

IntelliText:

An Environment for Electronic Manuscripts

Dr. Bob Jansen
CSIRO
Division of Information Technology,
Macquarie University,
Building E6B,
North Ryde NSW 2113,
AUSTRALIA
email: jansen@syd.dit.csiro.au

Professor Daniel Ferrer,
CNRS
Institut des Textes et Manuscrits
Modernes,
61 rue de Richelieu,
75084 Paris Cedex 02,
FRANCE
email: dferrer@sociologie.ens.fr

Abstract

This paper describes initial results of a project prototyping the use of electronic document software to the development of an electronic manuscript with an intricate genetic history, in which one single physical document (a notebook) records a number of interweaving textual stages. The focus of this discussion is on aspects relating to the reading of such documents rather than their authoring, taking special account of the cognitive and contextual features for such an interface.

Keywords

IntelliText, knowledge types, bookmarks, context, path, assertion, knowledge level, electronic book, electronic publication, hypermedia, hypertext

WORKING DRAFT ONLY

1. Introduction

Traditional editions cannot adequately represent the documents (e.g., writers' drafts and working papers) that genetic criticism uses to reconstruct the creative process that gives birth to the text, because: writers' manuscripts are multi-dimensional objects that cannot be linearised without undergoing severe mutilation; the informational content of the manuscript page is very high and cannot be conveyed by a printed page without either an arbitrary selection of relevant features or a multiplication of diacritic signs that result in practical unreadability; and of the impossibility to relate the genetic material to the inter-textual network in which it is embedded.

The drafts of *Finnegans Wake* offer an interesting case study. There is a wonderful facsimile edition (the James Joyce Archive, 63 volumes and several thousand dollars worth of Facsimile) but scholars find it exceedingly difficult to use, although the editors have resorted to the economically absurd extremity of printing some (costly) pages several times in different sequences.

CSIRO and CNRS ITEM have collaborated on the application of IntelliText, software developed by CSIRO, to the prototyping of an electronic environment addressing these issues. The study focussed on *Finnegans Wake*, Book III, Chapter 4, as an example of a particularly intricate genetic history, in which one single physical document (a notebook) records a number of interweaving textual stages. This paper describes the initial results of this collaboration.

Section 2 will describe relevant aspects of the IntelliText software, section 3 will describe results of prototyping an electronic *Finnegans Wake*. We will close with some points for further study.

2. Description of IntelliText¹

IntelliText is a computer program emulating an electronic 'book' but focussing on the reading and annotation of a book instead of the authoring of its content. Analogous to a real book, an IntelliText book is a single-user book, not a multi-user reading tool².

The electronic version of a corpus will value-add the more traditional paper-based version by:-

- enabling a reader to annotate the corpus electronically,
- enabling a reader to electronically correlate a corpus with other documents in their possession,
- supporting free-text retrieval capabilities thus enabling the reader to open the corpus anywhere based on simple search criteria,
- showing the reader pathways deemed interesting or relevant to particular themes as defined by one of the authors of the electronic version based on their own and/or other experts' viewpoints,
- enabling the reader to create their own pathways through the corpus thus identifying themes relevant to them,
- enabling the reader to subsequently retrieve a pathway and 're-play' it,
- enabling a reader to graphically depict and annotate relationships between objects comprising the corpus,
- enabling the reader to store and subsequently visualise different viewpoints of the corpus, viewpoints that may correlate to other experts' opinions as to themes, areas of importance, etc.

¹This section is based on more detailed descriptions of the IntelliText project in Jansen and Bray (1993) and Jansen (1993).

² Later versions may extend IntelliText into an interactive multi-user tool, see the section 4.

WORKING DRAFT ONLY

2.1. Problem with Electronic Publishing

Electronic publishing is currently the focus of much interest as evidenced in the increase in the number of electronic journals available through networks like the world-wide Internet and the increase in the numbers of titles available via CD-ROM. For example, in the United Kingdom, work has commenced on the upgrading of the JANET network linking research communities. This new 'superJANET' will enable academic journals to be sent around the network in electronic form complete with video clips, 3-D animations, and simulations (New Scientist, 1992). More recently, the 'super-highway', or 'super-tollway', concept as announced in several parts of the world, i.e., the USA, Japan, the European Union, and Australia, aims to foster the access to electronic information, including publications, by ordinary people. In fact, recent discussions indicate that these information highways will not be allowed to disenfranchise any sector of the community.

