ACTE : WORKBENCH FOR KNOWLEDGE ENGINEERING

AND TEXTUAL DATA ANALYSIS IN THE SOCIAL SCIENCES

François Daoust, Luc Dupuy and Louis-Claude Paquin

Centre d'Analyse de Textes par Ordinateur

Université du Québec à Montréal

P.O. Box 8888, Station A

Montreal, P.Q.

Canada H3C 3P8

S640@UQAM.Bitnet

1. A Sociological Approach to Knowledge Engineering

We will expose in the first part of this paper some neglected aspects of knowledge engineering, emphasizing the possible contribution of the sociology of discourse to the field. The second part presents the ACTE (Atelier Cognitif et TExtuel) project. ACTE is an integrated system now being developed by the Centre d'Analyse de Textes par Ordinateur (ATO) at l'Université du Québec à Montréal. Thirdly, we describe the SATO and D_expert Software applications and then sketch in a fourth chapter an integrated strategy for these two systems. We conclude by stating why the ACTE project should be promoted and implanted within public and para-public organizations.

1.1 The Coming of Age of Expert Systems

We have witnessed in the last twenty years or so the impressive development of Expert Systems (Farreny, H. 1985), Practical and technical knowledge associated with the definition and building of Expert Systems is now "textbook" material (McGraw & Harbison-Briggs, 1989). With its expanding capacity, computer technology greatly facilitates the embodiment of "expert minds". The technology is also having an important impact outside its very field: information processing has become a daily reality in public organizations. This however poses an important problem. Indeed, to what extent is it possible to confine "expertise" to scientific and technical activity? For the clerical organization of administrative structures requires problem solving abilities that, in the end, are not structurally different (though content may vary widely) from those that one finds in "traditional" scientific and technical activity.

1.1.1 The Scientific and Academic Fields

For the greater part of expert system history (Farreny 1985: 27-45), the type of "minds" encapsulated in expert systems represented but the smallest domain of overall minds in activity, namely the sphere of scientific and technical activity. The major problem to which researchers were confronted was one of representing the cognitive components of an expertise. If one were to eventually use a computerized problem-solving tool, one had to be able to translate knowledge structures into a data processing format. Not surprisingly, "formal" models were devised to achieve a representation of expertise, since the initial knowledge domain was already highly structured and documented (e.g. The valve is part of the motor which is part of the automobile).

1.1.2 The Industrial Field

With the establishment of personal computer technology, the notion of expert system is further applied to other spheres of activity. Since the mid-1980s, there has been a growing demand for expert system applications in the industrial field. In this context, the problem of representing the knowledge of a domain of expertise is coupled with that of acquiring this knowledge. Acquiring the particular knowledge of a domain of expertise raises two kinds of difficulties.

While it is relatively easy to dress a list of pertinent knowledge units in a domain of expertise, the problem is in ascertaining the various roles these units come to take. Identifying the knowledge functions that must be filled is even more uncertain as is identifying the strategic moments when these functions must be met in order to simulate expert reasoning. Moreover, the knowledge acquisition system must ensure the continuous assimilation of new "entries".

The variety of expert system applications also raises the more general concern of outlining knowledge. Indeed, one of the first tasks in conceiving and establishing expert systems is to identify the functions (Marcus 1988), of the domain of expertise to be structured. Since each domain is specific in itself, structuring its concepts and strategies must be done specifically. In other words, each domain has its own epistemological theoretical and methodological idiosyncrasies that must be taken into account fully. We have to acknowledge here that the strategies for solving or attempting to solve symbolic problems still are not well identified. There lacks even today a clear classification of methods used by various types of expert systems (McDermot 1988). We can only note the rise in "methodological experiments and the terminological expansion they entail.

The conception of expert systems used in the industrial field further adds to the problem of knowledge representation that of expertise transfer. The usefulness of an expert system lies in the fact that it should be able to mirror and retain the knowledge of experts in a given domain. This raises the problem of communication efficiency to the extent that messages produced by the expert system must be identical to those that an expert would produce. An expert system must be able to explain, justify and specify a course taken; ultimately, an expert system must offer a learning aid to its users (Barr & Feigembaum 1982). The use of an expert system also raises the problem of upgrading knowledge.

1.1.3 The Administrative Field

If we look at other types of expertise, say for example administrative expertise, we find that the development of knowledge-based systems has get to be achieved. In such a field, users and experts are mostly concerned with the handling (the production, the analysis, the management, etc.) of textual data. There are problems that arise above and beyond textual data simulation in the context of expert-system applications. We refer here specifically to difficulties brought about such operations (Gingras 1988) of storage, conceptual indexing and exploitation of great free format textual databases. Free format refers to that format which resembles a book or a periodical. Varying in geometry (be it a paragraph, a page or a chapter), this format is the main vehicle of knowledge pertinent to the administrative field: judicial texts, directories of policies and decrees, etc.

