Qualitative Sociology, Vol. 14, No. 4, 1991

ATLAS/ti—A Prototype for the Support of Text Interpretation

Thomas Muhr

At the core of ATLAS/ti is an explorative approach to theory-building. With linear textual data, such as transcribed interviews, as a starting point, segmentation and coding ("textual phases") of the text alternates with the building of conceptual networks and hypertextual structures ("conceptual phase"). The researcher may draw actual "maps," consisting of boxes and connecting lines, that depict the linkages among concepts as a network. Memos can be written for any entity at any stage in the process. For proceeding to a coherent text-outcome, features like the compilation of text units, and cut & paste operations between different text windows are available. All interaction with the program is through a graphic user interface containing windows, menus and icons, since special emphasis is placed in this program on its "readiness-at-hand."

INTRODUCTION

ATLAS/ti is a personal-computer program for the support of text interpretation that is being developed in the context of the interdisciplinary research project ATLAS (Archive for Technology, the Lifeworld, and Everyday Language) at the Technical University of Berlin, Germany. ATLAS/ti was from the beginning meant not only for our own research, but was conceived for general use in qualitative social research and linguistics. We expect that it will not remain limited to these areas (see the section on "Potential Areas of Application").

The interdisciplinary character of the research project contributes to quite favorable circumstances for the development of a (hopefully) useful

Address correspondence to: Thomas Muhr, Technische Universität Berlin, Fachbercich 2, IFP ATLAS, Hardenberstr. 28, D-1000 Berlin 12, Germany.

tool: The project entails close interactions between computer scientists, psychologists, linguists and future users in the context of concrete research undertakings. As we began to develop the program, we carried out a written survey among potential users in an effort to ascertain their views and needs concerning the current and anticipated application of computers in qualitative text analysis (Böhm, 1989). The results of this survey were taken into account in the conception of the software package. In addition to other desirable characteristics, the "user-friendliness" of the program was given an especially high emphasis. Our prototype is currently in the stage of "beta-testing" where users with different backgrounds explore the program in order to locate inadequacies and deficiencies concerning the matching of their range of tasks to the tool's procedures and its "readiness-at-hand." As with the results from the initial survey, information from the beta-test is fed back into the development process.

The purposes, underlying principles, specific functions and the user interface of ATLAS/ti are discussed below.

PURPOSES OF THE PROGRAM

ATLAS/ti is designed to offer qualitative oriented social researchers support for their activities involving the interpretation of text. This includes the capacity to deal with large amounts of text, as well as the management of annotations, concepts, and complex structures including conceptual relationships that emerge in the process of interpretation. One very early and important design decision was to leave creative, intellectual tasks with the human interpreter. The design objective was not to automatize the process of text analysis, but rather to develop a tool that effectively supports the human interpreter, especially in the handling of complex informational structures. We hold the view that an automatic interpretation of text cannot succeed in grasping the complexity, lack of explicitness, and contextuality of everyday knowledge. Any attempt to formalize this knowledge runs the risk of eliminating precisely the contextuality that is essential for human understanding, whether everyday or scientific (Dreyfus, 1990; Konrad, 1990).

The range of procedures that we incorporated in our program does not include quantitative statistical methods, or quantitative methods of content analysis. We do use quantitative aspects of the data for arranging and displaying the different kinds of information effectively (ordering codes by the number of referenced text passage, for instance). For more elaborate quantitative analysis the data, nevertheless, can be exported for further processing by appropriate software (SPSS).

UNDERLYING PRINCIPLES

In distinguishing the electronic tools intended for the specific task of interpretation from other types of software, such as word processing and data base systems, we characterize them as "Interpretation Support Systems" (ISS). These are to be differentiated from other programs not merely by their field of application, but particularly in terms of their novel architecture and of the system's specialization in terms of the requirements of the hermeneutic treatment of texts.

The Hermeneutic Unit

One important concept underlying ATLAS/ti is the "hermeneutic unit." The program is prepared to deal simultaneously with several research project that may or may not deal with related topics. Everything that is of relevance to one project (one research topic) is treated as one entity, regardless of the space in which it resides in the electronic environment. For instance, the primary texts (raw data), as well as the segmenting of these texts, the codes, the developing concepts, the linkages between the concepts, the networks formed by the various linked concepts, the memos attached to segments or concepts, etc., are all part of one hermeneutic unit. One obvious advantage of this bundling is that the researcher only has to deal with one entity. The activation of a hermeneutic unit is a straightforward selection of a single file; all associated material is then activated automatically.

Hermeneutic units serve various purposes: As a technical device, they constitute a container for all those pieces of information that are relevant to, applied during, and/or created in the process of one research project and can be handled as compact entities (e.g., for storing and retrieving). From a task-oriented perspective, a hermeneutic unit constitutes the thematic framework of an interpretation, embedding the user in one complete inquiry context. Last, not least, it serves cognitive economy by holding many chunks in one super chunk.

Methodological Influences

During the development of the program we were stimulated by the ideas, terminology, and methodological processes associated with "grounded theory" (Glaser & Strauss, 1967; Strauss, 1987; Corbin & Strauss, 1990), which was the predominant research method used by psychologists in a research project preceding ATLAS. This influence is partly discernible in the terms used in the system's user interface (menus, windows, etc.). This does not mean, however, that the tool is restricted to a certain methodology. The functions offered by ATLAS/ti are "generic" in the sense that they may be the building blocks that serve other approaches as well. The method of qualitative content analysis (Mayring, 1990), and other types of qualitative text analysis (Oevermann 1979; Volmerg 1983) are procedurally compatible with the basic functions of the program, as summarized by Böhm (1991). Despite its relative openness ATLAS/ti, however, represents a methodological conception of its own. This conception is partly the result of conscious planning, and partly also influenced by the unforeseen effects of our own reflections on the process of designing the tool ("what are we really doing?"). We are currently investigating how the application of this ISS may be changing traditional ways of interpreting text.

