Context & Knowledge Types vs Serendipity[1]

Bob Jansen† & George Bray*

† CSIRO Division of Information Technology
PO Box 1599
North Ryde
New South wales 2113
Australia
Phone: +612 887 9307
Fax: +612 888 7787
email: jansen@syd.dit.csiro.au

* SAND Consulting
23 Illinois Street
Sunnybank Hills
Queensland 4109
Australia
Phone: +617 344 3267
Fax: +617 839 0090
email: bray@syd.dit.csiro.au

Abstract

This paper describes various types of knowledge and postulates the notion of knowledge types as a method for integrating various types of diverse and heterogeneous knowledge. It defines knowledge types as an integration of the physical representation and the visualisation and behaviour of chunks of knowledge. It demonstrates this hypothesis by considering the electronic publication of conventional documents in an environment called IntelliText. IntelliText is implemented as a hypertext of basic knowledge types that can be coupled to form more complex knowledge types. The paper focuses on two knowledge types in particular, context and assertions, and shows how they are implemented in IntelliText using the path and note knowledge types. These two knowledge types are singled out in particular as are claimed they contribute most to the improvement of relevance and appropriateness of knowledge.

Keywords

context, electronic book, path, knowledge type, knowledge acquisition, knowledge representation, explanation, assertion, IntelliText

---

[1]The work reported in this paper has been supported in-part by a grant from the Department of Industry, Technology and Commerce (DITAC) of the Australian Government. under the FAIR/ISTAC program and in-part by a grant from the European Software Factory (ESF) as a contribution to a collaborative project between CSIRO, Australia, and the Institut National de Recherche en Informatique et Automatique (INRIA), France..

1. Introduction

The ever increasing explosion in the amount of knowledge available to a problem solver leads to the predicament that the problem solver has difficulty in 'seeing the wood for the trees'. One needs only to consider the eruption in knowledge-provider services as exemplified by the Minitel system in France, the Telecom 0055 service in Australia, or the Internet world-wide electronic information system, to gain an appreciation of the increasing scope of this problem. Experience however shows that more knowledge *per se* does not aid the problem solving process, and in fact in many instances the reverse is often the case. The access to ever increasing amounts of knowledge poses greater problems to problem solvers in that they are under greater pressures to retain an overview of what knowledge is available, relevant, and where to access it. Where problems require the integration of knowledge covering different disciplines, no one person can hope to achieve an optimal level of knowledge except in small focussed areas.

We are developing methods for improving the relevance and appropriateness of knowledge brought to bear on a problem (ie. the quality of the knowledge, the right knowledge in the right context) in concert with access to a greater diversity of knowledge. Improving the quality of knowledge necessitates filtering the morass of knowledge seemingly to pertain to the problem to cull out that knowledge deemed irrelevant. This culling process will be facilitated by appropriate visualisations of the knowledge in the right context[2]. No single existing technology solves the problem for the decision maker given that different problems require different types of knowledge. The problem seems to be so complex requiring a range of different technologies and different types of knowledge. Examples of tried technologies include decision support, free text retrieval (eg. Status, Stairs, Basis, etc.), artificial intelligence (ie. Mycin, Prospector, etc.) and more recently hypertext/hypermedia.

This paper will discuss the first set of results from research in providing access to knowledge from documents as part of the knowledge acquisition phase of 'intelligent' systems development (ie. providing the problem solver with access to the descriptive or narrative knowledge of the domain of discourse). The work has concentrated on the unearthing of the knowledge types used in scholarly publications and the subsequent building of an environment, *IntelliText*, for the storage, representation and visualisation of the knowledge found in such publications in concert with computational representations of this knowledge supporting various processing functions (eg simulation models, inference engines, etc).

Section 2 discusses knowledge types and their relationships to cognitive models of the document that an author may create and a reader may build up during their interaction with the document and introduces new knowledge types associated with electronic books. Section 3 introduces the IntelliText environment. We discuss the implementation of context using the *path* knowledge type and show the implementation of the complex *assertion* knowledge type. Section 4 will summarise related work in this area whilst section 5 will mention some future directions for research. A conclusion closes this discussion.

2. Types of knowledge and knowledge types

The existence of differing types of knowledge is not belied by researchers today. However there has been little, if any, attempt to formalise the notion of 'types of knowledge', and yet the application of type theory (Cardelli & Wegner 85) to this area would, in our view, bring similar benefits to the general understanding of knowledge and its various functions as found with data types in programming languages.

