Paper submitted to Australian Database-Information Systems Conference 1991

CSIRO HYPERTEXT RESEARCH PROJECT

Robert M. Colomb
John Robertson
Bob Jansen

CSIRO Division of Information Technology
Box 1599 North Ryde NSW 2113
02 887 9370
colomb@syd.dit.csiro.au

ABSTRACT

This paper described the archtecture of a hyperdocument system intended to provide access to a coherent body of knowledge, consisting of structured knowledge, text, graphics, data and executable models. The main focus of the research is in intermediate structure added to the documents to support the links between the structured knowledge and the textual knowledge. The system is implemented in a way relevant to delivery in an open distributed processing environment.

1. INTRODUCTION

The CSIRO Hypertext Research Project is intended to develop hyperdocuments which make a coherent body of technical knowledge accessible to users of the technology. The technical knowledge consists of

- structured knowledge (expert systems, system specifications);
- research reports and other documents, including graphics;
- mathematical and other formal models;
- data.

The ultimate aim of the project is to enable a user to access the knowledge relevant to a possibly complex requirement, and transform it into a useful form. The end product might be a view of the knowledge or it might be a product in its own right, for example the skeleton of an expert system.

Two domains of knowledge have been considered, in close association with domain experts:

- Wool: the physical and agronomic characteristics of raw wool and their relationship with wool processing (CSIRO Division of Wool Technology).

- Greenhouse: CO2 and its relationship with global warming (Macquarie University School of Earth Sciences).

Our role has been to develop technologies and prototype tools to assist the domain experts to prepare hyperdocuments: their responsibility is the knowledge and structures, ours is the software and structure schemas.

Our strategy is to build a series of successively more capable prototypes. Two prototypes have been completed and a third is in advanced stage of system integration. Briefly:

Prototype 0 is in the wool domain. Its knowledge consists of a small expert system and four research papers which contain key knowledge from which the expert system was built. It is conceived of as an advanced explanation facility for the expert system.

Prototype 1 is in the greenhouse domain. Its knowledge consists of 10 research papers and a model of the production of CO_2 from the global economy. It is conceived of as a tool to make greenhouse knowledge accessible to policy planners. In particular, the model has a hierarchical hypertext front-end for entry of parameters and has its output directed to a Wingz™ spreadsheet so that it can be manipulated and graphed by the user. Important parameters and outputs are linked to research reports which describe their meaning and significance.

Prototype 2 is in the greenhouse domain. It is similar to prototype 1, except that it has about 100 research reports and other documents.

Given a body of knowledge and an idea of the kinds of questions to be asked of it, we have been concerned with three key issues (Glushko 1989):

- how to structure the knowledge so that the questions can be answered (structure schemas);
- how to implement the knowledge management system so as to be acceptable to an end user;
- provision of tools to assist the domain expert in adding the additional structure to the knowledge according to the structure schemas.

Two additional key issues have not so far been addressed, but will be in the future:

- formal evaluation of the effectiveness of the end systems in meeting the user's needs;
- tools to assist in the maintenance of the structures under incremental change over the lifetime of the hyperdocument (software engineering).

Evaluation so far has been by informal means, by trying the system on selected users and receiving comments.

Prototype 0 (Jansen and Robertson 1989) focussed on the development of structure schemas. Besides the addition of links to the knowledge, which is of course the essence of hypertext, two additional structure types were introduced, assertions and a conceptual model, which will be discussed in more detail below. This prototype was implemented on a Macintosh™ II platform using Hypercard™ as a substrate.

Linkages in prototype 0 were implemented by a programmer using the Hypertalk™ language. Prototype 1 (Colomb and Jones 1989; Robertson 1990) concentrated largely on the provision of tools to enable the domain expert editor to make the linkages by pointing and clicking, and also experimented with the use of OCR technology (Omnipage™) to enter the basic documents into the system. This prototype was also implemented on a Macintosh™ II platform, but using Supercard™ as a substrate.