PROGRAM FUNCTIONS

Among the basic activities of text interpretation are usually the following:

* the selection of documents relevant to the particular research project;
* viewing, reading, and comparing of documents:
* coding and retrieving, i.e., grouping selected text passages within documents under certain categories, and extracting and compiling these categorized text passages;
* memoing, i.e., annotating documents, selected text passages, and codes;
* the creation of secondary texts, i.e., the actual products of text interpretation that may become included in scholarly research articles.

Most of these functions are already supported by existing programs for the analysis of qualitative data (although tools for making annotations are not yet common).

ATLAS/ti additionally offers:

* the integration of primary texts (raw data), text passages, codes, and memos into "hermeneutic units";
* hypertext functions on primary texts;
* complex search facilities within texts;
* semi-automatic coding;

* sorting of memos and codes into "families";
* the exploration and formulation of theories via the construction of conceptual networks;
* multi-authoring, i.e., many authors may collaborate on the same hermeneutic unit;
* examination of individual strategies of text interpretation;
* a knowledge-based tutorial component (planned).

To illustrate these functions I will provide a description of some basic as well as more sophisticated procedures.

WORKING ON THE BASIC LEVEL WITH ATLAS/ti

Getting Started

The work session sketched below is constructed from a didactical aspect: beginning with the more elementary, simple features of the textual phase, and progressing to the more complex operations of the conceptual phase. In "real life" the sequence of work may not necessarily proceed in this manner.

After having started the program, the researcher is presented a list of options (a "menu"). He/she may then start with creating or opening a "hermeneutic unit" by making an appropriate choice.

In order to create a hermeneutic unit the user is prompted by the program for a descriptive title. The title should reflect the subject matter of the investigation, something like: "How does the Bible relate to Kabala?". Next, this label is displayed at the top of a large window partitioned into several smaller subwindows. A window is an area on the screen in which information of various kinds can be displayed and manipulated. All manipulations, including the selection of an action from a menu of choices, are normally done with the mouse, a pointing device supplementing the keyboard (see also the section "User Interface").

The next step is to select those text documents for inclusion in the hermeneutic unit that are to be analyzed. After activation of the appropriate function (by "clicking" the mouse on an "icon" located at the left side of the screen), a list of text files is displayed, taken from the currently active subdirectory. One or more may be chosen—as usual by pointing with the mouse. A list of the "primary texts" thus attached to the hermeneutic unit now appears in a subwindow in the upper left corner of the main window. Primary texts constitute the raw data emerging from various research contexts which have been transformed into a computer-

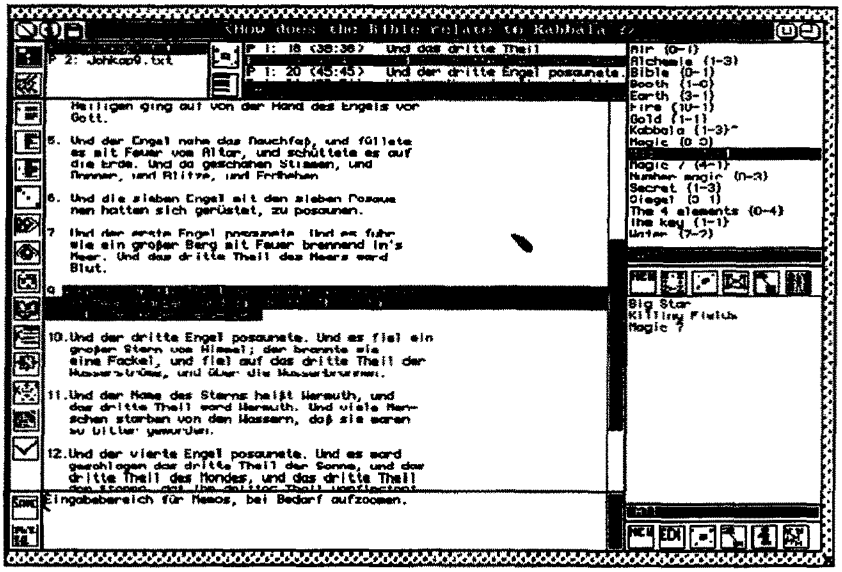

Fig. 1. A hermeneutic unit after some work has been done.

readable form (as ASCII-text) by such means as scanning, transcription of interviews, or retrieval from off-line data bases via wide area networks. Additional primary texts can be associated with a hermeneutic unit at any time; thus permitting data collection and analysis to be interrelated processes. Further selection of raw data may be guided by hypotheses developed in the course of the interpretation (see "theoretical sampling," Strauss, 1987). The different primary texts belonging to a hermeneutic unit do not have to come from the same directory, they may even reside on different disks or servers in a network. Secondary texts, incidentally, are what we consider the products of interpretation of the primary texts, and may become "data" for further analysis during the current or additional research projects.

To display any one of the primary texts, the user selects its name in the "list" window, located in the upper left corner of the screen. The document is highlighted, and its content appears in the largest of the tiled windows: the primary text area (see Figure 1).

The option to switch rapidly among different primary texts without annoying the user with requests for filenames, etc. enforces the comparison of the textual material, which is a main activity in text interpretation.

Naturally, the first task in interpreting text is to read it. It is recommended, especially with huge amounts of textual data, to create and read a paper print-out first. Reading on screen has proven to be an error-prone undertaking, as well as being a cognitively and physically burdensome procedure (300% higher rate of reading errors, compared to paper versions). This mode switching between screen and hard copy is supported through reference print-outs which supplement the printed primary text. Notwithstanding this advice, primary texts can quite comfortably be read on screen with ATLAS/ti.

