on the raw data within the document system. Working in this way, however, limits the researcher to basic word or phrase searches on the documents. This is analogous to working with different coloured highlighter pens when coding the original transcript. It does not allow for comparison between documents or between codes. For this the researcher needs to move into the index system.

The index system: Thinking about mountains of raw data

Other researchers, including myself, prefer to work primarily within the index system, and treat the document system mainly as a raw data storage area. In this way the document system is used for an initial search, code, and retrieval of factual data (e.g., male athlete responses, male coach responses). The results of this initial search are placed

at the database node in the index system (see Figure 2 for an example of a database node). This approach requires some forethought by the researcher as to what should be included in the database node and is determined by the research question, the background, and interests of the researcher. A node in NUD*IST 6 may be set up so that it has two or more sub nodes or children that sit under the one node. Remember that nodes hold all the researchers thinking about the node and can include lower order concepts within the one node. To illustrate, I set up a database node that separated the raw data into a *sex* node, which had two children nodes or sub nodes – "males" and "females". Further, I set up a *status* node, which had two children nodes – "coaches" and "athletes". What this did was store all the male interviews within the database child node "male", and all the coaches' interviews are stored within the database child node "coaches". I also set up a separate responses node that contained the responses to each question in the interview[3]. All the responses to question one are contained in the child of this node. The advantage of this is that it allows the researcher to automatically search and more easily compare responses across different groups. For example, I could easily compare male athlete responses to male coach responses at question 1. I could not do this as readily in the document system. Coding can then be done within the index system by searching and retrieving raw data that has been stored at one of the branches of the database node or the response node. The result of this coding is stored as a node in the index system as happened with coding in the document system.

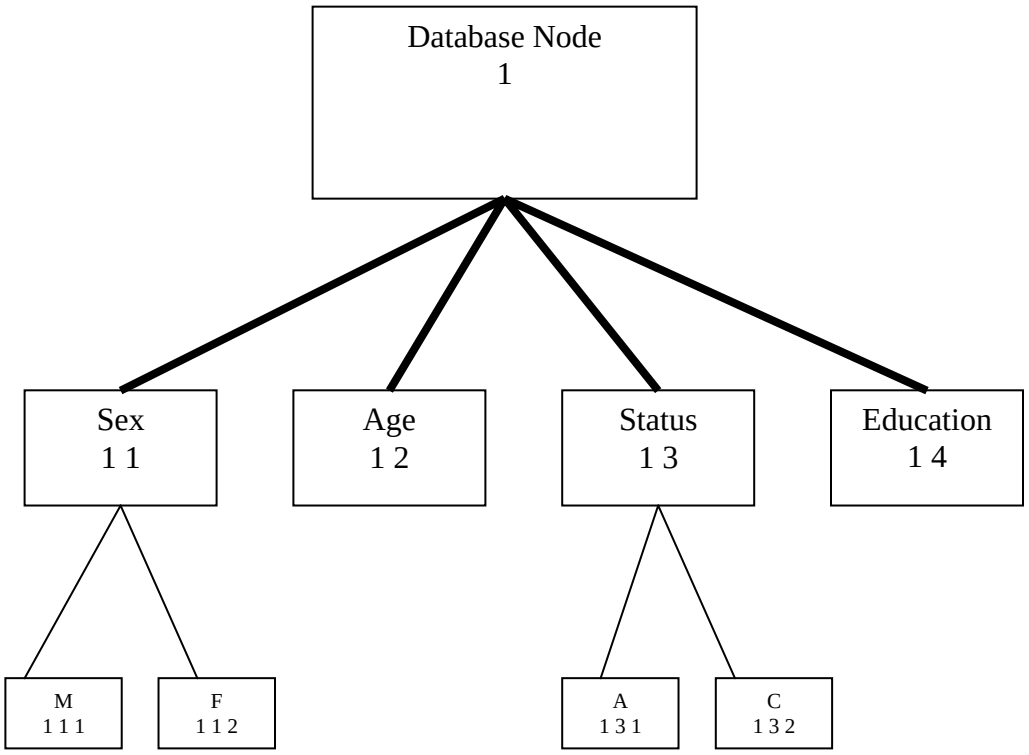

Figure 2. Part of the database node in my gender and gender identity project. Sex and status children displayed (M=male 1 1 1, F=female 1 2 1, A=athlete 1 3 1, C=coach 1 3 2). The numbers refer to the position that the node and children of the node have in the research projects overall index tree.