Knowledge can be classified into various categories according to the way chunks of knowledge may be utilised (Dolk, forthcoming). Particular categories include procedural, factual, descriptive and inferential. *Procedural knowledge* is that knowledge pertaining to the accomplishment of some task. It is usually represented as executable instructions (ie code in a computer, set of directions on paper, etc) and is characterised by a temporal component, a sequence of steps that may be followed to accomplish the required task. *Factual knowledge*

---

[2] We introduce the notion of knowledge visualisation into this aspect of the problem solving process as we believe that the relevance and meaning of knowledge becomes more easily apparent by presenting it in a fashion that highlights its relationship to the problem being solved in a way easier for the user to relate to.

establishes the existence of a particular state of the domain, the existence of a particular piece of knowledge in the domain, or the validity of a belief held about the domain. This knowledge is usually in the form of assertive statements. *Descriptive knowledge* is a form of knowledge used for describing situations, conditions, etc. It is usually natural language in the form of text, but includes other media types such as video, audio, and graphics in the form of pictures and tables, etc. *Inferential knowledge* is a form of knowledge used in the creation of new factual knowledge from existing factual knowledge. This is the form of knowledge usually associated with expert systems or knowledge based systems, in concert with factual knowledge. It is captured during the knowledge acquisition phase of knowledge based systems development and represented in a computational representation for processing by the inference process. The inference process (eg *Modus Ponens* or *Modus Tolens*) itself is a form of inferential knowledge.

Knowledge can be classified as *formal* or *narrative* (Kornell 87).*Formal thought* is that type of thought that can be represented in a formal framework (eg logic) whilst *narrative thought* is the rest, the hunches, experience, feelings, etc. The formal thought can be acquired and represented within the computational framework, ie the expert system, whilst the narrative thought remains firmly in the expert's head although it may be instantiated in objects such as books, films, etc.

Both views cited above recognise; a *computational knowledge scheme*, the formal or procedural, inferential and factual knowledge; and a *cognitive knowledge scheme*, the narrative or descriptive knowledge. Our research (eg. Jansen & Robertson 89) provides sound reasons for the recognition of both schemes and their potential if used within knowledge based systems. This work examined the potential of the computational representation of the domain knowledge to provide suitable explanations to an end user. For example, the work indicated that on a set of test questions supplied by end users, the computational representation could only provide answers for approximately 20% of end users queries, assuming that a suitable parser was built and available to translate between the different languages. Thus 80% of end users queries were insupportable from the computational representation, the expert system itself. One conclusion reached was that the descriptive, or narrative, knowledge would suffice to answer closer to 100% of end users queries if it were available at run-time[3].

We have adopted the notion of knowledge types as a formal representation of the different types of knowledge and propose an albeit vague definition as follows:-

a knowledge type is a complex conceptual object that encapsulates the representation, behaviour, and visualisation of a chunk of a particular type of knowledge.

For example, a book can be considered a knowledge type[4], it represents a certain type of knowledge in this case descriptive or narrative knowledge. It uses several schemes, namely natural language in the form of text, and graphics in the form of pictures and tables. It has a specific representation, namely marks on sheets of paper bound together into a specific sequence. It has certain functionality[5], ie browsing by page turning, as well as supporting various visualisations, ie text within pages, hard or soft cover, contents page, indexes, etc. Books also have a structure and although this is specific to each book, common features include titles page, chapters, chapter headings, pages, and headers and footers, etc. Smith 84 details various types of books differing from the western style codex. As a representation, books are cognitively efficient but computationally inefficient.

Another knowledge type is executable code as found in computers (ie. procedural or computational knowledge). Its representation, ie machine code instructions in the form of a binary code, can be executed directly. It can have several visualisations, but in its executable

---

[3] This situation should be contrasted with a face-to-face confrontation with the expert where the narrative knowledge as well as the formal knowledge is present. The expert can usually answer all of the user's queries - in fact the expert can tailor the contents of their discourse to suit the requirements and knowledge of the end user.

[4] By coincidence, it can also be a knowledge representation or a knowledge visualisation depending on the user's context.

[5] Books also have many other uses unforseen by the author. Some books shore up tables, keep doors open, are carried to signify the owner is busy, are left on coffee tables to indicate a characteristic of the owners, etc.

form these are generally not very helpful except to maintenance programmers. Higher level visualisations are available in the form of flow charts, 'high level' languages, etc. Executable code itself is cognitively inefficient but computationally efficient whereas some of its alternative visualisations can be more cognitively efficient.

2.1. Knowledge type representation and visualisation

A problem raised in the simple examples of a book and computer code above is the difference between a representation and a visualisation[6]. We hypothesise that knowledge types can have many visualisations and behaviours but only a single representation (Figure 1). The representation is the physical implementation of the knowledge, the visualisation and behaviour the interface level of the knowledge, and the knowledge itself is at the conceptual level. Each unique chunk of knowledge in concert with the type of knowledge represented is an instance of a knowledge type with its possible visualisations and behaviours and mode of representation inherited from the generic knowledge type definition[7]. In this simple scheme, a book could be said to be the visualisation of a 'book' knowledge type, the text and graphics the representation on the storage medium of paper which in conjunction with the implicit knowledge content form the 'book' knowledge type. However, care should be taken as it is quite valid to see the book as a representation, (eg. the representation of knowledge about books in general).