Coding

While reading through the text the researcher (hopefully) comes across text passages (called "indicators" in the terminology of grounded theory: in our terminology called "quotations") which constitute examples of aspects she/he is investigating. Thus it becomes necessary to mark these passages and then assign them a code that serves as a "handle" to the quotation(s) it refers to. Coding is done as follows: after having identified a text passage worth being coded, the selection is done by pointing to the start of the quotation, then, pressing a mouse button while moving the device to the end of the desired text passage, highlighting it. A quotation can start and end anywhere on a line. Coding proceeds with the user being prompted for a code, which, after being typed in, is displayed in another subwindow: the code list window in the upper right part of the main window (see Figure 1). Simultaneously, a new entry is displayed in the quotation list window located at the top center. Its "reference" is generated automatically: if its source is the first primary text and it is the 19th quotation and covers line 41 to line 43, this is indicated as: "P 1: 19 (41:43)." Additionally, the first 30 letters of the quotation are displayed behind the reference as a reminder. The name of the primary text, the selection in the text area within its surrounding context, the quotation, and the just created code are all highlighted. This visual state can later be exactly reproduced, after large numbers of quotations and codes have filled the hermeneutic unit, by clicking on that particular code ("de-coding").

Codes usually refer to more than one quotation, and vice versa, one quotation can be referenced by more than one code. Activating ("clicking" on) a code holding more than one quotation results in a list displayed from which the user may then proceed to select a certain quotation, display one after the other, or produce a text containing all quotations belonging to this code (which can be printed, displayed on screen, or directed to a file). Additional information, such as the name of the primary text a quotation is contained in, and the list of (potential) other attached codes and memos is reported for every quotation.

The design of the coding functions available in ATLAS/ti is influenced by the methodology of grounded theory, as already mentioned. Following is a short, incomplete description of the different coding strategies and how they are supported by the program's procedures:

> Coding: Glaser and Strauss distinguish between three strategies which characterize different phases in the analysis of a text, but which need not always follow one another in a specified order. The first strategy entails "open coding," an exploratory examination and breaking down of the text material, using close-to-text "in-vivo"-codes, with the aim of deriving preliminary categories. In "axial coding," those categories held to be theoretically relevant constitute the point of departure for coding further text passages. Categories are then related to one another, and subcategories may emerge, thus densifying the evolving theory. Furthermore, those text passages that have already been ascribed to other categories may be coded anew. After (usually) several attempts at open and axial coding, the interpreter decides upon a "core category" and thus proceeds to "selective coding," now focusing on the core category and the relationship it has with all other categories found so far (Gross, 1989).

Open coding with ATLAS/ti follows the procedure as described above, by first selecting a text passage and then ascribing a newly generated code. For axial coding, the researcher may assign one or more already existing codes from a list of codes. Axial as well as selective coding includes relating existing and newly created codes with the aid of a network editor. This will be described later in more detail.

Memoing

Assigning codes to text passages is not sufficient for the purpose of text interpretation in terms of grounded theory. In addition to, or instead of being coded, text passages may be commented upon with a memo. Without this memoing activity there is a chance that coding becomes reduced to a mere classification procedure. Coding and commenting are considered the central basic activities in the process of text interpretation (Legewie, 1989; Strauss, 1987). In fact, these methodologists believe the actual ("text-to-text") interpretive work crystallizes in memos; they constitute the raw material for the "secondary" texts which are the results of interpretation.

Memos may refer to a primary text as a whole, to the hermeneutic unit as a whole, to codes (e.g., as explanations or coding instructions), or they may be comments on other memos. Comments on other memos supports collaborative and argumentative processes in those instances where the program is being used simultaneously by a number of researchers. One user might give a critical comment to a comment or code created by another user of the system, thus producing lines of argumentation and a web of positions, counter positions, and so forth. A similar cooperative approach has been taken in the design of certain hypertext-systems (Conklin & Begemann, 1989).

The procedure for creating memos in ATLAS/ti closely resembles the actions necessary for coding. After selecting a text passage, the user chooses the commenting function instead of the coding function from the options in the left-most window. In many situations, commenting may directly follow coding and refer to the quotation just coded, so that a new selection of text is superfluous. The memo is given a title, which is inserted in a list underneath the subwindow displaying the list of codes (right lower part of the screen, see Figure 1). Next, the user is presented with a list of attributes characterizing the type of memo: comment, critique, summary, and user-defined. These attributes are later attached to print-outs, or may be used for selecting specific memos (e.g., all critiques). For writing the memo, a space is provided just underneath the primary text area; the procedures are similar to writing with a word processor. In this space the content of a memo will later be displayed upon activation. Activation is triggered by selecting a memo in the list of memos, which simultaneously displays the reference to the annotated text passage, or is produced indirectly, when the quotation it refers to is activated (which also may be triggered directly, or indirectly by selecting its code).

Creating Hyperlinks

The implicit structure on text passages inherent in a given text is that one follows the other in a sequential order from the beginning to the end of a text. The marking of text passages and the coding and memoing define additional implicit relations between text passages. For instance, the program allows the user to display and read quotations sorted by creation date (see the section on "Sorts & Filters"), which could well mean switching among primary texts or proceeding from the end towards the beginning of the text. The segmentation of text into text-nodes (which one could view as their delinearization), the definition of links between those nodes, and the option to traverse different paths through such a structured document, all fall within the realm of hypertext, or hypermedia if the information linked is not restricted to textual material (Conklin & Begemann 1989). In addition to these implicit links we also plan to include the ability to specify named relations between text segments. It will thus be possible to define two text segments as being "contradictory" and later display all cases of contradicting evidence for a selected text passage. Another interesting hypertext link will be "follows thematically," which will produce sequences of coherent text that can be traversed in order. One area of application for these functions is the analysis of patterns of argumentation. It should not be forgotten, however, that those intertextual links are not derived automatically from the text, since the user has to create those links "manually" before being able to use the sophisticated retrieval and transversal procedures.