Prototype 2 is largely addressing issues of scale. First, the Hypercard™-style substrate is not entirely suitable for hundreds of documents Second, considerable editorial work on the part of domain experts is needed to add the necessary structure to the knowledge. Prototype 1 has made it much easier to actually implement the links, but extracting the intermediate structures (assertions, concepts) is quite laborious. Prototype 2 is developed for a knowledge base of about 100 documents, but its architecture is scaleable for systems containing at least 1000 documents in a coherent area of knowledge.

Our focus has been on the intermediate structures added to the knowledge base by domain expert editors. The implementation uses standard information retrieval technology (key words, boolean queries, document vector metrics such as described by Salton, 1989) such as used by SuperBook (Egan, *et al*. 1989) and Intermedia (Meyrowitz 1990), and is implemented using standard database and networking substrates such as used by Shipman *et al*. (1989) and Maurer and Tomek (1989).

This report describes the architecture of prototype 2. It first presents the structure schemas, then the modes of enquiry which the system is designed to support. It then

describes the implementation architecture, then the tools to support extraction of the structure, then the tools to support the semi-automatic creation of links. A summary and conclusion completes the report.

2. STRUCTURE SCHEMAS

The structure of the knowledge is shown in Figure 1. The knowledge from which the system is built is classified into data, structured knowledge, active models and reports. The active models are programs which simulate some process. The user can enter parameters, run a model, and view its output in a spreadsheet with graphics capability. Some of the data consists of a set of specific model parameters which are deemed by the experts to be important in some way. A user can call up one of these stored parameter sets, run the model, and view the results in possibly different ways from the original researchers. The user can also make incremental changes in these stored parameters, allowing them to explore what-if? scenarios.

Editorial structure added by the domain expert in the editorial process includes the links within the reports which join sections of the document, graphics, tables, etc. in the conventional way. The larger scale structure includes *assertions*, which are brief statements of fact or conclusion from the reports, and a *concept model*, which is at this stage essentially a hierarchical thesaurus whose leaves constitute an index for the reports such as one might find in the back of a book. The assertions themselves are interlinked as hypertext.

Samples of assertions included in the system are:

- The greenhouse effect occurs largely in the first 10-15 km of the atmosphere, that is, the troposphere;
- Estimates are that about 50% of newly injected CO_2 will remain in the air during the next century;
- Since 1850, CO_2 has risen by 25%: at the maximum of the last ice age 18,000 years ago, CO_2 levels were 25% lower than pre-industrial values;
- The greenhouse effect is actually a well established theory in atmospheric science, keeping the planet at a habitable mean global temperature of about 15°C compared with -18°C without the greenhouse gases.

A hierarchical thesaurus is a set of words linked by *broader term/ narrower term (BT/NT)* with some words linked by *related term (RT)*, which have the expected meanings. For example

terrestrial atmosphere cloud		cirrus cloud	
NT	BT terrestrial atmosphere	BT cloud	
air	NT	NT	
cloud	cirrus cloud		tropical cirrus cloud
ozonosphere	…	RT	
…	stratospheric cloud		cloud formation