Although often touted as the answer to the problem of information dissemination and access, electronic publishing currently does little to support the reader of such publications beyond simple search/navigation facilities and 'techno-gadgets' (ie. colour animations, sound, etc). In fact, the most often cited reason for the abandoning of such publications is the 'lost-in-hyperspace' syndrome, where the reader literally gets lost in the space populated by the electronic document³. Where such publications span several documents, (ie the document collection, electronic library, or corpus), the problem is exacerbated by the seemingly lack of structure. This is analogous to the famous 'wood-for-the-trees' syndrome. The second most common reason is boredom, the reader gets bored with the techno-gadgets as they do not intrinsically aid them to read the document⁴.

Current paper-based technology avoids the former problem by relying on an extensive education process wherein we are taught how to read paper-based documents and how to successfully utilise document collections to solve particular tasks. We, as readers of paper-based documents, are used to extensive cognitive support when faced with the problem of retrieving information, browsing, visualising the document space, etc., support provided by the medium of publishing and our education processes. This support is generally not available in current generation electronic publishing systems. Paper-based versions avoid the second problem by not supporting active objects within the document, except for those publications, like 'pop-up' books, where certain objects support limited interaction.

2.2. What IntelliText Offers

IntelliText addresses this issue of reading an electronic publication by maintaining and extending the 'book' metaphor across the system's interface; implementing a 'path' metaphor as the mechanism for creating bounded navigation structures through the document space; and supporting the context-sensitivity of reader interactions wherever possible.

A reader of an IntelliText document, or corpus, is supported with extensive facilities for annotating the document space, for controlled browsing through the document space based on multiple levels of interaction to suit their skills and experience, for creating indexes to various points of interest within the document space, and graphically displaying areas of the document space. The IntelliText interface enables a reader to easily remind themselves why and how they reached a particular place in the document

³ Anne Marie Basset, a researcher of CNRS ITEM, gave the best description of this problem. She described it as a fear of confronting and interacting with a limit-less space, having no notion of location, boundaries, etc.

⁴ Note, that we do not consider technical reasons for the failure of electronic publishing to-date, i.e., poor screens for reading, bulky hardware, lack of true long-life batteries for portability, etc. These problems will be solved by suitable advances in the technology, advances that will be driven by consumer demand.

space. IntelliText presents the requested information in a layered approach where the newer information is layered over existing visible information. Thus the layers themselves provide an audit trail for the reader's actions. IntelliText leaves information 'open' and visible where-ever possible. It is up to the reader to dismiss information objects, not up to IntelliText to guess what may no longer be required.

2.2.1. IntelliText, Knowledge Types and Bookmarks

IntelliText is implemented as an hypertextual navigation network of instances of the basic *knowledge types*⁵(*section*, *note*, *word*, *figure*, *movie*, *reference*, and *path*). Each instance of a knowledge type is termed a *bookmark* and forms the basic unit of storage and retrieval. Bookmarks are identified by a type character and title, as well as defining the representation for their contents and a number of possible behaviours for when the bookmark is accessed. By default, *sections* will be displayed like a page from a book, *figures*, *notes*, *words*, and *paths* will have their contents presented in a floating window, whilst *references* will trigger the appropriate external application. In addition, we have implemented a complex knowledge type, *assertion*, by combining the functions of the *note* and *path* knowledge types⁶.

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. We use the term *signpost* in favour of the usual hypertext term *anchor* because an anchor conveys the notion of tethering onto a single object. Conversely, a signpost conveys the notion of pointing out likely, or interesting, places, so the direction of the information content is outward not inward.