The index system forces the researcher to think about relationships between nodes or concepts and is often thought of as the thinking system. It manages ideas and exploration of the ideas that emerge from coding. Exploration organises codes first and documents second. It stores and locates codes, categories, higher order concepts, and associated ideas all within a node. It aids the researcher by helping to structure the codes and ideas

that have emerged from, or been imposed, on the data into a hierarchical tree structure. The index system takes a top down approach to data organisation, which contrasts with the traditionalist bottom up approach to qualitative coding.

There are two types of nodes that can be used in NUD*IST 6. Free nodes contain information pertaining to a specific subject (e.g., physical free node - contains all references to physical attributes). The researcher may begin with free nodes and then structure them into the hierarchical index tree. The index tree contains information from numerous sub nodes in separate sections of the tree (Figure 3 is an example of an index tree). The index tree allows for categories, themes, and higher order themes to be developed and organised, and thus relationships to be represented. The index tree is a hierarchical coding structure that helps the researcher develop thinking about relationships between categories. For example, in my research one of the index tree nodes was *Categorisation* (see Figure 3) and contained three child nodes: *Particularisation*, *Particularisation and Categorisation*, and *Categorisation*. All coding related to the differing categorisation strategies was contained at the *Categorisation* child node, all coding related to differing particularisation strategies was contained at the