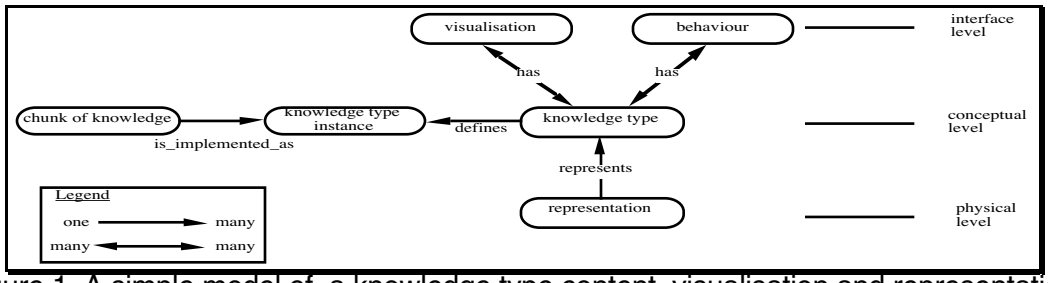

Figure 1. A simple model of a knowledge type content, visualisation and representation.

As an example of the difference between visualisation and representation, consider a piece of text, say this paper, which is represented as words on paper, but can be 'visualised' as either words on paper or sounds if the words are read out loud. In this scenario, the audio form is a visualisation and not a representation. Alternatively, the representations could be both text and an audio file with visualisations of text and audio. The choice of the former is predicated on a suitable translator for converting the stored text form into the audio form.

The use of different representations and visualisations enables the end user to interact with the knowledge in a manner appropriate to them and their problem, ie their context. Despite the multiplicity of representations and visualisations, each chunk of knowledge is normalised (eg. Debenham 89) and thus mechanisms should exist for translating between the different representations and techniques should exist for providing the required visualisations.

2.2. Complex knowledge types

Knowledge types can be complex and built up from other knowledge types. For example the book knowledge type could be composed from chapter, section, paragraph, sentence, word, picture and table knowledge types. In addition, the book knowledge type can contain citation and reference knowledge types. In this scenario, each knowledge type provides different knowledge to a reader of the book, and in turn, is used by the author and reader for different purposes. Figure 2 shows one model of the complex *research paper* knowledge type.

2.3. Assertions as a knowledge type

---

[6] At one level, a representation can be thought of as the visualisation 'seen' by the computer (Bob Colomb, University of Queensland, personal communication)

[7] Note however, that as in IntelliText, individual knowledge types may have non-default visualisation and behaviours under certain instances.

A previous study (Jansen & Robertson 89) hypothesised and demonstrated the usefulness of a knowledge type labelled an *assertion*, especially in supporting the explanation of complex knowledge. In this study, an *assertion* was defined as "a statement of interest an author makes in a research paper". Thus another model for the book knowledge type is a set of assertions representing the points the author is attempting to convey in the publication (Figure 3). As a sequence, the assertions may be one component of the rhetorical structure used by the author, hence the *supports* and *contradicts* relationships.

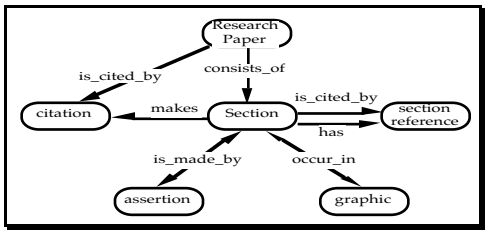

Figure 2 - one model of the research paper knowledge type.

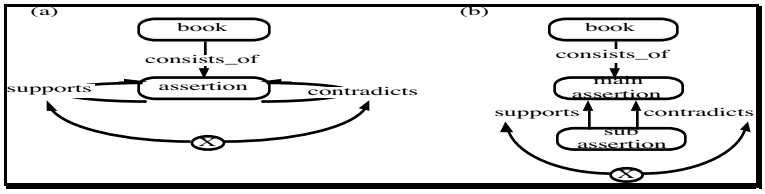

Figure 3 - two richer models of the book knowledge type based on the *assertion* knowledge type. Note, the circled 'X' indicates a mutual exclusivity constraint, in other words, an assertion may not support AND contradict another assertion at the same time. Model (b) on the right indicates that a book consists of one major assertion and several sub assertions either supporting or contradicting the main assertion[9].