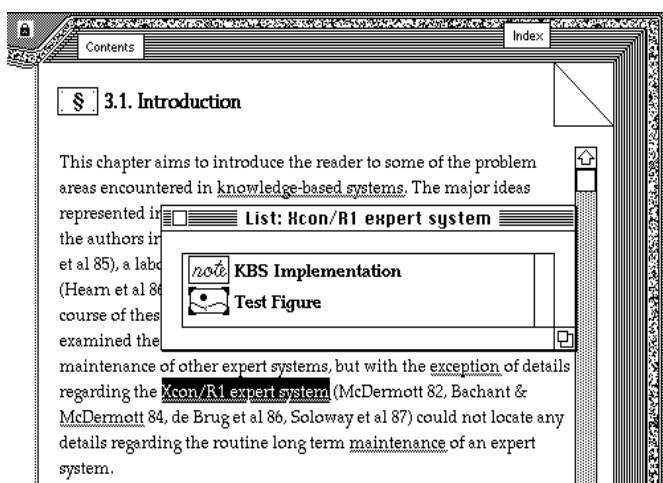


Figure 1 - A section bookmark showing the text view. Note the underlined chunks of text, each defining an signpost, a pointer to one or more bookmarks. The window shows the multiple bookmarks associated with the opened signpost.

In the map view, bookmarks can be represented diagrammatically showing relationships between them. In addition, the map view can be annotated with extra information. *Signposts* in both views are active and respond to mouse events. This scheme is depicted in figures 1 & 2. 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 chunk of text. Once defined, the author of the bookmark selects a default view for the bookmark so that when activated, the bookmark appears using the default view unless

⁵ For a description of the concept of knowledge types, see Jansen & Bray (1993) or Jansen (1993).

⁶ 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. In the current version, new basic types would require extensive design and programming of structure, visualisation and behaviour.

WORKING DRAFT ONLY

overridden by specific behaviour commands (see below). Note that the reader can subsequently alter the default view if required.

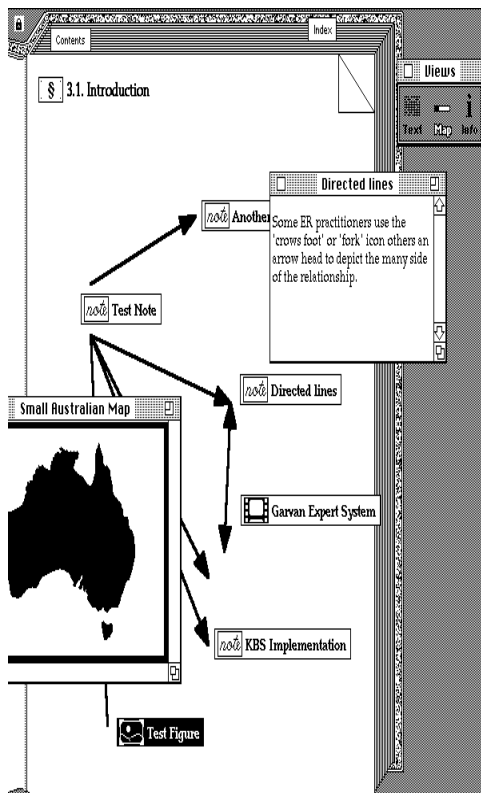


Figure 2 - a *map view* of a bookmark such as that shown in Figure 1. Active views allow the reader to move freely and associatively using the appropriate representation. Here, the *Map view* shows how the author's knowledge is organised.

IntelliText enables an author to define: a document as a sequence of bookmarks; the default and allowable behaviours for each bookmark; a behaviour for each hypertext-link to a bookmark; the default view for each bookmark; and signposts to facilitate associative, i.e., 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.

The use of behaviours, both the default associated with a bookmark and the specific ones associated with each hypertextual-link, enable an author to present the relevant aspect of each bookmark depending on the readers context. For example, to cater for visually-impaired readers, an author could create an audio path wherein all bookmarks speak their contents, or relevant parts of their contents, when activated. This could be accomplished, without duplicating bookmarks, by defining an appropriate 'speak' behaviour and associating this behaviour with each item on the visually-impaired-reader's path. Thus if a bookmark was activated from another path or by another mechanism, the speak behaviour would not be invoked.

2.2.2. 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 negotiated. Often, the screen changes completely and does not allow reference back to the source. IntelliText has a number of features which minimise this 'lost-in-hyperspace' predicament. These features, which we believe are associated with context, incorporate the basic knowledge type, *path*; the use of multiple windows thus presenting requested information in discernible layers; the current path, a time-ordered sequence of all bookmarks visited in a session; a time-sequenced windows menu containing the name of each active window; and the behaviour facilities associated with bookmarks and navigation links.

WORKING DRAFT ONLY

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 (Figure 3), thus if the reader selected a bookmark from a path window, the selected bookmark will remain highlighted. As well, the *Windows* menu contains a list of all active windows, visible or 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. 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.