Revisions

Almost every entity created in the process described so far can be easily modified: Codes and memos that have already been delineated can be renamed, deleted or disconnected from their related codes. Existing quotations can either be deleted, uncoupled from codes or redefined by simply reselecting them. Memos can at any time be edited, renamed or removed.

WORKING ON THE CONCEPTUAL LEVEL WITH ATLAS/ti

Textual vs. Conceptual Level

In ATLAS/ti we distinguish between a textual and a conceptual level in the overall process of computer-supported text interpretation. This distinction is influenced by an approach outlined by Anjewierden (1987) who uses a similar scheme for describing the analytical levels in the acquisition of expert knowledge for the construction of expert systems.

The coding and memoing procedures described up to now can all be subsumed under the textual level, which covers all those activities aimed directly at the textual data: segmentation of text, assigning of codes to text, making comments about text.

As Corbin & Strauss (1990) point out, "concepts are the basic units of analysis." This is where the conceptual level becomes significant, which is characterized by a shift of the analytic focus away from the textual raw data—documents and text passages contained in them—and toward the concepts developed. Working in the textual and conceptual levels need not be viewed as strictly consecutive phases; rather, switching back and forth between the two is both possible and useful.

To transition from the textual to the conceptual level, the user proceeds as follows: After having created a number of codes attached to a number of quotations, the researcher may surmise structural regularities that warrant further exploration. This can be effected by either of two methods: grouping into "code families" or "networking."

Code Families

The concept of "code family" stems from the already mentioned methodology of grounded theory. We use it as an implicit structuring method for codes. Code families, like the codes themselves, have names and have comments attached to them. Within one "family" the codes are conceptually related. On the other hand, one code may belong to an arbitrary number of code families, which can be interpreted as different aspects of the code. One interesting use is to set a filter (see the section of "Sorts and Filters") so the codes visible in the code list area will be reduced to those belonging to a specific code family. As an example, the researcher might define a new code family called "C-Family" and subsume all codes referring to text passages in which the respondent or interviewee talks about consequences, causes, or conditions. The search function described later (see the section on "Text Search") can be used to find text passages in which the actual term "consequences" occurs. This feature facilitates axial coding along this aspect (thereby shifting back to the textual level). Code families can also be used as attributes for codes. For instance, a code family "investigate further" may mark all those codes that the researcher intends to elaborate/refine later. This grouping of codes reduces the overall complexity emerging in the course of the analysis.

We have also introduced the concept of a "memo family." Memos can be clustered by the same means as described for code families. This is very practical for sorting and filtering memos (i.e., memo family "theoretical memos"), and contributes to a more consistent way of handling the different entities.

Concept Networks

When the concepts (codes) are structurally enriched by establishing named relationships among them, they form "net-works." Simultaneously, new abstract concepts (which are not directly referring to text passages) can be introduced. The main purpose of code networks is to systematically visualize, refine and explicate the implicit structures that were inferred in the course of interpretation, thus encouraging an explorative approach to theory construction. We chose a network representation in favor of a hierarchical tree representation, because of the added degree of freedom for defining structure. Trees restrict a structure to being hierarchical. Theoretical or semantic

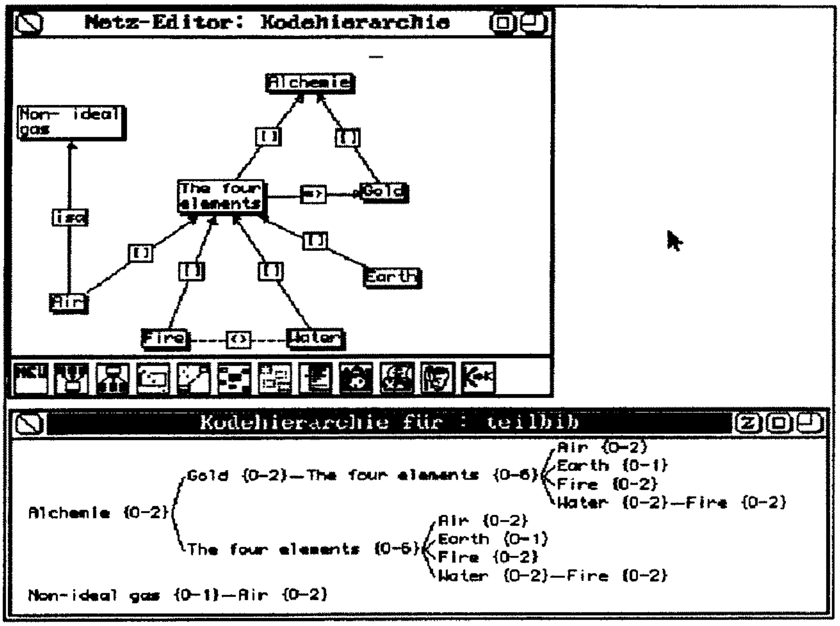

Fig. 2. Network and tree structure compared.