air pollution

cloud albedo
cloud models

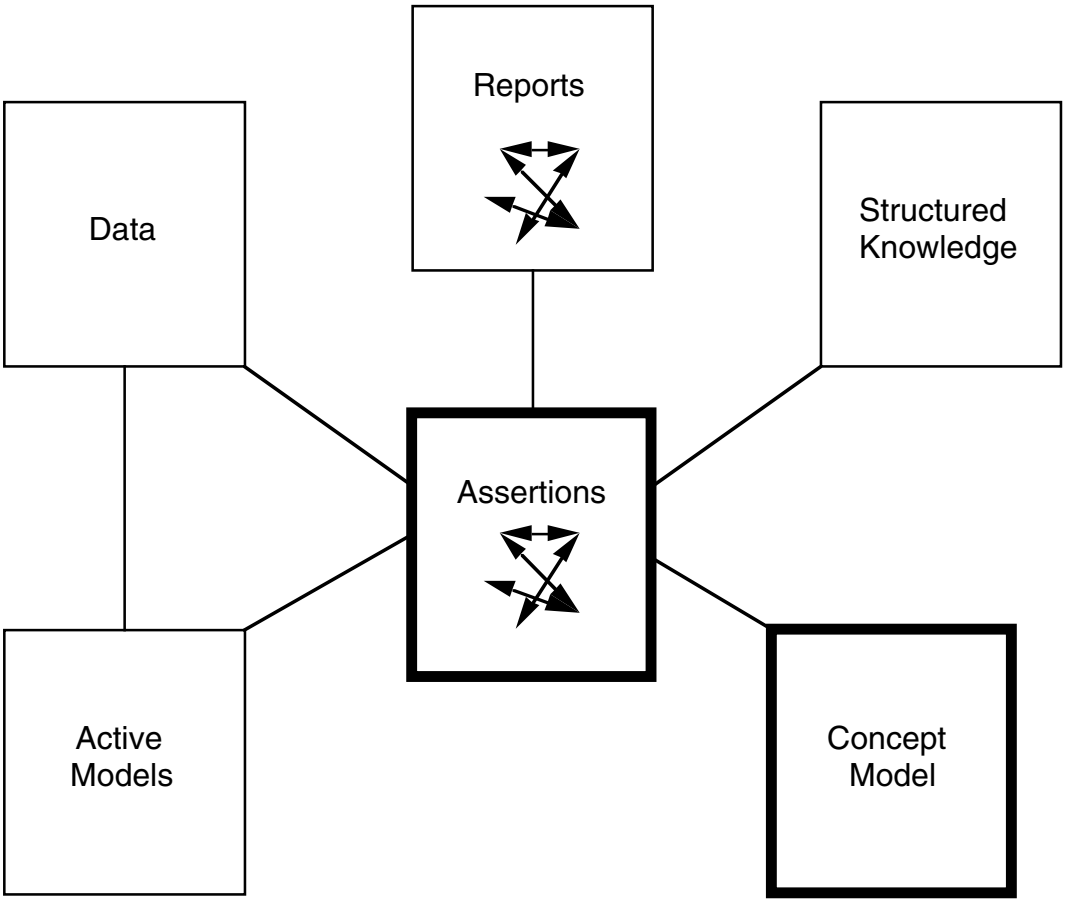

Figure 1 Structure Schema

The assertions form the central structure. Knowledge other than text is connected to the text only through the assertions. This structure was introduced at prototype 0, which was viewed as a deep explanation facility for an expert system. The user would interact with the expert system, with the capability of clicking on a why? button at any stage. Instead of getting a rule trace, as is conventional expert system practice, the system links the user with the research reports which justify the particular question or decision. We found that the documents themselves were too large a grain: the entire document is too big to serve as a good explanation for a specific decision. Links were therefore made to a statement in the document which was specifically relevant to the point.

However, it was found often that the statement needed to be paraphrased or a complex passage summarized to be useful. In addition, there were considerable numbers of general statements in the documents more or less relevant to the expert system. We decided to extract all these statements from the document into a table of their own. We in effect represent the document by a table of its significant statements, which we call assertions. These assertions are linked back into the document, so that it is possible to view the document as a justification of the assertions. A point in the expert system is

explained by a group of assertions, and the assertions are in turn justified by the documents.

In the prototype 2 system, there will be on the order of 1000 assertions, enough that the user will need assistance in navigating among them. The table of assertions will have therefore a hypertext link structure.

A *concept map* is the other main added structure. This is essentially a hierarchical thesaurus of about one thousand terms, based on and with a similar structure to the INSPEC thesaurus. This functions essentially as the sort of index one might find in the back of a book, except that it is organized hierarchically as well as alphabetically, and is composed of significant terms which actually occur in more than one document in the text data base. This follows a proposal of Regoczei and Hirst (1989) to provide assistance to access knowledge in text by the attachment of a concept map of the domain, and Glushko (1989) in its use as an entry point into the hyperdocument.

In the prototype 0 system, the concept map was associated with the assertions by a key word out of context search. In the prototype 1 system, the concept map was broader but not deeper, so that the lowest level was fairly abstract. The assertions were specifically linked to the lowest level of the concept map. At least in the first implementation of prototype 2, we will revert to the key word search method of linking, as there are many more assertions, many more terms, and the terms are derived from the text.