2.4. Context as a knowledge type

The difficulties with knowledge type representation/visualisation begin to be clarified when one introduces the notion of context. For example, context would be used to specify the situation(s) in which something was a visualisation or a representation. However, two main problems associated with context are: there is no universally accepted definition; and there is little knowledge available as to its representation, despite many disciplines having built up impressive knowledge on its use[10].

Various definitions of context exist, for example: context provides meaning and structure to a domain; "a set of entities (things or events) related in a certain way; these entities each have a character such that other sets of entities occur having the same characters and related by the same relations; and these occur 'nearly uniformly'." (Ogden & Richards 46); or a definition in terms of a 'compresence' relationship and relating the set of compresence relationships in a region of space-time, a 'compresence-complex', to events, objects, perception, etc. (Russell 92). These definitions embody context as a set of things and relationships between the things which can be simply represented using the notion of a Herbrand Universe (Herbrand 67).

Context however has a dynamic as well as a static character and in its dynamic form, may be more akin to a *path*, a sequence of places leading from a start place to an end place. A path should be regarded as one would a path through a forest, it represents a predefined route through the forest to cater for specific requirements, ie it may be the shortest path or the most scenic one, the scent trail, or the colour trail, etc. Paths are created for a specific purpose, but may be used for many unrelated purposes. Paths can cross and the user of a path may join the path after its designated starting place and leave before the designated end place. In this scenario, paths could be used to implement both the static and dynamic nature of context

---

[8] Although in reality it is probably a sequence

[9] From work by Charles Chung of the CSIRO Division of Information Technology.

[10] To make matters worse, we believe that the definition of context is itself context sensitive (ie. it can only be defined in terms of itself) and that possibly context obeys a 'Heisenberg Uncertainty principle' in that the act of perceiving it changes it.

in a single object, a *context knowledge type*. As an implementation of context each path has a reason for creation, as well as other attributes such as a creator and a creation date. These would define the *signature* of the path, the context wherein the path should be activated (ie. either because the reason for creation matches the nature of the problem, the creator is a person of standing in this problem domain, etc).

This notion of 'contexts as a set of paths' has its roots in the work of Minsky (Minsky 86). Minsky postulates the notion of *knowledge lines* represented as unique navigation paths through a knowledge space with each line representing some chunk of knowledge. A knowledge line encapsulates a time dependant sequence of places that represent some chunk of knowledge (eg. the knowledge to solve some problem).

3. IntelliText

IntelliText is an environment for the storage, representation and visualisation of the narrative or descriptive knowledge found in scholarly and technical publications. IntelliText provides an author with multimedia facilities for representing their knowledge and the reader with multimedia facilities for browsing the publication, examining the author's knowledge, and annotating the publication with their own views of the knowledge or in fact their own knowledge. In the latter cases, the reader becomes an author[12] in their own right producing their own 'books' from publications provided by other authors or input by themselves. IntelliText aims at representing static books as active objects, whereby all objects comprising a book can respond to stimuli from the reader. For example, figures, graphs and tables can be stored as application files and triggered from within IntelliText, thus presenting the reader with an object manipulable with all the functions supported by the application (eg. graphs stored in data form as tables within a spreadsheet program).

IntelliText retains and extends the 'book' metaphor across the system interface. This decision was made in line with a comment by McLuhan that every new successful technology is an iterative development of existing technology and carries with it the metaphors from the existing technology (McLuhan 74). Thus IntelliText, having been developed to cater for novice users, attempts to hide the technicalities of typical hypertext and multimedia environments in an electronic simulation of a book. It is hoped that this decision will make it easier for novice or infrequent users to interact with IntelliText.

3.1. IntelliText, knowledge types and bookmarks

IntelliText is implemented as a hypertext of instances of the basic knowledge types, *section, note, word, figure, movie, reference,* and *path,* each defining a representation for its content, two possible visualisations of the content, and two possible behaviours for when the knowledge type is accessed. By default, *figures*, *notes*, *words*, and *paths* will be presented in a separate window, whilst *references* will trigger the appropriate external application. In addition, we have implemented a complex knowledge type, the *assertion*, by combining the functions of the *note* and *path* knowledge types[13]. Each instance of a knowledge type is termed a *bookmark* and is the basic unit of storage and retrieval.

IntelliText implements two elementary visualisations for knowledge types, the *text* view, displaying the content in a textual representation, and the *map* view, displaying a graphic representation. In the *text* view, the phrases with additional related information are marked with a *signpost*, a placeholder containing pointers to other bookmarks storing the additional information. In the *map* view, bookmarks can be represented diagrammatically showing relationships between them. In addition, the *map* view can be annotated with extra

---

[11] IntelliText is the result of a collaborative project between the CSIRO Division of Information Technology and Academic Press (UK) Ltd. The aim of this software is to investigate the usefulness of hypermedia technology in concert with knowledge representation methodologies for the production of electronic publications. It is planned to use the IntelliText environment in Knowledge Based Systems to enable the provision of appropriate explanations and justifications to end users. The first commercially available 'IntelliText' will be Jansen & Compton 93.