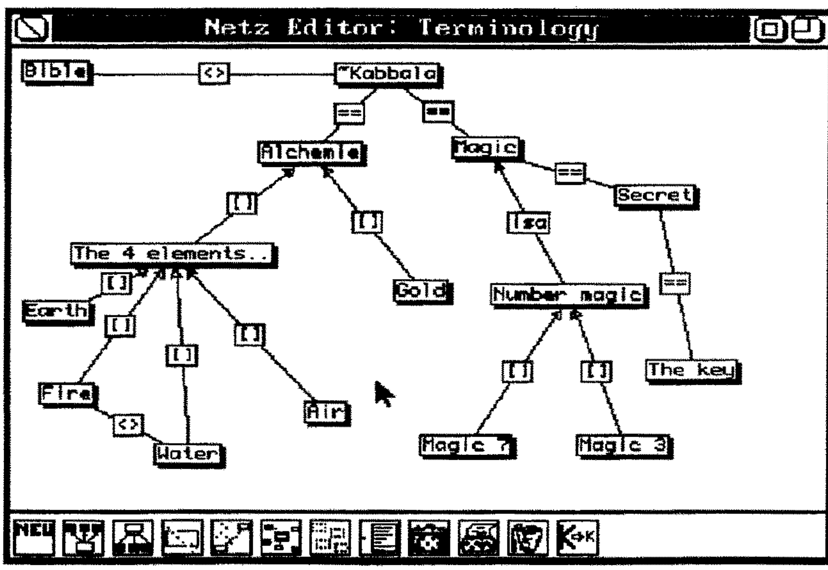

Fig. 3. Network editor.

Table 1. Built-in Code-Code Relations

Relation	Network-Symbol	Formal properties
c_1 *relates to* c_2	==	reflexive, symmetric
c_1 *is part/member of* c_2	[]	transitive
c_1 *causes* c_2	=>	transitive
c_1 *contradicts* c_2	<>	symmetric
c_1 *is a* c_2	isa	transitive
c_1 *describes* c_2	*}	asymmetric
c_1 *user-definable relation* c_2	??	??

frameworks seldom can be grasped as simple hierarchies. Nevertheless, every (cycle-free) network may be transformed into a tree structure (ATLAS/ti offers this option for completeness), but nodes are then displayed redundantly, cluttering what is meant to be clarifying:
Note the duplication of nodes in the tree representation.

In order to create and edit networks of codes, the user activates a special graphic network editor. The codes are displayed in small boxes, which can be selected and moved with the mouse. A relationship, such as "cause-of," that the researcher has defined between "anxiety" and "stress," for example, is displayed as a line with an arrow "–" between these two codes (see Figure 3). The layout of the codes in the network may be done manually or automatically calculated by the program, using formal properties of the relations linking the codes.

The process of linking codes is straightforward for the user: First, the set of codes (more than one code can be linked at the same session) to be linked to other codes are selected with the mouse. After the selection of the destination code, the user is asked to designate the type of relationship, which can be chosen from a set of predefined ("built-in") relations ("is-associated-with," "is-a," "is-cause-of," "is-part-of"), or can be constructed on the fly in the course of the activity. Newly defined relations can be stored for later access. For a list of built-in relations see Table 1.

ATLAS/ti "interprets" certain formal characteristics of the types of relationships. For instance, the relations "is-cause-of" and "is-a" are transitive. A transitive relation allows the derivation of relations not explicitly stated. For instance, in the example: *If* man is a vertebrate *and* vertebrate is an animal, *then* man is an animal, the last assertion is derived from the previous two. A counter example is the "contradicts"-relation. *If* A contradicts B *and* B contradicts C, one can't safely say that A contradicts C. The program lets the user collect (and eventually print) all quotations belonging to a code and to its transitively derived (sub-)codes. A visual consequence of the formal characteristics is that the network editor knows

Table 2. Entity Linkages

	Primary Text	Quotation	*Code*	Memo
Quotation	implicit N:1	named hypertext links N:M	–	–
Code	N:M	N:M	named relations N:M	N:M
Memo	N:M	N:M	N:M	N:M

if and in which direction to draw an arrow and how to place the nodes when using an automatic placement strategy: more abstract codes are placed above the less abstract ones.

Relations are – not as much as concepts – dependent on the content of the research project. Thus the user must be able to define new relations when needed. For instance, for argumentation analysis relations like "supports position" or "is argument against" might become relevant. These user-defined relations then co-exist with the built-in relations.

Links between codes can be traversed by double-clicking on a code in the code list window. A menu-like list pops up presenting all neighbours of the active code, i.e., all codes that are linked to the selected code. From this new list another code can then be activated, and so on.

Though they are meant mainly for connecting different codes, networks can also include other entity types: quotations, memos, primary texts. Links can be established among almost any entity types; see Table 2.

The first entry under Primary Text means that any number ("N") of quotations may come from one primary text, but one quotation can never be part of many texts. "Implicit" means that this link was not established by the user when creating a network, but as a result of a coding or memoing activity. "N:M" means that a code can link to any number of other codes, which in turn can be connected to any number of codes. A "named" relation is used, for instance, to establish the code-code-link "is-a" as opposed to code-memo links, which are unlabelled.

The different entities visible in the network-editor also display a certain "behavior" when they are clicked: for memos a text editor is activated, presenting the memo's contents for further editing. Codes display the researcher's comments that are specifically related to them, also for further refinement. When a quotation is clicked on, the program activates the corresponding text segment. (This is the same procedure as clicking on a quotation in the list window.) This "intelligence" of the entities, i.e., that they "know" their appropriate procedure is the result of the object oriented approach used for the development of the ATLAS/ti system.

In addition to linking entities, network editors provide for a number of editing tasks: creating new, and deleting existing "nodes" (such as memos and codes), "cutting" already established links between nodes, arranging the spatial placement of nodes, removing and importing codes from the list, and printing the network.

Network-Views

The user may construct from scratch or by derivation from existing networks ("snapshot") one or more independent network views. Each of these views must be given a unique name. They differ from each other in respect to the layout of the network nodes within the network editor. This feature facilitates exploiting spatial properties not easily grasped in relationships expressed verbally. Experimenting with different views further supports an explorative approach to theory construction. Nevertheless, all network views share the same network: deletion of a code of "cutting" a link between two or more nodes is instantaneously propagated throughout all views, thus preserving consistency.

