software review

QSR NUD*IST and **ATLAS/ti**
QSR NUD*IST (Non-numerical Unstructured Data Indexing Searching and Theorizing), v. 4.0
Qualitative Solutions and Research, Australia
Distributed in the US by Scolari / Sage Publications Software
For PC and Macintosh platforms
www.scolari.com / nudist / nudist.htm

ATLAS / ti, v. 4.1
Thomas Muhr, Scientific Software Development
Distributed in the US by Scolari / Sage Publications Software
For the PC platform only
www.atlasti.de

Reviewed by
Jodi Vender and Trudy Suchan
The Pennsylvania State University

Qualitative data are non-numeric (text, image, or sound) so have different problems of data preparation, exploration, aggregation, and reporting than numeric data. We review two software packages that assist the researcher in managing qualitative-data documents—assist in coding categories and ideas, and in constructing and testing theories about the data. While QSR NUD*IST and ATLAS / ti perform the same basic functions, they are designed around different models. The best-known qualitative data software, NUD*IST, imposes a hierarchical organization on data and analysis. ATLAS / ti employs a hypertext model for data organization and exploration.

Both NUD*IST and ATLAS / ti allow the researcher to investigate text (reports, interview transcripts, survey responses, meeting minutes, field notes, newspaper clippings) or non-text (maps, charts, photographs, voice recordings, films). Our experience is with interview transcripts, so we emphasize that form of data in this review. The focus-group transcripts in Olson, et al's study of a map (this issue) could be analyzed similarly with one of these packages. Both NUD*IST and ATLAS / ti also could be used to analyze maps. Each map in a series could be a data document coded with categories of information, or smaller units of analysis can be delineated within the larger image. Gluck's cartographic images in corporate annual reports (this issue) could thus be analyzed using these packages.

NUD*IST can manage both imported documents (files whose text is stored within the program's database in ASCII format) and external documents (documents that exist outside the database, such as maps and graphics, that cannot be saved as ASCII text). For all documents one can create codes referencing text, search coding patterns, write and edit memos about the data, and generate summary-level statistics. Imported documents can also be edited, annotated, searched, and text itself retrieved (external documents are essentially bitmapped images). ATLAS / ti allows the same basic data-management functions but facilitates a different coding style. NUD*IST requires the researcher to specify the text unit—the smallest chunk that can be coded and retrieved (such as a page, section, paragraph, line, or word; a text line is recommended in the manual)—set by hard returns in imported documents. The researcher codes a text unit, which begins and ends only approximately with the data item of interest. ATLAS / ti, in contrast, allows selection of any-length text passages for coding, and allows overlapping passages or text units, so context dependence of the item coded can be preserved.

NUD*IST allows text to be appended to documents, and the user may type comments directly into a document and code them as "document annotations" that can be searched and browsed. In ATLAS / ti, once a document is imported, it should not be changed. Coding is linked to locations in the document, not to particular words or phrases, so if text is changed and text lines shift, the coding becomes nonsensical. NUD*IST has a formal structure for entering header information. While NUD*IST, then, allows in-document notations and has a structure for a header, the ATLAS / ti user must create memos attached to the document, text passages, and / or codes to cover these needs.

Creating and applying codes is first-level analysis, exciting but eventually tedious. These software packages relieve some of the tedium of applying codes. The index system in NUD*IST is designed to store and locate codes, or categories, for thinking about data. The codes are stored at a node, a "container" that references text units coded for that code. Codes can be created in advance; they can also emerge from exploration of the data (for this project, the data were first coded in ATLAS / ti, so all codes were created in advance for use in the NUD*IST portion of the review). Previous versions of NUD*IST structured all coded nodes into a hierarchical index tree that showed relationships in the data in parent-child format. This form still dominates in version 4.0, but there is an option for free nodes—codes not hierarchically linked to others. Coding can be performed manually by selecting text and attaching a code to it, or by entering text unit numbers directly into the index system. Codes in use are letter and number combinations that are related to position in the index-system hierarchy. As coding grows complex, working with the abstract letters and numbers in earlier versions required a cross-reference list to the terms or concepts they stood for; version 4.0 has begun to make visible the natural language codes. Codes in ATLAS / ti are natural language terms or any terms created by the user. As coding of documents proceeds, it is easy to create, assign, modify, and merge codes through pop-up and pull-down menus and click-and-drag functionality. Both programs have aids to search and code repetitive occurrences. Both programs show text and codes in side-by-side display: see Figure 1 (NUD*IST) and Figure 2 (ATLAS/ti) for sample displays.