3. MODES OF ENQUIRY

Enquiry into the system is via several modes, all of them fairly conventional:

- entry of the document base directly through the table of contents;
- entry of the document base via the assertion table;
- browsing the document base once entered;
- entry of the assertion table through an *explain* button associated with the structured knowledge base, the model base, or the data base;
- entry of the assertion table through the concept map;
- entry of the assertion table via a key word query.

4. IMPLEMENTATION ARCHITECTURE

Prototypes 0 and 1 were implemented using Hypercard™-style substrates on a Macintosh™ II platform. This approach was not considered suitable for prototype 2 for two reasons. First, the existing implementations of these products do not support very well soft links into stacks of the size required. The second, and more important, reason comes from consideration of the operating environment these types of system will encounter as hyperdocuments become a commercial reality.

We can expect to see computing environments where the user interface resides on a workstation, but that the processes and databases required reside elsewhere, on systems connected to the user interface workstation via networks of various kinds. The entire computing environment will appear to the user as if it were on a single machine, but the internal architecture of these systems will have a radical separation of the user interface from the processes. Another reason for this kind of separation is that a single user will

interact with a large number of processes and databases provided by a variety of organizations, and also each process will interact with a variety of users with interfaces having different characteristics. These trends are driving the development of user interface management systems (Bass and Coutez 1988), and are being standardized by the open distributed processing (ODP) standards process of the International Standards Organization. It is therefore important to clarify the architecture of hypertext systems, separating the user interface issues from the database and process structure issues.

For these reasons, prototype 2 has its database aspects implemented on a SUN Sparcserver™ platform using the TITAN™ database as a substrate. Its user interface resides on a Macintosh™ II using Hypercard 2™ as a substrate.

The database aspects include:

- the sections of text and graphics which form the documents;
- the hard links between sections, both within and among documents;
- lists of index terms which appear in a section of text, which form the basis of soft links;
- the hierarchical thesaurus/concept model;
- the assertions;
- document vector representations of assertions;
- databases of research data.

The user interface aspect includes the presentation of the various kinds of objects in the system and the command/navigation facilities available to the user.

5. TOOLS TO HELP EXTRACT THE STRUCTURE

The editorial work needed to extract the structural information from the documents is a dominant cost in the production of a hyperdocument. The tasks are:

- extraction of the assertions, which is similar to producing a summary of the document;
- identification of the key terms, which is similar to producing an index for the document, such as may be found in the back of a book.

Our approach is to develop simple tools which amplify the ability of the expert editor to perform the task, rather than possibly more complex programs which do the job automatically (Salton 1989, Jones 1989). Figure 2 shows the process. First, a table of words occurring in the text is presented to the user, who partitions that table into a table of "keep" words, which carry the semantics of the documents, and the remainder, or "stop" words (the "stop" words are divided into generic stop words such as "the" and domain-specific stop words). The "keep" words are processed against the text to produce a table of phrases containing the words. This table is presented to the user, who selects from it the phrases which convey the semantics. This table forms the index for the document. A further pass against the text produces a table of candidate assertions, either sentences or sentence triplets, which contain the index phrases. This final table is presented to the user, who extracts from it the assertions desired.