To edit a network view, the user is presented a list of existing views from which one is chosen. More than one network editor may be opened at a time, so that different parts of the structure can be inspected simultaneously.

THE OUTCOMES OF WORKING WITH ATLAS/ti

The results of the procedures described in the previous two sections are, first of all, a number of primary texts containing selected passages (quotations) annotated by codes and/or memos. Code families and concept networks have been developed, supplemented by memos. While some operations can be best categorized as analytical, like the partitioning and classification of the text, others are better characterized as synthetic, like the construction of networks and the writing of memos. Further synthetic procedures may include the merging of memos, quotations and the researcher's narrative into a coherent text to be considered the outcome of the whole undertaking. For the purpose of integrating the different pieces of text, a global "cut, copy & paste" facility is available.

Printing

Each type of information within a hermeneutic unit has specific print functions at its disposal. In addition to paper print-outs any text can be directed to the screen or a disk-file (ASCII). For example: from one hermeneutic unit all quotations from all texts can be printed, or the hierarchy of all objects contained in the hermeneutic unit (primary texts, text segments, codes, memos); from primary texts all quotations within the current text, or the annotation hierarchy (primary>text>quotations>codes,>memos) can be printed. When dealing with quotations, ATLAS/ti will print the current quotation, or all quotations ascribed to the current code, or the former including the quotations of all transitively related (sub-) codes. Alternatively, all codes and their quotation references (location information about the primary text and the beginning and end indicators) can be printed, or the entire code-code network as a tree (see the section on "Concept Networks"). Of course, also all memos can be printed that are assigned to the code, or simply all memos, or memos together with the quotation to which they refer. Lastly, the user make print-outs of the graphic depictions of complete or partial networks.

MULTI-AUTHORING

Each and every entity created during the process of text interpretation – primary texts, selected quotes, codes, memos, etc. – is explicitly ascribed to the current user – the "author." By using ATLAS/ti, a particular hermeneutic unit (consisting of a corpus of attached primary texts, selected passages, codings and memos) may have achieved its current form as a result of the activities of a number of different text interpreters, each figuring as a separate author. Up to now, collaboration has had to take place sequentially in ATLAS/ti; only once one author has finished working on a hermeneutic unit can the next continue. However, we plan to allow users to work simultaneously in local area networks. Network views designed by individual authors can be selected and instantly generated ("show all entities created by XY"). Of course, hermeneutic units can either be made public by their "owners" and thus become generally available, or remain private and inaccessible to the public.

In addition to providing for an analysis of specific sequences of operations (which can be reconstructed since every entity is time-stamped at the creation date), the above-described cooperative work process offers the option of systematically comparing various styles of text interpretation from different authors. The modeling of cooperative work processes ("groupware") is a theme of great interest in current computer science research (Winograd, 1987).

AUXILIARY FUNCTIONS OF ATLAS/ti

The description above covers only the major functions provided to the qualitative researcher by ATLAS/ti. The program is, in addition, equipped with a number of features that aid in the practical aspects of performing an analysis.

Text Search

Text search–handled with care–can be a useful option for the qualitative researcher, even though she/he usually would not be interested as much in word usage as in "reading between the lines." However, simple search procedures such as those available in most word processing programs usually allow only a single sequence of keystrokes to be searched for at a time. Some use the "wild card" '*', anchored to word beginnings or endings, in which the "*" stands for any character (e.g., the search request "go*" would find "going" and "goes," but also "gold"; the search request "*you" finds "you" but also "bayou"). As these examples indicate, wildcards must be used with care.

A more sophisticated method to handle textual variations is the disjunctive (OR) search used in ATLAS/ti (Willenborg, 1991). It makes quasi-parallel searches for a number of search strings possible. With the simultaneous use of wildcards a search request might thus look as follows: "*caus*|why|*efore|since." The symbol "|", which can be read as a (Boolean) "OR," divides the individual elements of the "search swarm." Such a search might yield text passages including any of the following strings: "because," "causes," "causation," "before," "therefore," as well as "why" and "since." Search swarms which have proven to be useful for locating relevant passages of text, can be given names in ATLAS/ti (search-categories), under which they can be stored for later re-use (for instance: CAUSE:= *caus*|why|*efore|since). Thus, instead of retyping the series of strings, the user may simply specify the search category "$CAUSE" (the dollar sign indicates that the content of CAUSE is to be used for the search rather than the string "CAUSE" itself.)

Auto-Coding

Auto-coding combines ATLAS/ti's text-search capability with the automatic selection and coding of a text passage, resulting in a function that may be useful in cases where textual surface phenomena are relevant, or a preliminary screening of the data is needed. Whereas the direct selec-

Table III. Manual and Semi-automatic Coding

	MANUAL	SEMI-AUTOMATIC
FIND TEXT	Read until text passage is found	Search for indicating string(s)
MARK TEXT	Use mouse or/and keyboard to select the passage	Mark word, phrase or paragraph embedding the string found
CODE TEXT	–Create new code –Select one or more existing codes –Assign to active code	–Assign to active code

tion of a text passage by the researcher and the attaching of the appropriate code represents the genuine act of interpretation by the human interpreter, auto-coding takes place for the most part in a purely technical manner. Here the researcher selects a code under which text passages are to be subsumed, specifies the range of text to be included (word, sentence, or paragraph are the options available in the program), and then formulates the search request (see the section above). All text passages that meet the criteria of the search request are then located, the surrounding word, phrase or paragraph are marked and ascribed to the specified code.