1.1.4 Problems of Methodology

The models used for analysis in a "scientific and technical" approach are very much inspired from analysis in cognitive psychology that center on the genesis and production of concepts (Eliot 1987; Reitman Olson & Rueter 1987). We have to admit that the social structuring of scientific and technical practice tends to favor clearly defined nuclei; scientific experts are usually well-known and recognized for their contribution to the field, which singles them out as potential sources for the establishment of a domain of expertise.

In those fields however where knowledge is stocked in textual archives, experts are not as easily identified. Often, the expert is a group where each member controls one specific area of expertise. One need only refer to the administrative policies of government institutions to convince oneself of the differences that oppose the "exact sciences" to the "humanities". For the latter, knowledge is structured as a polysemous universe where arguments have differing, if not contradictory, aspects. It is thus clear that expertise can only be apprehended in relation to the socio-cognitive characteristics of the groups of contributors.

Strict cognitive methodology leaves out the social dimension of expertise. Knowledge is constructed by and in discursive activity and micro-social interaction. We must then take into account many prospects, especially when knowledge pertains to lesser defined domains such as those found in public and para-public organizations. For example, a notion such as "l'intention de frauder" (the intend to fraud), is not easily analyzed in abstracto; recourse to the context provided by the archives is thus essential. Efficient use of the textual archive potential requires adapting knowledge-acquisition methodology to everyday concerns of administrative practice.

This we consider important because textual archives are the dominant mode of conservation of socio-cognitive structures that are but knowledge pertaining to such and such social practice. All things considered, we have to admit that any form of knowledge depends in all or in part on a socio-linguistic structure that allows the storage, handling and transmission of elements of a domain of knowledge.

1.2. Discourse Analysis as a Framework for Knowledge Engineering

If Expert Systems are rapidly spawning, the same cannot be said however of our theoretical grasp of expert knowledge and the socio-linguistic structure acting as its life-support system. Oddly enough, knowledge engineering remains "the principal bottleneck in the development of expert systems" (Feingenbaum & Barr 1982: 84). We concur that this is so because the current definitions of knowledge neglect the collective and contingent dimensions (Poitou 1987). Knowledge is basically a group product and as such, it is a direct result of collective organization. Knowledge is also contingent upon time. It is embedded in textual archives, e.g. discursive structures that are dependent on institutional frameworks. These structures are constantly modified and updated. Experts in the administrative field deal with information that is neither stable, unified or homogeneous. These aspects of the elaboration of knowledge have yet to be taken into account in the discursive matrix where they are stored: texts (working papers, accounts, progress reports, articles, etc.). The handling and definition of a textual format should work toward a more productive knowledge engineering.

1.2.1 Discourse Analysis as a Means for Conceptual Domain Analysis

Discourse analysis as practiced in sociology (Ghiglione & Matalon 1985; Lecomte 1985, 1988) is based for the most part on the hypothesis that statements of a discourse appear as regular-shaped object-core forms and networks. Analyzing discourse morphology amounts to constructing a model of the text by listing the conceptual objects found in the syntactic strata of the text and by reconstituting the semantic itineraries followed by these objects. Morphological discourse analysis capitalizes on the distinctive feature of natural languages ob being their own metalanguage, that is to say they can both depict reality and the representation of reality. This in all is a reading by retrieval and sampling of textual segments describing the major stakes of the discourse. Structured together, these segments form a new text that presents itself as the result of interpretative practice. The construction of textual sequences is carried out along a nominal and verbal axis. In the first instance, the text is apprehended from the relations woven by nouns forms, e.g. the systematic repeating of a semantic category by means of various nominal or pronominal phrases. In the second instance, verb forms and gerunds (nouns derived from verbs) found a logic of action by steering the courses taken by statement subjects. Some verb forms may be used to mark the opposition between the continuous and the discontinuous, between potentiality and actuality, etc. In this perspective, natural logic guides our study of object structuring; grammar (semantics plus syntax) helps us in singling out the material regularities of the language in which these objects are represented.

Discourse analysis represents thus a mode of textual intervention that may take into account conceptual elements and the organization they entail. Up until now, the methodological benefit of establishing a link with discourse analysis in the realm of knowledge acquisition (expertise transfer) has not been fully appreciated by experts. They do recognize however the importance of protocol analysis, to name but this particular instance of textual archives.

2. The ACTE Project

2.1 The Project's Context

In large organizations such as those of the government, textual production - in the form of reports, guidelines, memos, etc. - is increasing at a rate that hinders its exploitation. "Text workers", researchers, managers and decision makers whose main activities are reading and analyzing texts, text workers are thus submerged by a mass of documents they must analyze according to specific objectives: accumulation of facts, events or knowledge, interpretation, strategy planning, decision making, etc.