[12] Note that the terms 'reader' and 'author' refer in this case to the function being performed by an end-user rather than the traditional demarcation associated with conventional documents.

[13] There is no limit to the number of complex knowledge types that can be defined, as long as their definition is based on the available seven basic types. New basic types would require extensive design and programming of structure, visualisation and behaviour.

information. Bookmarks in both views are active and respond to mouse events. This scheme is depicted in figures 4 & 5. One anticipated important use of the map view is the representation of the structure of the author's knowledge. For example, the map view could be used to graphically depict the author's rhetorical structure used in a particular section of text.

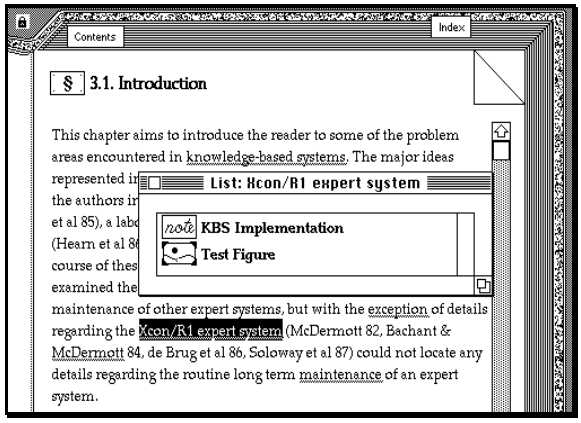

Figure 4 - A *section* bookmark showing the text visualisation. Note the underlined chunks of text, each defining an 'signpost', a pointer to one or more bookmarks. The window shows the multiple bookmarks associated with the opened signpost.

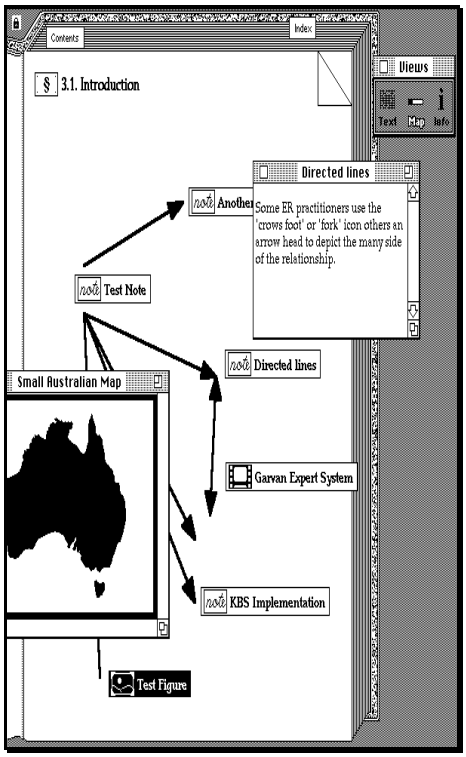

Figure 5 - a map view of a bookmark such as that shown in figure 4. Active views allow the reader to move freely and associatively using the appropriate representation. Here, the Map view shows how the Author's knowledge is organised.

IntelliText enables an author to define: a document as a sequence of bookmarks; the specific behaviour and visualisation for each bookmark; and signposts to facilitate non-sequential navigation between bookmarks or to annotate the existing contents. The reader 'opens' a bookmark, either by browsing, by sending it a specific 'open' request, or from a signpost, to view its content, related notes, information, etc.

3.2. The implementation of context

Loss of context is a typical problem with hypertext and can be experienced when a multi-hop link to other information is followed. Often, the screen changes completely and does not allow reference back to the source. IntelliText has a number of features which we believe will together minimise this 'lost-in-hyperspace' predicament. These features, which we believe are associated with context, incorporate the basic knowledge type, the *path*; the use of multiple windows thus presenting requested information in discernible layers; and the *current path* facility, a time-ordered sequence of all bookmarks visited in a session. During an IntelliText session, all windows will remain visible unless specifically closed or hidden by the reader. More importantly, the state of each window will reflect its state when last activated, thus if the reader selected a bookmark from a *path* window, the selected bookmark will remain highlighted (Figure 6). As well, the *windows* menu contains a list of all active windows, visible

and hidden, in the order of their creation, and the current path can be formatted to show where the reader branched off from a path by indenting such diversions (eg. the path window in figure 7). In this fashion IntelliText supports the reader in remembering their reasons for why they requested more information and the sequence in which the information was presented, important aspects of context.