The equivalent of the NUD*IST index tree display is, in ATLAS/ti, the object explorer, which enables a hierarchical, tree view of families, codes, memos, and more. Exploration and theory development in ATLAS/ti, however, typically proceed through use of other grouping devices. Families, for example, are user-selected subsets of codes that filter the large set of codes and quotes for focused analysis. The researcher also may employ supercodes; these store the method to "calculate" collected entries rather than storing hard-wired links as normal codes do. ATLAS/ti features a network editor to promote experimentation with grouping of codes, quotes, and memos; one uses drag-and-drop capabilities to bring components into a network-editor window, then creates connectors to indicate the type of relationship between network components (e.g., is-a-kind-of, is-a-cause-of, contradicts). The query tool is another grouping aid; the researcher can develop queries and the software will assemble quotes from all documents that meet the query criteria. Queries may be Boolean, may search on proximity (e.g., one within another, one following another), or may be semantic (searching for meaning among the components assembled in a network).

In NUD*IST, one can nominate up to (but no more than) 26 coding categories to compare for one or more documents. These are maintained as coding stripes (references to particular nodes keyed by a letter of the alphabet that appear in the margin alongside text units; see Figure 1). Thus it is possible to see where coding at a particular node begins and ends, as well as where it overlaps with other codes. Margin codes, the letters used in coding stripes, are identified by a key at the top of the report.

Figure 3 is a NUD*IST project with five windows open. The document explorer indicates that there are five imported documents and shows the header for the selected document. The node explorer shows that there are 52 free nodes (structure collapsed) and 65 nodes within the index tree (structure expanded). The right-hand side of the node explorer gives information about the selected node (2 1 1, Used/Land/Agriculture; here "used" or "not used" is the first level of coding on all codes which were created in advance as explained above) indicating that fourteen text units from three documents are coded at the node. Viewing the text associated with node (2 1 1) in the node browser brings up