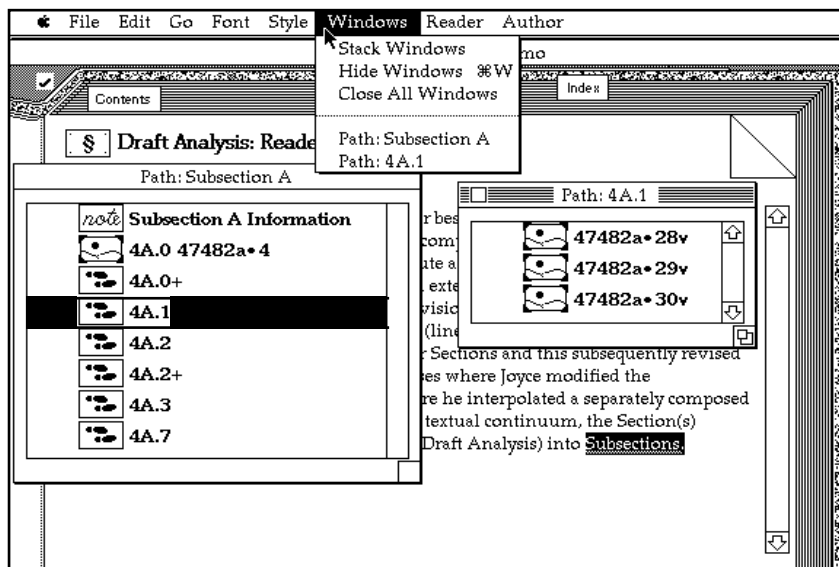


Figure 3 - aspects of context. The currently active window is *Path: 4A.1*, from the shading of the window header bar. All windows indicate their last active state, e.g., the reader activated the signpost *subsections* in the text window and the path *4A.1* in the *Path: Subsection A* window. The *Windows* menu indicates that the window *Path: Subsection A* was opened before window *Path: 4A.1*, but not why. The reason for activation comes from the active state of window *Path: Subsection A*.

The use of different behaviours enables appropriate and relevant parts of a bookmark to be highlighted, or, as described above, the bookmark to be presented in a relevant fashion. This is analogous to the notion of situated cognition (Slezak, 1993) seen more frequently in the AI literature.

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 (i.e., '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 has already 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 4).

The user selects a path, or a number of paths, from those available using the paths' signature and then proceeds to follow the path(s) until they select to stray or they reach their destination. Multiple pathways can be followed simultaneously, by opening each required pathway and selecting relevant locations one at a time. Paths that are no longer suitable to the current task can be closed, analogous to the pruning of a search tree in conventional expert systems.

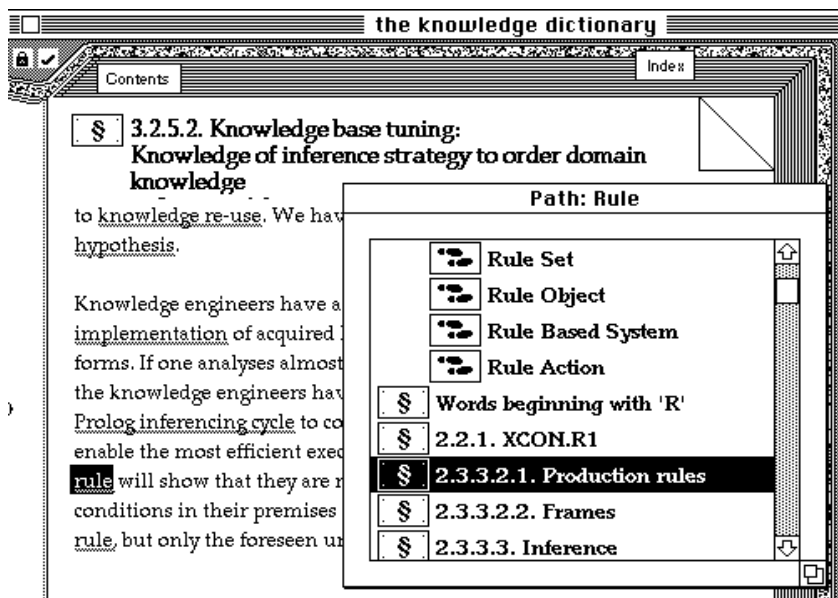


Figure 4 - A path window layered above the bookmark from where it was invoked. Note, the highlighted signpost *rule* which was activated to show the *Rule* path. This path crosses other paths as indicated by the other paths in the *Path: Rule* window. The window also shows the sequence of places on the *Rule* path.