Automatic coding would appear to contradict our original insistence that the activity of human interpretation must not be eliminated. We view this operation, however, not as an act of interpretation as such, but rather as a mechanized procedure for structuring data, which precedes actual interpretation. In this way, specific data pieces are tentatively made available from the given primary texts via algorithmic procedures. Their content must subsequently be interpreted and validated. For a comparison of manual and semi-automatic coding operations see Table 3.

Unless otherwise specified by the user, the names of primary texts, quotation references, codes, and memo names are displayed in their windows on ATLAS/ti's screen in alphabetic order. This default ordering of the entries can be changed any time. For rearranging the list of codes, for example, the user can choose among the following orders: time-of-creation, time-of-last-use, number-of-quotations, number-of-code-references. For the identification of core concepts (Strauss, 1987), for instance, an ordering by number-of-code-references (degree of embeddedness) may prove helpful when focussing on possible candidates. Core concepts refer to codes emerging from the phases of open and axial coding (see above) that have revealed themselves as especially fruitful for the developing "grounded theory."

ATLAS/ti also offers a means of reducing the number of objects displayed in a window. This is done with the help of "filters." The kinds of filter-options offered include: without-links to other objects, those-from-today, only-mine, only-those-belonging-to-a-set-of-collaborators, only-those-for-family-X. For instance, it is possible to display only those quotations that have no link to either codes or memos, which we call "free" quotations. Free quotations may then be systematically perused and eventually be linked to existing or new codes or memos. For comparative analysis it is useful to filter codes by the actual primary text. Switching among primary texts then quickly reveals similarities and differences in the use of codes. Another useful filtering criterion for codes and memos is the code/memo family (see above). To switch rapidly among different families, the user can open a special list window showing all defined families. By clicking on an entry, the contained codes/memos are displayed in their list windows. For memos it is possible to select only the comments for display that contain critiques, etc.

The sorting function also allows the user to calculate simple frequencies, such as the number of evidentiary passages or the degree to which a code is networked with other codes. In addition, it generates code-primary-text matrices (occurrences of codes for every primary text), which is useful for further processing by a statistical program.

Import and Export of Networks

The coding of text, and the structural interconnections the researcher defines among codes are currently restricted to the hermeneutic unit that contains the context in which the codes and their structure have originated. However, since the user also may want to begin investigations with an already existing "theory," ATLAS/ti includes a procedure for importing or exporting networks of codes from and to hermeneutic units. The technical representation of a network export file follows a notation based on first order predicate calculus (PROLOG). As such it can be imported as the foundation of a knowledge base for a knowledge-based system.

A thesaurus maintenance system is being developed in the context of the project ATLAS (Willenborg 1991), which is to be coupled with ATLAS/ti, allowing the evolving conceptual networks to be used for an additional broader purpose: a taxonomy for the description and retrieval of large numbers of data texts which are being collected as part of the project to eventually make up an archive.

THE ATLAS/ti USER INTERFACE

The central design issues for the ATLAS/ti user interface were interactivity and ease of use. Windows, menus and a mouse all are instruments supporting an easy-to-comprehend and easy-to-use style of interacting with a computer. The basic idea behind window technology is the notion that the needed context for any given function of the program should be visible. Objects of interest should be manipulated directly and not by the use of complicated command languages or deeply structured menu levels. This form of presentation is also referred to as a desktop metaphor, the style of interaction is called "direct manipulation" (Norman & Draper 1986, Shneiderman 1986). Window based software is becoming more and more popular, partly because of the decreasing cost of the necessary powerful hardware.

An interactive tool must not only be functionally correct and complete (to the extent that this is possible); it must also take into account specific human modes of work. Especially in creative activities with an extensive range of freedom, human beings tend toward a non-sequential mode of proceeding, characterized by numerous interruptions. In the process of coding, for instance, the researcher may be reminded that a memo should be written as certain things become clearer, such as the realization that this code can now be re-named and divided into two partial codes, etc.

Such a procedure demands fast, uncomplicated switching among various functions and the option of resuming interrupted activities (Miyata & Norman, 1986). Changing of the active work space (the active window) can be done by selecting it with the mouse, while the remaining work space is left on the screen to be resumed later.

ATLAS/ti makes use of context-dependent pop-up menus, e.g., if the mouse cursor is in the code list window, the code list menu pops up at the cursor's position. Frequently used options are made available in the form of "icons" ("buttons"). Icons behave like physical buttons: when pressed, specific actions are started. While for the use of menus first the menu itself and then the specific option must be selected, icons are permanently visible on the screen, and can be directly activated with a mouse click.

Another user-interface consideration is the provisions of "help." Throughout ATLAS/ti brief explanations can be activated online for each menu option and each icon "button." A hyper-text-like online help function offers tool-specific and task-oriented information concerning methods of text interpretation. In the future, the help system will be supplemented by a knowledge-based tutorial system that will supply active support to novices in text interpretation (see the section on Future Plans).

POTENTIAL AREAS OF APPLICATION OF ATLAS/ti

Although the "Interpretation Support System" we have developed is predominantly designed for use in qualitative social research, the areas of application are not thematically restricted. The following list of possible domains is intended only for illustrative purposes and could be easily expanded.

* Extraction of expert knowledge from transcribed interviews and think-aloud-protocols (experts could be engaged in discussions and/or observed in situations where expertise is required ("think-aloud"), resulting in textual material, which can then be searched for the concepts and relations essential for the domain of expertise. The conceptual networks can be exported in PROLOG notation and made a component of a knowledge base capturing the experts' knowledge);
* Analysis of structures of argumentation in round-table discussions;
* Interpretation of religious texts (the Bible, the Koran, etc.);
* Software documentation;
* Legal scholarship and practice: laws and relevant commentary.