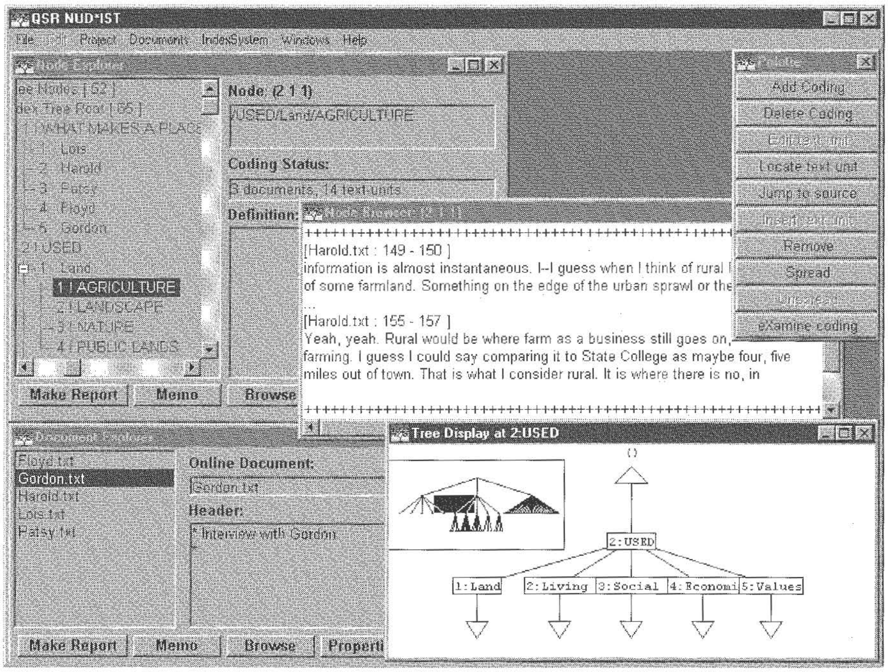

Figure 1.

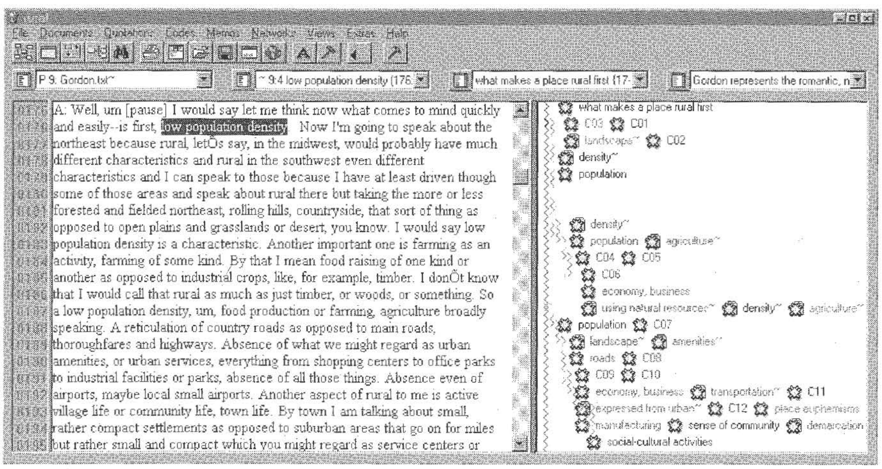

Figure 2.

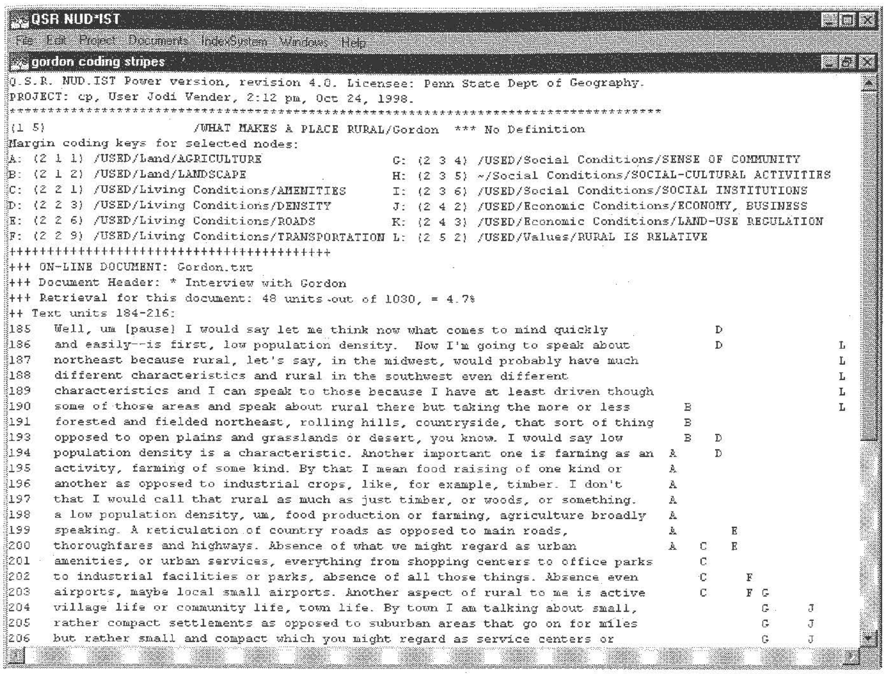

Figure 3.