On the other hand, computer tools and methods for understanding texts have been developed and further improved in research laboratories with regard to performance as well as theoretical validation. Furthermore, the generalized use of "word processing" programs and textual databases has made possible the exploitation of the noticeable source of knowledge and expertise that constitutes the body of texts produced within an organization. The time has then come to promote the transfer of expertise from laboratories to organizations.

2.2 The Philosophy Behind ACTE

The ACTE project is specifically oriented toward the integration of primary textual sources in a process of knowledge engineering. Knowledge can be extracted and formatted from the textual basis to become either goals, facts, rules or inference mechanisms. ACTE is designed as an interactive environment for interfacing standard and textual databases, a textual parser and an expert system generator. The textual databases and parser will be managed by SATO (Système d'Analyse de Textes par Ordinateur); once the textual data has been properly translated into goals, facts or inference rules, the data can be taken in charge by the D_expert environment for the generation of expert systems.

2.2. The Expertise of the Centre d'ATO

With contributions dating back more than a decade ago, the Centre d'ATO has a vast experience in the development of software tools for textual analysis: the SATO, Déredec, FX and D_expert programs for example. Researchers at the Centre d'ATO also share a work experience in public and para-public organizations, most notably as training officer in knowledge engineering and computer-assisted textual analysis. Here is a partial list of projects commissioned at the Centre d'ATO by the Government of Quebec:

- at the Ministère des Affaires municipales, use of SATO for municipal law analysis;

- at the Ministère de l'Agriculture, use of D_expert in a diagnostic system context;

- at the administrative offices of the Conseil du Trésor, use of SATO for the treatment of the administrative policy directory; use of D_expert for the establishment of an expert system for the attribution of a certain type of service contract (SAGAC);

- at the Ministère de l'Education, use of SATO for the location of terms proper to the domain of evaluation and for textual measure strategy planning; use of D_expert in an expert system context as a guide for choosing statistic strategies;

- at the Ministère de l'Environnement, use of SATO and D_expert in the SAGEE project (Système d'Analyse pour la Gestion des Evaluations Environnementales);

- at the Ministère du Revenu, joint use of SATO and D_expert in a training program for tax auditors.

Originally established around a few key researchers, the Centre d'ATO was first a research team (1983), then a research service body (FCAR, 1984) and a research laboratory (UQAM, 1986). It now employs ten full time researchers and its budget nears the half-million (Canadian) dollar mark.

2.4 The Project of a Workbench for Knowledge Engineering and Textual Data Analysis

The ACTE project, a workbench for knowledge engineering and textual data analysis, was born of two needs. It has become necessary to increase the strength and user-friendliness of existing systems. This is indeed the case for D_expert, a functional prototype in LISP from which we wish to retrieve a working module that could be installed on personal computers in use.

The close link between the two systems (SATO and D_expert) appears to provide an adequate answer to the development and generalization of projects already in progress. Indeed, the methodology developed for retrieving knowledge lies for the most part on textual data analysis. Moreover, handling texts with software tools could greatly benefit from expert system methodology that allows for a gradual construction of complex algorithms.

Schematically speaking, the relevance of the ACTE project is summed up in the following arguments:

1. the feasibility of such a tool depends on the existence of tried-and-true software;

2. this project will allow experts to develop and strengthen existing software in order to increase their performance and efficiency;

3. the ACTE project will provide the means to improve and standardize the ergonomic of existing tools in view of their inclusion in the integrated module.

4. the project will also provide the means to increase the office interlink capacity of the system by adding a communication protocol allowing database access programming.

5. by thus uniting two major programs for "office" work, solid and general groundwork is laid for the development of specific applications responding to a variety of needs.

The project is divided into two consecutives phases.

First, the integration and optimization of existing tools: SATO is a lexico-textual analyzer and D_expert an expert system generator. All the while, the computational linguistics section of the Centre d'ATO pursues other research toward developing a lexico-syntactic analyzer of French (ALSF), in collaboration with the Institut national de la Langue française (CNRS, France). Second, the installation in the workbench of a number of linguistic applications stressing potential descriptions of the analyzer, e.g. term counting, the automatic construction of structured indexes, automatic abstract production, etc.

This workbench will be able to treat in real time great volumes of texts without imposing any theoretical preconstruct with regard to word processing modalities. Its architecture will favor the possible integration of modules or procedures from various sources and links with differently specified feeders.

3. A Description of SATO and D_expert

3.1. SATO

3.1.1 The Features of SATO

SATO (Daoust 1985, 1989) is thus a system for computer-aided textual analysis; it is meant to facilitate content analysis. In a way, it is a textual database system that allows the user to annotate multilingual texts and manipulate them in various ways: concordance listings, lexicon construction, word categorization, countings of any kind, lexico-statistic analyzers. SATO functions in a menu or command mode and has a help mechanism.