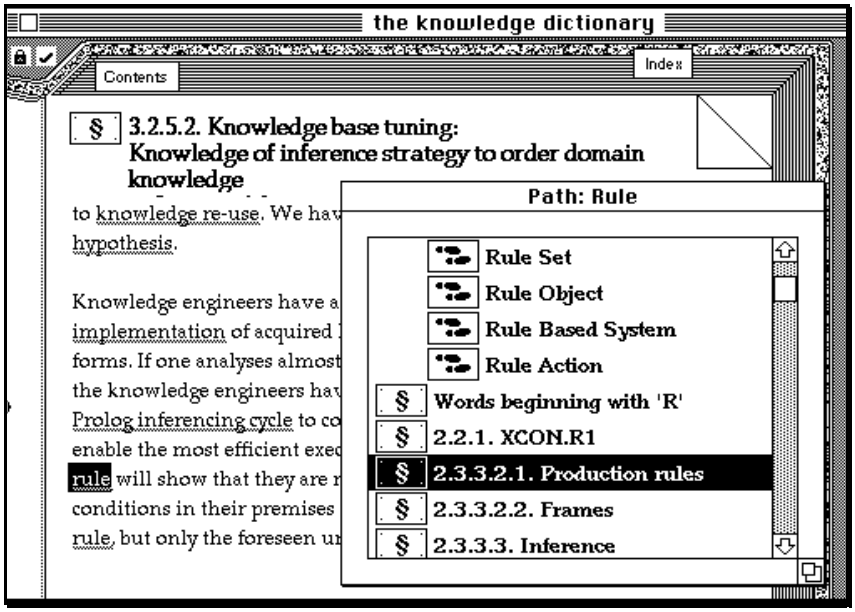

Figure 6 - A path window layered above the bookmark from where it was invoked. Note, the highlighted signpost 'rule' which was activated to show the path. This path contains other paths and thus indicates that other paths cross this path. The window also shows the sequence of places on the path.

The *path* knowledge type, as an implementation of context, enables the reader to be guided to that subset of bookmarks of the domain that the creator of the path believed were appropriate and relevant. 'Someone has created this path, so it must lead somewhere'. 'This path was created for a purpose so if this matches my current purpose then I should follow it'. *This is a much more accurate way of navigating - the knowledge represented by a path has more value because it already has been processed or assessed*. The implementation of the *path* means that in IntelliText there is no notion of a hypertext network as found in current generation hypertexts. An IntelliText domain can have a myriad of pathways each defined for a particular purpose, but each able to be used for any suitable purpose (Figure 6).

The user selects a path from those available using the paths' signature and then proceeds to follow the path until they select to stray or they reach their destination. If no matching signature can be ascertained, then there are several possible courses of action available for the reader. A *default path* is always available. This is the path created by the author in the creation of the publication, the *table of contents* associated with any book[14]. The reader could however invoke a conventional 'find' function specifying suitable selection criteria for matching against the contents of available bookmarks. Alternatively, the reader could go to any random place in the hope that a suitable path can be found or a more suitable navigation strategy is presented.

'Random access' is common in conventional systems. Serendipity (and hence the chance to get lost) is forced upon the reader - there has been no cognitive assessment of the value of an 'information sequence'. This situation is workable in a conventional publication because extensive cognitive stimuli exist to aid the reader in determining their place in the publication and there is only a single supported path which is also the physical browsing sequence. For example, the reader of a conventional book can determine qualitatively their relative position in the book with respect to the start or end by comparing the thickness of all pages on the left and right of the book. In electronic publications however, these cognitive stimuli are generally unavailable.

A further aspect of the path knowledge type as implemented in IntelliText is the ability of the reader to create a new path from the current path (ie. the places they have visited) or select a concept and create a path for it. The correspondence of paths and context allows the

[14] IntelliText retains the table of contents as an interface element although it is a path, albeit a special path, the author's path.

reader to update the available contexts to cater for unforseen situations. This facility supports the reader in tailoring IntelliText to suit their circumstances. It can also be useful for educators preparing a passage for their students. Teachers "approve" information in this way and give it greater value.

3.3. Assertion - a complex knowledge type

Assertions represent the 'interesting' or important points that the authors are trying to make and thus form an extremely valuable entry point into the domain. To assist entry, they are indexed in their own right whereby the reader can select an assertion from an index and see the complete path for that assertion (ie. the bookmark with the full assertion text and all places where that assertion is made) (Figure 7). Readers can create their own assertions, thus highlighting the points of interest to them. The relationship between assertions can be visualised using the map view, and thus while the text of an assertion can indicate the relevance of its source bookmark to the reader, the map view might indicate the appropriateness of the assertion, and hence its source bookmark, for the reader in the light of the overall domain.