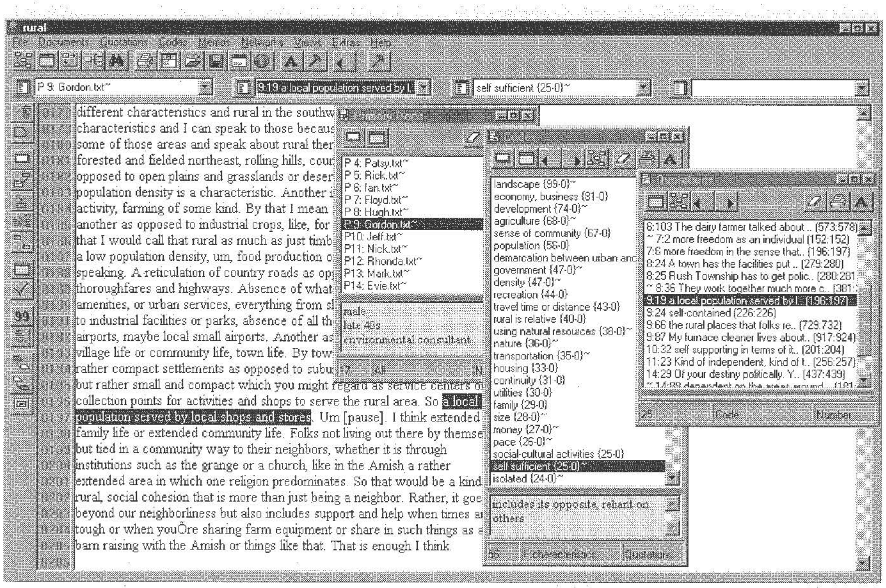

Figure 4.

the text units for the three documents coded at the node. For example, the interview with Harold contains five text units relating to the category agriculture. Coding of this or any other passage can be viewed or manipulated through the coding palette. The tree display depicts the hierarchical relationships among the nodes in the index tree.

Figure 4 is an ATLAS/ti hermeneutic unit (project). We have displayed three lists as detached windows simultaneously, but the same lists are accessible, one at a time, through drop-down lists above the large document display window. The primary documents window shows the list of documents (one for each interview transcript) linked to the project. The tilde indicates where a memo is attached. The comment for the selected document is shown in a text pane below the document list. Self sufficient is highlighted in the codes window. {25-0} after self sufficient indicates that 25 quotes from all of the documents are coded self sufficient and there are as-yet zero links from within network editors to this code. Again, a memo for the highlighted item shows in the text pane. The bottom margin gives information about the window: 56 codes are listed; the current filter is the family ("F:") called characteristics; and the list sort criterion is set to order by number of quotations. Bringing up the quotations window for the code self sufficient and selecting one item opens the relevant document and highlights the full quote in the large document-display window. The numbers preceding and following the quote segment in the quotations window indicate document and coding sequence (i.e., ninth document, 19th passage coded) and location (lines 196 to 197) respectively.

The make-report feature in NUD*IST generates a copy of any part of a document's text. One may choose to include any combination of the document text, document header, and document memo. Coding options for display include: no coding; summary only (list of nodes referenced and the number of text units coded at each); coding stripes; and cross references (with or without node titles) to all other node addresses coding a text unit. It is also possible to create a report listing all documents in the project, with options to include the header, coding status (no coding, or the number of documents and text units coded at a particular node), and data for each node (coding status plus node definition). Reports generated on particular nodes can show either a) general data (definition, date of creation and modification, sibling and child codes) for the node with options to display associated memos and / or document summaries; or b) references for the node with options to display headers, text, coding stripes, cross references with or without node titles, for all documents or only one. A report (or any section of it), or anything else one creates on screen, because everything is a text document, can readily be printed or exported to a word-processing application.