3.1.2. A General Outlook

SATO is conceived to let the user question his text and control each step of treatment. To do so, however, requires that SATO first read the text and recognize its many components: words, punctuation, numbers...) found in the text.

A most interesting feature of SATO is that it is possible to associate numerical or symbolic value properties to the words or forms of the text. It is as if we could add dimensions to a text in order to annotate it.

One can imagine for example that defining a property for a lexicon amounts to adding a column in the directory of textual forms. Just as one can imagine that defining a property for a text amounts to adding a line to the original text in which it would be possible to annotate each and every word of the text.

3.1.3 SATO Tools for Analysis

The tools for analysis found in SATO are arranged in six categories.

Lexicon generation, compiling each occurrence of each form in the text or parts therein.

Concordance listings, i.e. the listing of textual segments in which appear one or several words.

Word counting, i.e. the tally of words in a sequence of textual segments: sentences, paragraphs, etc.

Automated analyzers applied to the text in order to generate several interpretations of said text:

LISIBILITE supplies various indexes of reading fluency\difficulty of the text;

PARTICIPATION allows the user to evaluate the importance of a given class of words in a body of sub-texts;

DISTANCE gives a measure of lexical contrasts between various texts. This algorithm can also help in locating words or class of words that contribute most to distinguishing these texts.

DECRIRE can ascertain the distribution of a property on the text or lexicon.

Finally, the user can create his own analyzers by combining the latter primitive tools. The analytical sketch produced could then take the form of a procedure summoned by the EXECUTER command.

Categorization commands can assign values, more often than not categories, to the words or forms of the text:

PROPRIETE allows the user to define a new property, delete or modify an existing one. SATO is equipped with a legacy mechanism: a new property can inherit values from the mother property; a lexical property can be "projected" onto the text; the textual (symbolic) property can be "abstracted" at the lexical level;

VALEUR assigns a value to one or several words or forms;

DICTIONNAIRE allows the user to consult a thesaurus already on file;

CONCORDANCE can also assign values to words in a given context of phrases or co-occurrences.

Last, there is a full-screen categorizer that can very easily annotate any word or lexical form. This annotation is also easily modified and does not alter in any way the text itself, whose body and essence is respected.

Service commands control at will the printing and display of results. The user can thus elect to conceal certain words or property annotations during the display. He can underline (or display in several colors) words that possess a given morphology or particular property values, e.g. words characterized by a certain length or frequency, words annotated according to a specific grammatical or semantic category, etc. Interest in these techniques is mostly fueled from the fact that they can be used in a comparative analysis context. A lexicon considered in itself provides little interest, but when compared with other lexicons from other text, then we can observe significant regularities or irregularities. In SATO, the DOMAINE command can define any sub-class of words in the text in order to proceed with such comparative analysis.

It would then be possible to compare chapters, or define a domain that would be composed of sentences where one or several determinate words appear. It is also possible to compare sub-texts composed of words marked in a specific category. Let us say, for example, that we have identified every line of dialogue of every character in a play. We could then define each intervention as a sub-text.

3.1.4 A Straight and Efficient Syntax

SATO is constructed around a syntax that can describe with great ease and flexibility the primitive objects, i.e. words, of the text. In fact, this description constitutes a search pattern that covers the material aspect of the word as well as its property values.

Here are some examples:

parle the word "parle";

parle$ all words beginning with "parle";

p|ent all words beginning with "p" and ending in "ent";

p_rle all words beginning with "p", followed by any other character and ending in "rle", such as "parle" or "perle";

parl(e,ent,ure) "parle", "parlent", "parlure";

|ent*freq=5,>5 all words ending in "ent" and whose frequency is equal or greater than 5;

ab$*ALP=(fr,an) all words beginning with "ab" that come from the French or English alphabets;

$ALP~fr all words that are not French.

This descriptive word syntax, when combined with a general structure command, gives SATO great flexibility. We thus have at our disposal solid ground on which to install analyzers and promote more efficient communication between the user and the processed text. SATO uses are indeed almost limitless, even more so since the system can handle efficiently hundreds of pages of text at one sitting.

3.2. D_expert

D_expert (formerly known as Déredec-EXPERT : Paquin 1986, 1989) is an expert system generator developed in French in Quebec. The current version, written in Le_Lisp, works on Macintosh with a minimum of 2 mo memory, on IBM compatible computers with OS2 and 3 mo memory, and on VAX/VMS.