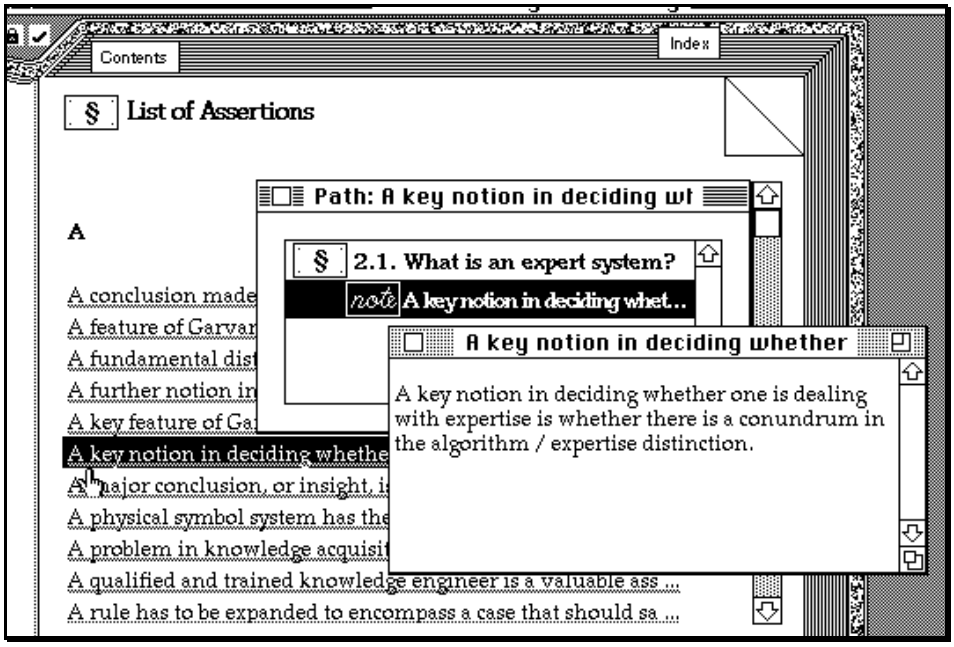

Figure 7 - this shows the implementation of assertions in one IntelliText. An index of all assertions has been created, with the first 60 or so characters of the assertion text as the index entry. The action of selecting an entry causes the Path window to appear from which the user can select either the note entry being the text of the assertion of places where the assertion is made (in this case section 2.1).

4. Related work

As an exercise in hypertext or multi-media, there is much related work to that found in IntelliText (eg. Xanadu (Nelson 87), the Hyperbook (Thomas 92), DynaText (DeRose 90), the work on Hypertext and paper (Rada 91), Ebook3 (Savoy 89), etc). Most of this work is aimed at producing hypertexts for the electronic manipulation of documents. IntelliText however, is aimed at the task of acquiring knowledge from documents and hence its primary focus is on the types of knowledge authors convey via the authoring process (see also Candlin & Saedi 83), its representation in conventional books, and its representation and use within electronic knowledge sources. IntelliText does however address the problem of hypertext navigation, or the 'lost-in-hyperspace' syndrome, by its use of the path knowledge type as the central backbone for supporting navigation.

Ruqian & Cungen 90 describes work in the area of knowledge acquisition from texts by the use of grammars to describe knowledge in the following types: meta information; causalities; planning knowledge; functions and mechanisms of physical objects; definitions and properties of physical objects; associations between attributes; goal descriptions; classification examples; and knowledge tables. In its recognition of different types of knowledge, the grammar extends the concept of knowledge types as espoused in this paper. IntelliText however has focussed on those knowledge types more common in documents with the aim

of determining a generic knowledge type definition so as to enable the reader to create any required knowledge type without recourse to low level environment programming.

Shasha 85 describes a data model to support knowledge exploration, drawing inspiration from database theory, hypertext systems, knowledge representation and a study of textual fragments, called fragment theory. It poses interesting questions regarding the difference between hiring a consultant and reading their book, as well as describing the Netbook environment, an attempt to replace books and documents by a collection of text fragments. The focus on replacing books and the reliance on text fragments is one major difference between this work and the IntelliText research. It is interesting that this work describes the notion of an information unit as the basic constituent of a text fragment, each unit representing an idea or a fact. Using this definition, an information unit is a subset of the knowledge types implemented in IntelliText.

5. Future work

The work reported in this paper has led to many opportunities for future work including the unification of knowledge types and models, the nature of the authoring process, and context. These three areas are probably the most significant in the development of the next generation of knowledge acquisition hypertexts.