Several default reports are easy to produce in ATLAS / ti: The user can choose a code and print all quotes for it, similar to the ability in NUD*IST to report on a node. ATLAS / ti includes a system-generated coding history; it records who assigned the code to a quote, useful with multiple coders on a large project, and it retains the lineage of merged codes in a system-generated comment attached to affected quotes. A matrix of each document by each code with counts in cells of coding frequencies also is easy to produce. But beyond basic lists, print functions in ATLAS / ti are scant. It is evident from browsing the ATLAS / ti listserve that researchers are frustrated with the lack of print capability, particularly wanting to print out a full document with its codes for project documentation. The software developer's bias is toward on-screen work rather than large print jobs. While using fully the graphic network capabilities is easy and important to exploratory thinking, the only ready way to capture that work is with the computer's print screen function, which in our experience truncates all but small displays.

ATLAS / ti was developed to make full use of Windows graphics capabilities, so its appearance is modern. NUD*IST has DOS roots, and in this version, the tree display that graphically depicts hierarchical relationships among nodes remains crude and rather annoying in its inflexibility; this is especially vexing using the package on a Macintosh computer. Fortunately, Version 4.0 offers several alternative methods of working with the system not previously available. Both NUD*IST and ATLAS / ti are focused on qualitative data analysis so make provision for export to other software for other functions. NUD*IST, for instance, exports to Inspiration and Decision Explorer, among others, for more sophisticated graphical display and model building. Both NUD*IST and ATLAS / ti export to SPSS for further statistical analysis if appropriate.

For further reading on these two software packages, and other software adapted to qualitative data analysis, see *Computer Programs for Qualitative Data Analysis* by Eben E. Weitzman and Matthew B. Miles, from Sage Publications, Inc. (1995).

reviews

Cartographic Encounters: Perspectives on Native American Mapmaking and Map Use. G. Malcolm Lewis, ed. Chicago: University of Chicago Press, 1998. xx, 318 p., illus., maps, index. $60.00. Hardcover (ISBN 0-226-47694-4).

Reviewed by Jovanka R. Ristic
American Geographical Society Collection
University of Wisconsin-Milwaukee.

From the earliest days of the encounter between native North Americans and Europeans, maps and mapmaking have played a significant role in the exchanges between peoples with vastly different world views. This book brings together the ideas of a number of scholars representing a variety of academic disciplines in an attempt to trace the history of mapmaking and map use by native Americans, and especially how they were influenced by contact with whites.

The book grew out of a series of lectures which comprised the eleventh Kenneth Nebenzahl, Jr., Lectures in the History of Cartography, held at the Newberry Library in Chicago in 1993. G. Malcolm Lewis, in addition to organizing the program which included talks by Elizabeth Boone, Patricia Galloway, and Peter Nabokov, gave the keynote lectures and served as the editor of this volume. Realizing the need to expand the scope of the work beyond what had been covered at the Nebenzahl Lectures, Lewis solicited contributions from four other scholars who approached the subject of native American cartography in different ways and from different perspectives.

Arranged to reflect the chronology of events concerning this topic, the book is divided into three parts. Part 1 focuses on the 400-year period of the first encounter, Part 2 deals with the ongoing second encounter, and Part 3 attempts to predict future encounters.

Part 1 consists of three chapters written by Lewis which review the history of past encounters. He discusses maps and mapmaking among native North Americans as described and transcribed by whites in the field between 1511-1925, native maps studied by scholars in government bureaus, archives, museums, and libraries between 1782-1911, and perceptions of native cartography ca. 1970, when a 60-year hiatus in scholarly interest in the field was about to come to an end.

After a discussion of possible pre-encounter indigenous mapping, Lewis goes on to describe numerous examples of the types of cartographic encounters which occurred between natives and whites in the field. Evidence of native American maps, mapmaking, and map use during the first 400 years of contact exists for the most part only as described and transcribed by whites. Much of native mapping