This expert system generator was conceived so as to enable non-computer experts to generate their own expert systems. Indeed, we favor outside interventions in knowledge engineering, even more so in a decreasing fashion, so as to ensure that the organization become autonomous in the instantiation and maintenance of the expert system tailored to its needs. D_expert is first characterized then by its easy handling. There are no commands to learn - only a minimal understanding of its architecture is needed to learn how to make the most of the system.

A six-hour training session is usually all it takes to initiate a user to the system.

3.2.1 Describing Knowledge

Clarity in description was not sacrificed so as to better the performance of the program. The character sequence length (a maximum of 256) promotes the natural expression of the terminology of the domain of expertise, where terms are most often composed of several words. The proposed structuring of knowledge is that of the valued object (also called "granule"), which offers modularity, flexibility and readability. The valued object is defined by the value of its characteristics (also called "features"). It can reduce a multitude of various yet parent terms into knowledge systems. The value of these features is never a procedure, which in turn guarantees the independence of granules.

In order to facilitate their management, the granules are grouped according to various taxonomic criteria in what we term bases. The granules can be inscribed in a knowledge dependency chart; a multiple selective property legacy procedure establishes as many links as needed with other granules and manages the transmission of values. The semantics of these links is left to the user. Features and links are managed separately from granules.

The inference rule is the only control structure. Beside producing facts (inferring), it can:

- ask the user questions by making him instantiate the value of a feature (on Macintosh, this choice can be made by selecting a zone in an image);
- search databases;
- transfer values from one fact to the other;
- make computations;
- manage iterativity;
- send messages;
- evaluate LISP expressions;
- send commands to managing systems (DOS, OS2 or VMS(VAX)).

The register of available actions expands with the needs of users.

The pertinent facts to a given problem are grouped in queries. This allows the user to submit a problem and get a tentative result, submit another problem and complete or edit the first problem to submit it anew and so on.

3.2. Editing Knowledge

With D_expert, the user constructs and edits his expert system through embedded menu selections. All available granules are described listing every admissible value of each of their features in a dictionary (called here "knowledge structures"). Rule filters and inferences, as well as facts, are devised by choosing a granule in the dictionary and selecting a value for each feature of granule.

It is always possible to add that to or substract that from the dictionary: granules, set features of granules, values to set features of granules.

A spreader takes charge of revising set rules after changes have been made in the dictionary. A spelling corrector can modify any character sequence referring to a base, a granule, a feature or a value from the display of a granule base, rule or query. All identically located occurrences of this sequence, in knowledge structures as in rules and queries, will then be replaced.

Many import-export links have been established through text files:

- any chosen character sequence can be summoned from a textfile by entering a key;
- term lexicons may constitute value registers;
- knowledge structures are imported/exported in MORE 1.1 and THINKTANK format;
- knowledge structures and inference queries are imported/exported in matrix database format.

3.2.3 An Inference Engine

The inference engine compares the premise of inference rules to the facts (forward sequencing): all pertinent rules are cited at each cycle (the research tree diagram is read horizontally); the conflict between pertinent rules is solved by ordering them an increasing scale according to the number of filters in their premise; the treatment of incertitudes is done by combining trust coefficients (MYCIN); the tracer being multi-level, it is possible to specify which information is needed as for treatment progress: rule identity, filter results, coefficient and statistical drawing.

Feature values that are the object of a query can be documented from a free format text. A mechanism allows the navigator in a question hierarchy by selecting the "Aucun(e)" option by defect. Facts constructed on the same granule are consolidated if their feature value are not in contradiction. Facts that reveal hierarchical links are generalized; parent facts are generated with the values of their offspring. The results obtained and the answers given can be stored. It is also possible to generate a link with the primary files of a word processor by merging some results.

4. ACTE: A General Overview

4.1 Presentation

A workbench for knowledge engineering and textual analysis in the social sciences, ACTE will be constituted integrating SATO and D_expert. From a methodological standpoint, this integration is motivated by the interest in calling upon both textual analysis and expert system technology. The domains where this workbench can be applied are numerous. For example:

- information management;
- the construction of systems glared toward specific needs for text analysis;
- knowledge engineering from textual material;
- expert or knowledge-based system generation;
- etc.

From a computational standpoint, the integration of SATO and D_expert in the same programming environment does have its advantages. Both modules will fully share common function libraries (screen management, disk access, etc.). This pooling of resources also entails code economy, and savings in time required for writing and expurging this code. The material transfer of capabilities from one module to the other will be easier and speedier. Instead of exchanging informations through files, both modules will be equipped with a common memory space (blackboard).

At the moment, D_expert is a functional prototype; its development cycle is completed, its capabilities have been verified in the workplace. Actually written in LISP and with every structure stored in RAM, the unavoidable increase of the knowledge and inference rule base dictionary, which follows the development of expert systems from model to prototype, has revealed the inadequacies of the system in its present state. RAM crowding entails a greater configuration still and makes it impossible to run another application at the same time.