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 book⁷. 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 location 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 unavailable⁸.

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 and signpost a path for it. The correspondence of paths and context allows the reader to update the available contexts to cater for unforeseen 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. IntelliText and Finnegans Wake

The collaboration between CSIRO and CNRS ITEM focussed on Finnegans Wake, Book III, Chapter 4, as an example of a particularly intricate genetic history (see Figure 6), in which one single physical document (a notebook) records a number of interweaving textual stages. Analysis of the notebook shows that Joyce had a complex

⁷ IntelliText retains the table of contents as an interface element although it is a path, albeit a special path, the author’s path.

⁸ IntelliText does offer a ‘progress bar’ window for showing the current location within the book, but there are cognitive and semantic problems related to the physical versus logical arrangement of electronic books and their effect on current location (see section 4).

WORKING DRAFT ONLY

style of writing, a style that used any piece of white space when required for any reason whilst working on many subsections at any point in time. Thus, pages from the notebook are used in multiple draft stages, multiple sub-stories⁹, etc.

1A	1B	1C	1D	2A	2B	2C	3A	3B	4
*0			*0	*0		*0			
*1			*1	*1		*1			
*2			*2	*2		*2			
*3			*3	*3		*3			
*4			*4	*4		*4	*0		
					*0		*1		
					*1		*2	*0	
5			5	5	2	5Δ		*1	
					*3			*2	
6			6	6	4	6	*3	*3	
7Δ			7Δ	7Δ	5	7Δ	4Δ	4Δ	
(8)			8Δ	8Δ	(6)	(8)	5	5	
							6Δ	6Δ	
9			9				7	7	
	*0	*0					8Δ	8Δ	
10Δ	1	1	10				9Δ	9Δ	*0
(11)	(2)	(2)	(11)						
		(3)						10	
								11	*1
		4Δ						12	*2
		5		9	7	9		13	(3)
		(6)		10Δ	8Δ	10Δ		14	4Δ
		7Δ		11Δ	9Δ	11Δ		15	5Δ
		8Δ		12Δ	10	12		16Δ	(6)
12Δ	3Δ	9	12Δ	13Δ	11Δ	13Δ	10Δ	17Δ	7Δ
13Δ	4Δ	10Δ	13Δ	14Δ	12Δ	14Δ	11Δ	18Δ	8Δ
(14)	(5)	(11)	(14)	(15)	(13)	(15)	12Δ	19Δ	9Δ
15		12	15	16	14	16	13	20	10

4A	4B	4C	4D	4E	4F	4G	4H	4J	4K	4L	4M	4N	4P	4Q	4R	4S	4T
*0						*0		*0	*0		*0	*0					
	*0		*0	*0	*0		*0		*0+		*0+	*0+	*0	*0	*0	*0	*0
*0+		*0				*0+				*0			*0+		*0+		*0+
*1												*1	*1	*1	*1	*1	*1
															*1+		
															*1‡		
*2	*2	*2	*2	*2	*2	*2	*2	*2	*2	*2	*2	*2	*2	*2	*2	*2	*2

Figure 6 - draft stage history of Finnegans Wake, adapted from Hayman & Rose (1978). Table 1 shows overall state whilst table 2 shows subsection integration for chapter 4 only. The meaning of the symbols is as follows: the number indicates the draft stage, * indicates a holographic form witness, + indicates re-drafted pages of the draft, ‡ redrafted pages of draft +, Δ indicates an incomplete witness, and the parentheses indicate the material related to multiple drafts.

The project decided, initially, to follow the layout of the Finnegans Wake Archive (Hayman and Rose, 1978) and then to build the various alternate pathways for accessing the archive, e.g., by subsection, draft stage, etc¹⁰. This strategy supports access by those researchers familiar with the structure of the archive, whilst tantalising them with

⁹ Finnegans wake is in fact a collection of stories bound together by Joyce to form final book, akin to, for example, Chaucer's Canterbury Tales.