In terms of unifying knowledge types and models, we can define knowledge types, ie a book, and draw models describing their structure and behaviour. However, it seems that the models of the knowledge types may themselves be knowledge types. We postulate that any unification of this sort hinges on the development of a reflexive paradigm for knowledge and is, at least in our mind, related to the question of context. The work in developing IntelliText highlights the complex nature of the authoring process and how much we as readers are educated in the use of current document formats and media. The existing structure of books can be attributed to the media used in their implementation, a media that does not, *per se*, encourage associative access[15].The question that remains unanswered by this and related research is how much the adoption of the new electronic media will alter the structure of books and the authoring and reading processes. The nature and use of context within computer systems is an emerging area of increasing research importance that aims to support the contextual nature of knowledge, a nature that for now has been ignored by knowledge base systems developers. The relevance of any chunk of knowledge, its appropriate visualisation, its meaning are all issues linked to the context of the knowledge. We discussed the implementation of context in a path, but after little time it becomes clear that this is only one form of context. Much work remains to be done in uncovering the nature of context so as to enable its use in Intelligent Systems.

6. Conclusions

The work in developing IntelliText has shown that: the notion of knowledge types is implementable in non-trivial systems; conventional documents can be broken up into a sequence of knowledge types although converting an existing publication is not trivial; support for knowledge type recognition must be provided for during the authoring process of new documents; and there must be methods for assisting the automated conversion of existing documents. The authoring of electronic publications based on knowledge types places greater onus on authors to design their publication prior to producing the content. Paybacks include an improved ability to re-use existing chunks of knowledge in other publications and the explicit definition of assertion knowledge types as part of the document's rhetorical structure.

The use of knowledge types to represent electronic documents provides different models of documents than those proposed for 'electronic authoring'. The models are knowledge-based rather than structure-based although some overlap does arise.

Context as a knowledge type, and its implementation as a path, has provided a simple technique for the navigation of hypertexts that: provides the knowledge deemed most

---

[15] An interesting example of a book written for associative browsing is Horn 89, wherein the author has an extremely busy representation enabling the reader to associatively navigate the book.

relevant and appropriate; minimises the 'lost-in-hyperspace' syndrome associated with conventional hypertexts; and recognises the cognitive value of an information sequence to a reader.

References

Candlin CN & Saedi KL, *Processes of Discourse*, Journal of Applied Language Study, Vol. 1, No. 2, August 1983, pp103-133

Cardelli L & Wegner P, *On Understanding Types, Data Abstraction and Polymorphism*, ACM Computing Surveys, Vol. 17, No. 4, 1985, pp471-523

Debenham JK, *Knowledge Systems Design*, Prentice Hall, Advances in Computer Science Series, Brent RP (ed), 1989, ISBN 0-130516428-1

DeRose S, *DynaText: Electronic Book Indexer/Browser*, EPSIG News, Vol. 3, No. 4, 1990

Dolk D, *A Hyperknowledge Framework of Decision Support Systems*, forthcoming

Herbrand J, *Researches into the Theory of Demonstration* in From Frege to Godel: a Source Book in Mathematical Logic 1879-1931, J van Heijenoort (ed), Harvard University Press, 1967, pp525-581.

Horn R, *Mapping Hypertext*, The Lexington Institute, 1989

Jansen B & Compton P, *The Knowledge Dictionary: A Novel Method for the Engineering of Knowledge Based Systems*, Academic Press, 1993 (in press)

Jansen B & Robertson J, *Management of Wool Dark Fibre Risk Knowledge Using Hypertext*, CSIRO Division of Information Technology, Technical Report TR-FD-89-05, May 1989

Kornell J, *Formal Thought and Narrative Thought in Knowledge Acquisition*, International Journal of Man-Machine Studies (1987) 26, pp203-212

McLuhan M, *Understanding Media*, Abacus, 1974

Minsky M, *The Society of Mind*, Picador (pub), ISBN 0 330 30039 3, 1986

Nelson TH, *Literary Machines*, ISBN 0-89347-055-4, 1987

Ogden CK & Richards IA, *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*, Routledge & Kegan Paul Ltd.(pub) London, 1946

Rada R, *Hypertext and Paper: A Special Synergy*, International Journal of Information Management, 11,1991, pp14-22

Ruqian L & Cungen C, *Towards Knowledge Acquisition from Domain Books*, Current Trends in Knowledge Acquisition, 1990, pp289-301

Russel B, *Human Knowledge: Its Scope and Limitations*, Routledge (pub), 1992 edition

Savoy J, *The Electronic Book Ebook3*, International Journal of Man Machine Studies, 1989, 30, pp505-523

Shasha D, *NETBOOK a Data Model to Support Knowledge Exploration,* , Proceedings of the Very Large Data Base, Stockholm 1985

Smith KA, *Structure of the Visual Book*, Keith A Smith (pub), New York, Book No. 95, 1984

Thomas BH, *Hyperbook*, in Proceedings of the ACS'92 Conference, Hobart Tasmania, 1992