This brings two needs to the fore. One, we need to abstract those capabilities pertinent to the inference engine from those that pertain to data management (knowledge dictionary, inference rules and queries). These data, structured in tree diagrams, will be stored on disk in file form; only access indexes will be stored in RAM. Two, we need to work in compiled coding (C or PASCAL) rather than in interpreted LISP coding for greater strength and efficiency.

The integration of D_expert in the software environment of SATO will also mean that inference rules will be able to tap on-line the search patterns of SATO, execute SATO commands, access information servers, etc.

Inversely, SATO will benefit directly from the inference module capacity. As we've seen in the preceding chapter, SATO can be likened to a tool box offering a myriad of performing instruments that are handled interactively to "dissect" a text or verify reading hypothesis. Files bearing commands can be treated in sets, thus making for macro-commands. However, we would need to construct more sophisticated specific analyzers to incorporate more complex control strategies such as the conditional treatment of actions, case structures, etc. In order to construct these analyzers, the use of inference rules, a modelization mode that is becoming more popular, does appear timelier than the development of an ad hoc language. Moreover, the addition of a database manager unit will instantiate properties in the lexicon and the text as well by using free format values (character sequences).

The workbench for knowledge engineering and textual data analysis will comprise a lexico-textual analysis engine provided by SATO and an inference engine provided by D_expert. Interfacing will come from the most part from SATO; data management will be handled by a specialized module.

4.2 Integrated Ergonomics

SATO and D_expert will share the same user-interface in which the screen (25 lines at 80 characters/line) is divided in three zones: a text editor at the top, an object editor in the center and a menu dash at the bottom. A Command will allow the user to jump from one screen to the next. Our main concern is to avoid window superposing, such as rolling menus, for delays in restoring a screen are incurred and are also most unacceptable at a 2400 baud communication speed.

Except for Macintosh, the command menu dash is borrowed from the SATO library. It is located at the bottom of the screen and perused with cursors: direct access to a command is possible by typing its first letter; the current selection is displayed in reverse video.

The object editor is a buffer (virtual page) that inscribes a certain number of objects, words of the lexicon and text in SATO and elements of valued objects in D_expert. In SATO, this editor can manually annotate words by giving a value to one of their properties. In D_expert, the object editor can select a part of the structure to be edited, a granule feature for example. Commands pertaining to selection at the embedding level are then proposed: in this case, the commands proposed would be "Effacer le trait" (Delete Feature) and "Ajouter une valeur" (Add Value). The object editor is inspired from the current SATO annotation screen, but changes will be made to enlarge the typology of admissible objects to those of D_expert and allow limited backward perusal.

The text editor, a new concept, is equipped with a 100-line RAM buffer. It presents a minimum of options offered in the menu dash and can retrieve and save "small files". Commands composed from embedded menus are displayed in the text editor (command log). If the case arises, it is possible to bring some changes in commands already given, select them and submit them anew. This mechanism facilitates the ordering of treatable files and introduces SATO commands in inference rules. With the D_expert screen, the writing of inference rules whose actions are SATO commands begins; once the premise is completed, the SATO screen is displayed and SATO actions are applied by trial onto the text. When the result is satisfactory, commands are selected, the D_expert screen is again displayed and the selected commands complete the rule now being written. The text editor also serves to apprehend and edit character sequences that constitute knowledge objects and free format symbolic properties of words.

Coupled with the menu library, contextual help is displayed in a loop over the text editor. At the moment, the text sits in an external file, but is however stored at the beginning of the process. The workbench will rely on an indexed file to limit memory use. This file will also display the documentation added to features and values by the builder of the expert system, warnings and errors.

In addition to the interactive mode, the workbench will have a command language at its disposal. SATO already has such a language but we have to define one for D_expert. Finally, one sole syntax will consolidate both languages. An interface toolbox will be provided so that the builder should be able to tailor the ergonomic of retrieval screens and multiple choices offered to users to meet the needs of the organization.

4.3 Internal Structure Overhaul

D_expert is composed of two modules: a knowledge editor and an inference engine. Recoding D_expert implies two things: one, the transfer of knowledge data interface and holding functions to SATO interface libraries and a SGBD proper to ACTE; two, the PASCAL recoding from the LISP code of functions pertinent to the expert system shell. In an inference cycle, the restriction, i.e. the listing of all rules whose premise can be matched with the factual surmise (database), will be carried out according to a database strategy. Access to inference rules on disk will be direct since their address is to be inscribed in a RAM index.