¹⁰ Note, that for copyright reasons, we built the object structure mapping each page of the notebook into an IntelliText bookmark, but only added the contents of selected pages where testing of specific functions or interface requirements was necessary.

WORKING DRAFT ONLY

alternate access methods available through use of the electronic medium. The completion of the navigation structure would be the responsibility of the researchers themselves, creating new, and editing existing, navigation structures as appropriate¹¹.

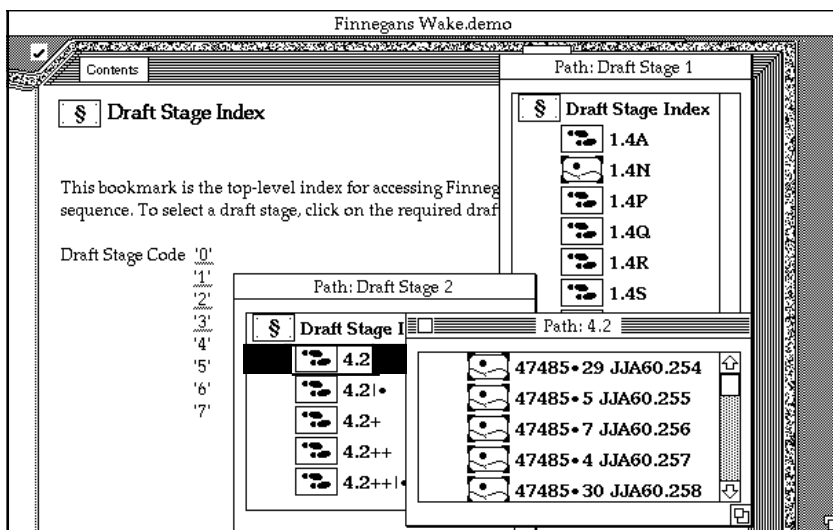


Figure 7 - access by draft stage is facilitated by a draft-stage index from where a reader can invoke a particular draft stage. In this example, the reader has invoked draft stages 1 and 2. Draft stage 1 has been subdivided by subsection although not all subsections are determined at this point in drafting the manuscript. Draft stage 2 is treated as a single story, albeit with amendments, i.e., extra draft material, extra draft material on the extra draft material, etc.

3.1. Navigation Pathways

In addition to the physical layout of the book, i.e., the sequence of bound pages, which was implemented as the default browsing sequence of bookmarks mirroring the table of contents, there are two orthogonal access paths defined: by draft stage; and by subsection. Access by draft stage (Figure 7) enables a reader to follow the development of book III through the major production milestones, (i.e., the draft stages¹²). Within each draft stage, the bookmarks are arranged in story (i.e., subsection) sequence. Access by subsection (Figure 8) enables a reader to follow the development of a particular subsection through all draft stages. Within a subsection, the bookmarks are arranged in draft-stage sequence.

¹¹ Discussions with some of the researchers at ITEM indicate that one outcome of their research is the identification of new, or alterations in existing, navigation structures.

¹² Each draft stage represents a re-write of book III from the previous draft stage and incorporating all changes since the previous draft stage.

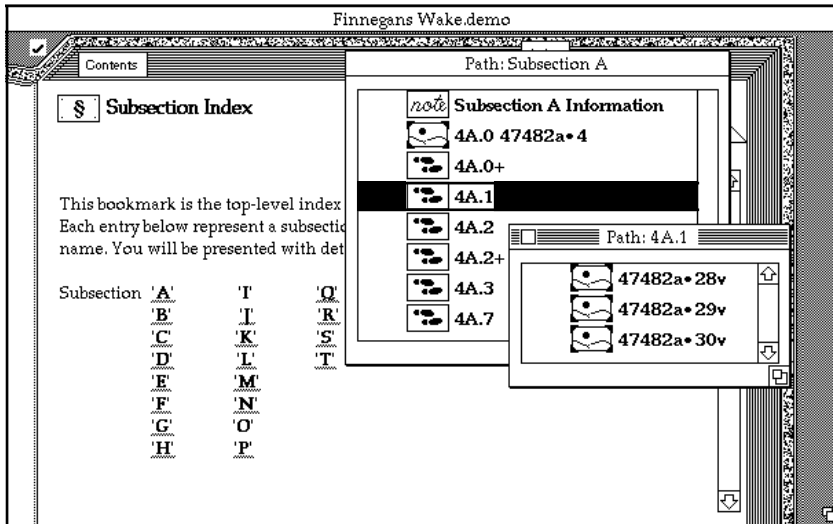


Figure 8 - access by subsection is facilitated by a subsection index from which the reader can select the subsection of interest. In this example, the reader has selected subsection A and in particular draft stage 1, consisting of three pages from the notebook.