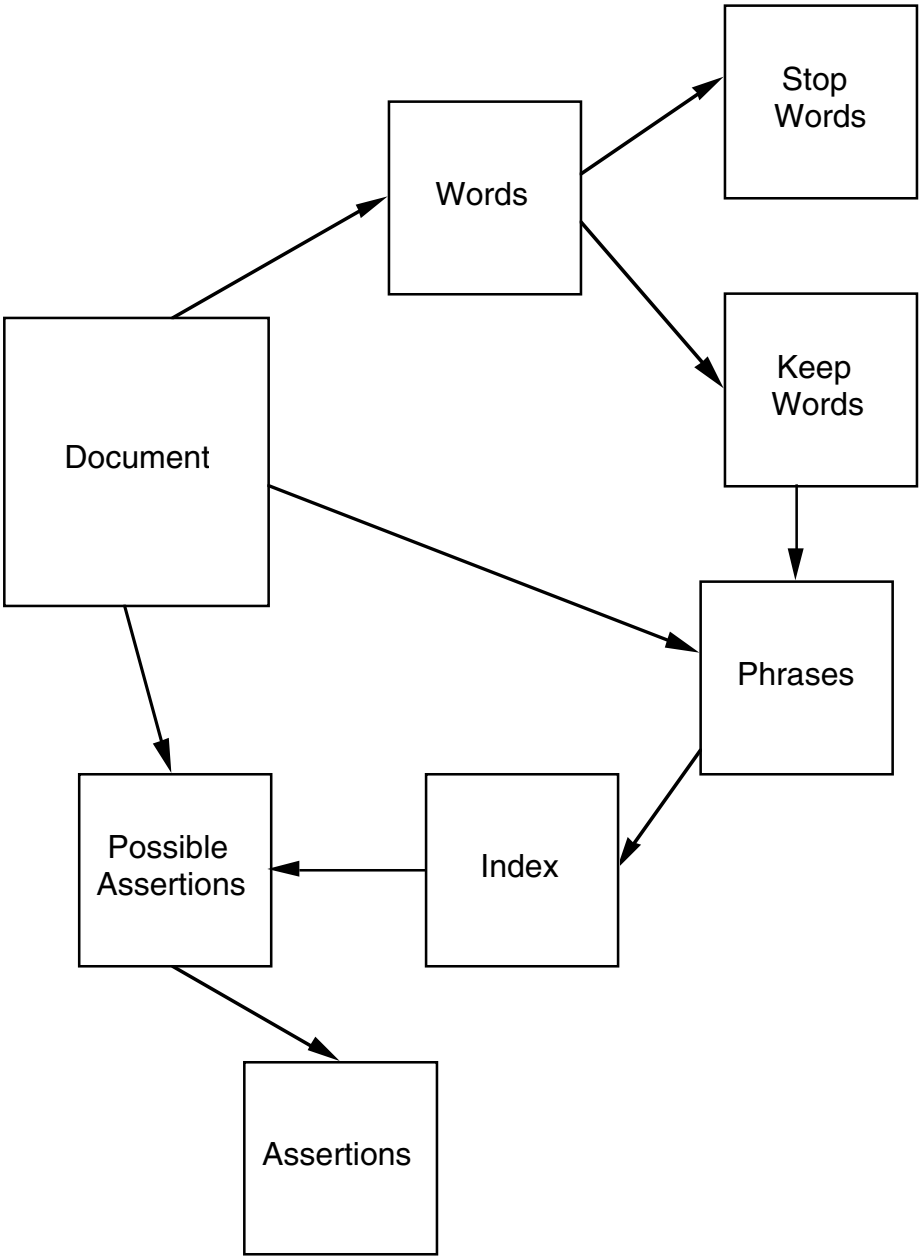

Figure 2 Text Management Tool for Index and Assertions

A crude version of this software was used to assist in the creation of an index to a 200 page book, allowing the author to perform the task in about one day's work. The current version has been used to create an index of about 1000 entries into a set of six research papers in the area of climate modelling.

6. TOOLS TO HELP CREATE THE LINKS

In the prototype 0 system, links were implemented in Hypertalk™ as an integral part of the software, entered by a programmer. In the prototype 1 system, links were implemented using a "sticky button" concept by the domain expert editor, who is able to navigate around the documents, establishing links by point-and-click methods. This method was found to be very satisfactory, so that in the prototype 2 system, the hard

links are created by a similar system which creates linkage records in the database. Dynamic generation of linkages can be seen in systems such as *Intermedia* (Megrowitz 1990). In addition, interactive applications have been created for the database for bulk examination and maintenance of the link records.

Soft links are created and maintained at the database level by the sets of index terms associated with each section, so that this kind of link is created automatically once the index terms have been identified by the editor.

Navigation around the assertions is done by classical text retrieval methods, with the help of the concept model/hierarchical thesaurus. One method used is based on a distance measure derived from document vectors computed from trigrams and tetragrams of characters (Teufel 1988) but functions similarly to the methods described by Salton (1989). Distance between assertions is computed at document entry time, and near documents are assigned hard links to speed retrieval. It is also possible to enter a query and return assertions in order of distance from the query.