This operation will entail transforming tree data structures into standardized files. One file model should give all knowledge data: the knowledge dictionary, inference rules (premise filters and conclusion actions) and queries (bodies of facts). This file model should allow for a better documentation of knowledge data: creation dates, editing dates, full text comments. The access key to cognitive data will be composed of a base/granule/feature triplet. The granule will not be represented anymore by a tree diagram, bur rather by a body of files that will have the first two identifiers in common: base/granule. The status of the object will now be explicit, not deduced anymore from the position it occupies in a given structure. Filter operators will affect inferences and facts; their array will be extended to necessity, obligation, optionality, etc. Value could be a pattern.

In the wake of the integration of D-Expert and SATO, some modifications will have to be made to the latter module. First, knowledge engineering of SATO results must be undertaken. A SATO granule base will be set up, as well as predicates, to test SATO-generated results in premise positions of inference rules. Each result will be divided in feature/value. For example, the description of a numerical property will have as features:

i) the class of words described (word pattern);

ii) the number of words it comprises;

iii) the average frequency;

iv) the mean.

Now displayed or stored in a single file, SATO results must then be kept in check in order that they be filtered by inference rules. This step will mean instantiating SATO command dependency files to store and further explain the contexts of these results.

5. ACTE: A Tool Benefitting Organizations

The proposed workbench is conceived to answer the present needs of organizations for textual databases, mainly selective access to knowledge within the texts. This access will be direct in real time by means of search patterns: finding for example all passages referring to maternity leave in collective agreements. If, however, the texts are sitting in a conventional database and they can only be accessed through a thesaurus, the workbench will intervene in the textual indexing process, being the choice of typical content terms and their reference.

In addition, the workbench enriches the methodology toward the transfer of human expertise to knowledge-base systems. Its ability to inscribe properties in the lexicon or text, which properties can later be questioned, allows for a constant, objective and reproducible tracking of concepts, independent of approaches defined in the text. It is then a question of using a metalanguage inherent to the text itself to single out organized and hierarchized invariants by their recurrence. After morphologically categorizing the words of the text, it is then possible to track terminological phrases from the co-occurrence of categories, e.g. traitement de textes {[nom] de [nom]}. On the one hand, analyzing nominal groups from a body of texts helps tracking knowledge units and their structuring into valued objects. Once pertinent concepts have been extracted from all substantives, noun configurations (called ingredients) that are associated with these concepts are seeked out. Thus, for example, in the case of the substantive "projet", we will have such configurations as "l'assujettissement d'un projet", "la pertinence d'un projet", etc. Adjectival forms found in located contexts reveal quantifiers and argument scales that virtually position other possible qualitative or quantitative value.

On the other hand, analyzing verbal groups helps in the inference rule writing process. Indeed, the study of action verbs allows the tracking of object-defined operations. Their inflection and context provide thus modulation (active, passive, necessary, optional, etc.), localization and temporality.

While it does access texts selectively by content query, in addition to providing help in the conversion of discourse objects into valued objects, the workbench is also conceived as a general tool for text content analysis. The neutrality of the instrument, by which several potentially contradictory levels of analysis may coexist, favors return steps between text model constitution and their empirical validation.

In ACTE, there is no deterministic projection of a pre-constructed model onto the text. The semantic knowledge and procedures belong to the user alone. The approach favored by the workbench is that of text organization upgrading by the addition of successive descriptions of the text alternating with the exploration of tentative results. With the lexico-textual analyzer, the user can very well project onto the text his own category systems elaborated from explicit hypotheses as to text interpretation. Listings could then be made on categories and nouns alike. This method brings the reader to ascertain those textual elements apt to bear meaning and settle the criteria by which these elements will be singled out and posted.

These are summarily the characteristics of the lexico-textual analysis component of the workbench. Moreover, by using a knowledge and inference rule dictionary enhancing this knowledge, the expert system generator can modelize and encapsulate expertise which will later be summoned by users.

This component of the workbench will make possible the creation of expertise incorporating the knowledge of (textual) database query languages. Through the workbench, access to information contained on several supports, each having its own characteristics, will not require mastering various query languages.

The inference engine mechanism, which sets actions off when factual contexts are met, suits not only the analysis of texts or decision-making but also the generation of texts (letters, reports, etc.). Beyond adding inferences to facts, a given factual context can trigger several textual segments describing and commenting the situation at hand. Such uses do not however exhaust the capabilities of this workbench. It has the potential to change our relation to the text, to this day linear, strongly marked by the dimensions of time (reading speed) and space (from the first to the last page), and make it automatic in that questions put to the text are immediately answered, and answers given can be questioned in turn, etc. User creativity is thus stimulated and promotes systematic recourse to texts. ACTE is also a powerful tool that modelizes textual tasks, thus allowing for the autonomous construction of tailor-made "textual applications" to be later integrated into everyday organizational procedures.