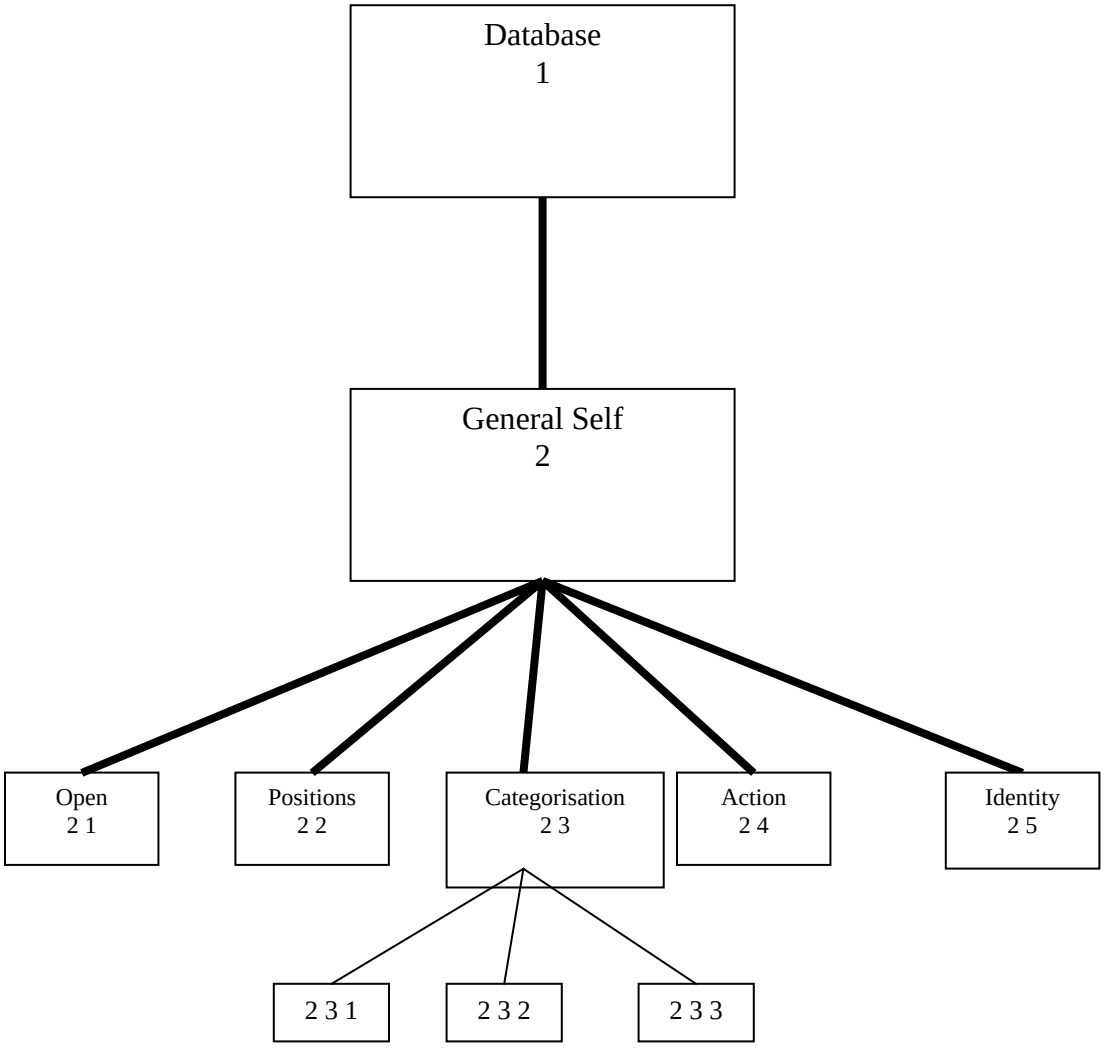

Figure 3. Part of the index tree in my gender and gender identity project. General self branch including Categorisation sub branch and children is displayed (2 3 1 = Particularisation, 2 3 2 = Categorisation and Particularisation, 2 3 3 = Categorisation). The numbers refer to the position that the node and children of the node have in the research projects overall index tree.

Particularisation child node, and so forth. *Particularisation*, *Particularisation and Categorisation*, and *Categorisation* are all examples of the categorisation process (Billig, 1996) and thus were organised within the *Categorisation* node.

Inductive and deductive types of analysis are also possible in the index system. I used both an inductive coding scheme where I followed a discursive psychological theoretical approach to the interviews and a deductive coding scheme where the items from the Personal Attributes Questionnaire (Spence & Helmreich, 1978) were imposed upon the raw data.

As mentioned in the document system section, results of text searches within the document system are held in the index system, in the text searches area. These can become nodes in their own right within the index system. Conversely searches of coding held at nodes are held in the index search area, and can also become nodes, as with the text searches. The index system, therefore, allows for greater types of searching to be conducted. Boolean searches such as collation searches, contextual, negation, restriction, and tree-structured can be easily and efficiently conducted (see Weitzman & Miles, 1995 for a discussion of each type of search operator).

Hence the index system allows for the researcher to logically and coherently structure his or her thinking so that theory building is enhanced. Memoing allows for the researcher to track thinking associated with nodes and child nodes such as how these where developed, defined, decided upon, and other related information. The index system itself allows the researcher to quickly and easily edit, modify, and build an index tree that represents emerging relationships amongst nodes. Further, the ability of the index system

to use searches as data sources allows for system closure. All of these can potentially enhance theory development and building by organising data logically and coherently.

Therefore, both the document system and the index system have a logical and coherent structure for organising, managing, and thinking about the data. Both allow for flexibility within their structure. To illustrate further the following section is a brief overview of my own research project as it was set up within NUD*IST 6.

An example – The construction of gender and gender identity in elite Australian sport.

Figure 1 shows the document system and index system as they appeared in NUD*IST 6. This research explored gender and gender identity construction in elite Australian sport using an inductive discursive psychology theoretical and methodological framework and a deductive content analysis. Of analytic interest were how elite sportsmen and elite sportswomen did being male and female, and what it meant to be male and female in elite Australian sport. In the index system there were 223 nodes in total. These nodes included inductive and deductive coding as discussed above.

Thirty-seven elite Australian athletes and 38 elite Australian coaches were interviewed using a semi-structured format. Each transcribed interview, that is the raw data, was stored within the document system in NUD*IST 6. Thus Raw Data 1 (see Figure 1) held the transcribed interview from participant 1 and so on. Each document also had attached to it a memo. I set up a memo for each transcribed interview that held my field notes associated with each interview.

As mentioned previously I like to work predominantly within the index system. Thus in this case the database node was the first section of the research project that was set up (see Figure 2). All basic demographic information was organised as a child node[4]. All the participants' responses to question 1 were put into the *response* node in the child node question 1. This allowed me to more easily search just responses to question 1 both inductively and deductively. All of these nodes were housed within the index system.