7. SUMMARY AND CONCLUSION

We have described the architecture of a hypertext system containing about 100 documents, which is intended to be scaleable to systems with thousands of document in a coherent body of knowledge. The structure schemas require to be populated by editors with domain knowledge, and a number of tools were described which assist these people.

Construction of a practical knowledge management system, however, involves a number of issues beyond the technology:

For a hyperdocument to be useful, the data contained in it must be sufficiently complete, reliable and up to date that the system is safe to use in preference to the existing library or information systems. Since most originators of information now use computer technology in publishing, it would seem sensible to get their cooperation and obtain the information from its source. In many cases, most of the information is copyright, so that it would be necessary to get cooperation from publishers in any case. The system would need to have some way manage intellectual property, as well.

The system will depend greatly for its effectiveness on the quality of the editorial structure added to the raw knowledge. First, this means that a substantial cost must be borne to employ the necessary skilled staff to perform this task. In addition, since the material may in some cases be controversial and will be used by people with differing and sometimes opposed points of view, it will be essential that the editorial function be carried out either by neutral parties or at least parties with declared biases. It is likely that, to be successful, a complex system will need to cater for multiple overlapping editorial agencies.

Many of the users of a hyperdocument will either use several different documents, or have substantial private knowledge bases of a similar kind (or both). It would be useful if knowledge bases from multiple public sources as well as private knowledge bases could be managed using the same system. An organization working on a specific problem could thereby integrate several sources of public knowledge with its private

knowledge. This involves the separation of the information from the software which manipulates it, standard ways of describing structure, and the distribution of the knowledge among many computer systems with good inter-system security. It will also require that the architecture be adapted to deal with the notion of *context*: an item of information will mean quite different things to different people depending on their viewpoint and on what other things the item is to be related to.

ACKNOWLEDGEMENTS

Professor Ann Henderson-Sellers and Mervyn Jones of the School of Earth Sciences, Macquarie University as well as Robert Rottenbury and David Charlton of CSIRO, who contributed much of the knowledge; and to Rosemary Irrgang and Kai Foong of CSIRO, who developed much of the demonstration software.

REFERENCES

Colomb, R.M. and Jones, M. (1989) "Managing the Knowledge Needed to Manage the Greenhouse Effect" *Greenhouse and Energy Conference* CSIRO Institute of Minerals and Energy, Sydney Australia.

Egan, D.E., Remde, J.R., Gomez, L.M., Landauer, T.K., Eberhardt, J. and Lockbaum, C.C. (1989). Formative Design Evaluation of SuperBbook *ACM Trans Inform Systems* **7** (1) pp. 30-57.

Glushko, R.J. (1989) "Design Issues for Multi-Document Hypertexts" *Hypertext'89* pp. 51-60.

Jansen, R. and Robertson, J. (1989) *Management of Wool Dark Fibre Risk Using Hypertext* Technical Report TR-FD-89-05 CSIRO Divison of Information Technology, Sydney, Australia.

Jones, R. (1990). The AIDA project: Research into automatic document analysis. *Australia-Japan Joint Symposium on Natural Language Processing*. University of Melbourne, Australia pp 57-65.

Maurer, H. and Tomek, I. (1989) *Hypermedia in Teleteaching* Report 277, Institutes for Information Processing, Graz, Austria.

Meyrowitz, N. (1990) "The Links to Tomorrow" *UNIX Review*, March 1990, pp 50-67.

Regoczei, S. and Hirst, G. (1989). On extracting knowledge from text: Modelling the architecture of language users. *Proc Third European Workshop on Knowledge Acquisition for Knowledge Based Systems* , Paris, France, pp 196-211.

Robertson, J. (1990) "Information Management, Expert Systems and Hypertext" *Libraries and Expert Systems Workshop and Conference* Charles Sturt University, Wagga Wagga NSW Australia.

Salton, G. (1989). *Automatic text processing*. Addison-Wesley, Reading, Massachusetts

Shipman, F.M., Chaney, R.J. and Gorry, A. (1989) "Distributive Hypertext for Collaborative Research: The Virtual Notebook System" *Hypertext'89* pp. 129-135.

Teufel (1988) "Statistical n-Gram Indexing of Natural Language Documents" *Int. Forum Inf. and Docum.* Vol 13, No. 4, pp 3-10.