The integration of the SATO and D_expert software packages could also benefit the non-"guru" population (e.g. humanists) by giving them direct access to the knowledge engineering process. To this extent, we hope that semantic and cognitive data processing and non-numerical computational processes will be made more readily available to those who wish for a more complete analysis of the means by which social reality is constructed.

BIBLIOGRAPHY

Barr, A., Feigenbaum, E., (1981) The Handbook of Artificial Intelligence, Stanford, California, Heuristech Press & William Kauffman, Inc.,1981, Volume I, 409 pages.

___, (1982) The Handbook of Artificial Intelligence, Stanford, California, Heuristech Press & William Kauffman, Inc.,1982, Volume II, 427 pages.

Daoust, F., (1985, 1989) SATO, Système d'Analyse de Textes par Ordinateur, Centre d'Analyse de Textes par Ordinateur, Université du Québec à Montréal.

Dijk, T.A.v., (1977) Text and Context. Explorations in the semantics and pragmatics of discourse, London and New York, Longman,1977,

Eliot, Lance (1987) "Investigating the nature of expertise : analogical thinking, expert systems, and ThinkBack", Expert Systems, Vol. 4, 1987, No. 3, pp.190-195.

Farreny, H., (1985) Les systèmes experts. Principes et exemples, Cepadues-Éditions,1985, 254 pages.

Grize, J., (1983) "Schématisation et logique naturelle", Dans:Borel, Marie-Jeanne, Grize, Jean-Blaise, Miéville, Denis, Essai de logique naturelle, Berne, Éditions Peter Lang SA,1983, Sciences pour la communication, N° 4, pp. 99-146, 241 pages.

___, (1984) Sémiologie du raisonnement, Berne, Éditions Peter Lang SA, 1984, Sciences pour la communication, N° 9, 261 pages.

Lecomte, A., (1983), "Algorithmes de la séquence", "ACP Adéla", 24 pages. (Draft)

___, (1974) Essai de formalisation des opérations linguistiques de prédication, Grenoble, Université scientifique et médicale de Grenoble, 1974, 296 pages.

___, (1981) "La frontière absente", Dans:Conein, Bernard, et al., Matérialités discursives, Lille, Presses Universitaires de Lille,1981, pp. 95-104, 209 pages.

___, (1988) "Le marmot et la mamelle, critique des représentations du raisonnement", Centre de Coordination pour la Recherche et l'Enseignement en Informatique et Société (CREIS), Représentation du réel et informatisation, Saint-Étienne, I.U.T. de Saint-Étienne,1988, 21 pages.

Lecomte, A., Marandin, J.-M., (1986) "Analyse de discours et morphologie discursive", Charolles, Petöfi,1986.

Marcus, S., (1988) Automating Knowledge Acquisition for Expert Systems, Boston, Kluwer Academic Publishers,1988, 270 pages.

___, (1988) "Introduction", Dans:Marcus, Sandra, Automating Knowledge Acquisition for Expert Systems, Boston, Kluwer Academic Publishers,1988, pp. 1-6, 270 pages.

McDermott, J., (1988) "Preliminary Steps Toward a Taxonomy of Problem-Solving Methods", Marcus, Sandra, Automating Knowledge Acquisition for Expert Systems, Boston, Kluwer Academic Publishers,1988, pp. 225-256, 270 pages.

McGraw, K.L., Harbison-Briggs, K., (1989) Knowledge Acquisition. Principles and Guidelines, Englewoods Cliffs, N. J., Prentice Hall,1989, 376 pages.

Newell, A., (1968) "On the analysis of human problem solving protocols", Dans:Gardin, J. C., Jaulin, B., Calcul et formalisation dans les sciences de l'homme, Paris, Éditions du Centre National de la Recherche Scientifique,1968, pp. 145-186, 324 pages.

Paquin, L.-C. (1989) "An approach to Expertise Transfer: Computer-Assisted Text Analysis" Advances in Computing and the Humanities; en collaboration avec Luc Dupuy (à paraître).

___, (1988) D_expert, progiciel générateur de systèmes experts, Centre d'Analyse de Textes par Ordinateur, Université du Québec à Montréal.

___, (1987) "Déredec-EXPERT, un environnement informatique pour la génération de systèmes experts", 7èmes Journées Internationales 'Les systèmes experts et leurs applications', vol. 1, Paris, EC2: 61-78.

Poitou, J.-P., Chabot, R., (1989), "De l'analyse des situations de travail au système expert.", Université de Provence, TIP - CREPCO -UA, 12 pages. (Draft)

Reitman Olson, Judith, Rueter, Henry H. (1987) "Extracting expertise from experts : Methods for knowledge acquisition", Expert Systems, Vol. 4, 1987, No. 3, pp.152-168.

Swartout, William R., Smoliar, Stephen W. (1987) "On making expert systems more like experts", Expert Systems, Vol. 4, 1987, No. 3, pp.196-207.