FUTURE PLANS

ATLAS/ti is to include a tutorial component that offers the novice advice and help in the selection and application of various methods of qualitative text interpretation. These hints will be suggestive, not mandatory. The intelligent tutorial system will cover specific methodological aspects of text interpretation and, therefore, be quite different from automatized text interpretation.

Various other improvements of the program include the development of interfaces to other software packages, and refinements of the retrieval operations, in particular a query "language" controlled via a graphic interface for the more complex search and de-code functions allowing a combination of Boolean search and hypertext-navigation.

The program will be ported to MicroSoft Windows 3.0, OS/2-Presentation-Manager and MacOS. With these user interfaces ATLAS/ti can profit in several respects: device independence (printers, screens, etc.), multiasking (i.e., parallel use of ATLAS/ti, a word processing system of the users choice, and a retrieval system for access to the text bank), easy access of other programs from within ATLAS/ti (statistical packages), etc.

HARDWARE AND SOFTWARE PREREQUISITES

ATLAS/ti is written in Smalltalk/V 286 (Digitalk Inc.), but will be transferred to MicroSoft Windows 3.0, OS/2-Presentation-Manager and

MacOS. Both require a "mouse." Recommended for the program's use are IBM-compatible 80386 (or better)-based PC-ATs with system speed >=20 MHz and >=4 MB RAM. 80286 based systems can also be used, but because of the facilitation of a graphic user interface (GUI) the demand on resources is relatively high, compared to solely character-based solutions. ATLAS/ti requires approximately 3 MByte of free space on the hard disk. For an ergonomically satisfactory work place, large monitors (>=14 inch diagonally) are preferable; ATLAS/ti uses color and standard VGA resolution. On HP Laserjet, Canon LBP 8 and compatibles (e.g., Hewlett-Packard DeskJet) graphic printouts can be produced. Any IBM compatible printer is sufficient for textual output. With the MS-Windows version, availability and access on peripheral hardware is controlled by Windows.

REFERENCES

Anjewierden, A.
1987 "Knowledge acquisition tools." *AICOM*, 0(1), 29-38.

Böhm, A.
1989 "Bedarfserhebung für eine EDV-Unterstützung bei der Archivierung und Interpretation von Texten." (Bericht aus dem Interdisziplinären Forschungsprojekt *ATLAS*, 89-1, TU Berlin).

Böhm, A.
1991 "Vorschläge zur psychologischen Textinterpretation." (Bericht aus dem Interdisziplinären Forschungsprojekt *ATLAS* 91-1, TU Berlin).

Corbin, J., & Strauss, A.
1990 "Grounded theory research: Procedures, canons and evaluative criteria." *Qualitative Sociology*. New York: Human Sciences Press, 13, 3-21.

Conklin, J., & Begeman, M. L.
1989 "gIBIS: A tool for all reasons." *Journal of the American Society for Information Science*, 40(3), 200-213.

Dreyfus, H. L.
1990 "Hermeneutics and artificial intelligence." (Lecture held at ATLAS colloquium, TU Berlin).

Glaser, B. G., & Strauss, A. L.
1967 *The discovery of grounded theory: Strategies for qualitative research*. Chicago: Aldine Publ.

Gross, B.
1989 *Leben mit der Halbwertszeit*. (Diplomarbeit) Berlin: Technische Universität.

Konrad, E.
1990 "Phänomenologie und Künstliche Intelligenz." (ATLAS-Bericht 2/90). Berlin: IFP ATLAS, Technische Universität Berlin.

Mayring, P.
1990 *Einführung in die qualitative Sozialforschung. München: PVU.*

Mergenthaler, E.
1987 *TAS/C Textanalysesystem*, Benutzerhandbuch. Ulm: Ulmer Textbank.

Miyata, Y., & Norman, D. A.
1986 "Psychological Issues in Support of Multiple Activities." In Norman, D. A. & St. W. Draper (Eds.) *User centered system design* (pp. 265-284). Hillsdale: Lawrence Earlbaum.

Muhr, Th.
1990 "Technikgestaltung für den Alltag am Beispiel der Softwareentwicklung für die sozialwissenschaftliche Technikforschung." In Frey, D. (Ed.) *Bericht über den 37. Kongreß der Deutschen Gesellschaft für Psychologie in Kiel 1990,* Vol. 1, pp. 368-369. Göttingen: Hogrefe.

Norman, D. A., & Draper, St.W.
1986 *User centered system desig..* Hillsdale: Lawrence Erlbaum.

Oevermann, U., Allert, T., Konau, E., & Krambeck, J.
1979 "Die Methodologie einer 'objektiven Hermeneutik' und ihre allgemeine forschungslogische Bedeutung in den Sozialwissenschaften." In H. G. Soeffner (Ed.) *Interpretative Verfahren in den Sozial- und Textwissenschaften,* pp. 352-434). Stuttgart: Metzler.

Shneiderman, B.
1986 *Designing the user interface.* Reading, MA: Addison Wesley.

Strauss, A.
1987 *Qualitative analysis for social scientists.* Cambridge: Cambridge University Press.

Volmerg, U.
1983 "Validität im interpretativen Paradigma – Dargestellt an der Konstruktion qualitativer Erhebungsverfahren." In Zedler, P., & Moser, H. (Eds.) *Aspekte qualitativerr Sozialforschung.* Obladen: Leske & Budrich.

Willenborg, J.
1991 *PfleSaurus: Ein System zur Erstellung und Weiterentwicklung von Thesauri.* (Diplomarbeit). Berlin: Technische Universität.

Winograd, T.
1987 "A language/action perspective on the design of cooperative work." *Human-Computer Interaction.* New Jersey: Lawrence Erlbaum Associates, Inc., 3, 3-30.