Analytic work was then carried out by browsing the question 1 child node in the index system and applying codes to the data. Post-doctoral research has also subjected the data to a modified version of a grounded theory (Willig, 2001). This will be used to illustrate the workings of my coding and thinking as it is perhaps more familiar to the reader than discursive psychology. In the grounded theory open codes were allowed to emerge from the data. These codes became free nodes in the first instance. After numerous passes through the data it was felt that some free nodes appeared to be related to a common theme. Participants described themselves in reference to social or vocation roles – *etic* free node, and also with reference to informal or culturally specific roles – *emic* free nodes. It was felt that the underlying theme associated with these two free codes was reference to a role. Hence the two free codes were re-organised under a newly created *role* node in the grounded theory node section of the index tree. This cannot be seen in Figure 1 due to the large number of nodes in the project. It was also felt that some nodes were being discussed in particular relationships with other nodes. I hypothesised that when participants described themselves in terms of attributes, traits, behaviours they often used roles to exemplify these traits, attributes, and/or behaviours. For example " I think I'm very understanding, like when I'm with my kids I really listen to what they're

saying". Hence a Boolean collation search was conducted. Thus it can be seen how theory building can be aided by the logical structure of NUD*IST 6 where free nodes can be moved into index trees and sub divided into child nodes.

Some limitations of NUD*IST 6

NUD*IST 6 is a logically structured program that coherently organises qualitative data in a manner that facilitates the researchers' thinking about the data. Its logical structure helps the researcher make sense of the data. There are some limitations with the program, which are primarily technical in nature, such as the inability to switch between text units of a line to text units of a paragraph. In addition, one of the most detracting features of the program is that it is not able to graphically represent relationships between concepts. However QSR has developed a program that accounts for NUD*IST 6's shortcomings, this program is called NVivo. NVivo is a PC only program and allows the researcher to work with text units of differing sizes such as individual letters to whole pages, to single words all within the same node and project. Further, NVivo allows the researcher to graphically represent his or her thinking through a multi-layer graphical modeller. This modeller is linked to the data so that the researcher is easily able to explore relationships in the data and add any items from the data directly to the model. For further information about NVivo an overview can be found at the following website http://www.qsr.com.au/. However NUD*IST 6 is able to handle larger amounts of data and allows for easier automatic searching and coding of data than NVivo at the present.

This chapter has aimed to provide the reader with a brief overview of the inherent logic subsumed within NUD*IST 6. What stands NUD*IST 6 apart from other programs and a manual approach to qualitative data analysis is the logical, coherent, and flexible

nature of the NUD*IST 6 framework. However like all programs, be these qualitative or quantitative, NUD*IST 6 will only be as good as the researcher who uses it.

Footnotes

1. See Weitzman & Miles (1995) for a comprehensive overview and evaluation of each grouping and commonly used programs within each grouping.
2. The author notes the current debate between significance levels and effect size.
3. Note that this could have been included under the database node.
4. I could have subdivided the data differently, so that under sex the breakdown was female athlete, female coach, male athlete, and male coach. However this would not have enabled me to the compare athletes to coaches' responses as a whole or female to male responses as a whole. The former division is more flexible in terms of greater search breakdown.

Acknowledgments: The author would like to thank Steve Christensen for his helpful comments on an earlier draft of this chapter.

Address for correspondence: Department of Psychology, University of Southern Queensland, PO Darling Heights, Toowoomba, Qld, 4350, Australia. Email – lamontm@usq.edu.au

References

Billig, M. (1996). *Arguing and thinking* (2nd ed.). Cambridge, UK: Cambridge University Press.

Fielding, N. (1998). *Computer analysis and qualitative research*. London: SAGE.

Gibbs, G. (2002). *Qualitative data analysis: Explorations in NVivo*. Buckingham, UK: Open University.

Miles, M. B., & Huberman, A. M. (1994). *An expanded sourcebook: Qualitative data analysis*. Thousand Oakes, London: SAGE Publications.

Spence, J. T., & Helmreich, R. L. (1978). *Masculinity and femininity: Their psychological dimensions, correlates, and antecedents*. Austin: University of Texas Press.

Weitzman, E., & Miles, M. B. (1995). *Computer programs for qualitative data analysis*. Thousand Oaks, CA: SAGE.

Willig, C. (2001). *Introducing qualitative research in psychology*. Buckingham, UK: Open Press.