3.2. Transcribed and Holographic Forms

To be useful to researchers, the electronic version had to be capable of representing both the *holographic form*¹³ of the notebook as well as the *transcribed form*¹⁴ (Figure 9). At present, the prototype only contains selected holographic images whilst the transcribed forms are being added later. For demonstration purposes, we have constructed partial transcribed form for some holographs.

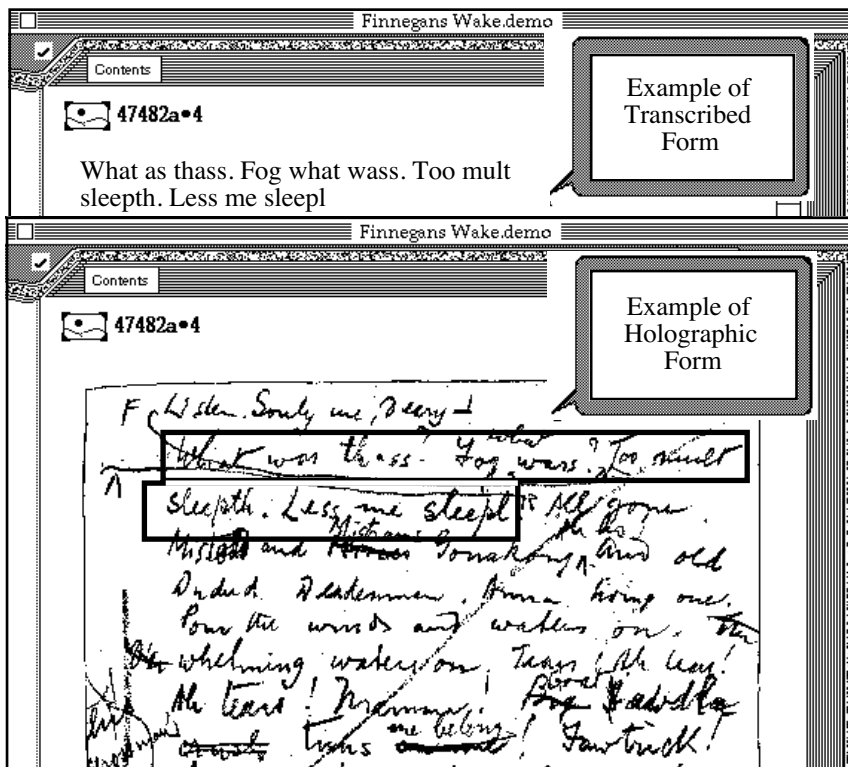


Figure 9 - holographic and transcribed forms are implemented using the IntelliText Views facility whereby each bookmark can have two views, the text view and the map view. The transcribed form is implemented using the text view whilst the holographic form uses the map view.

3.3. Re-use and Behaviour

¹³ The term *holographic form* refers to images of the actual pages as found in the notebook

¹⁴ The term *transcribed form* refers to the contents of a holograph in a computational form, i.e., word-processing form, marked-text form, etc.

WORKING DRAFT ONLY

As indicated earlier, Book III chapter 4 of *Finnegans Wake* was chosen as it is a good exemplar of the case where one single physical document records a number of interweaving textual stages. Analysis of the notebook indicates that this interweaving causes pages of the notebook to be re-used in different draft stages and/or subsections¹⁵. Thus the issues of re-use and behaviour are critical in representing the development of this manuscript. When navigating to a bookmark, the reader should be shown what aspect of the bookmark is relevant to their context: the reasons for navigating to the bookmark.

IntelliText enables the author and reader of a book to define many different behaviours for the bookmark and to invoke specific behaviours during navigation. This enables IntelliText to indicate, for example, what part of a holograph is relevant when following the path for subsection A draft stage 2, if the holograph is re-used for other subsection and/or draft stages (Figure 10).

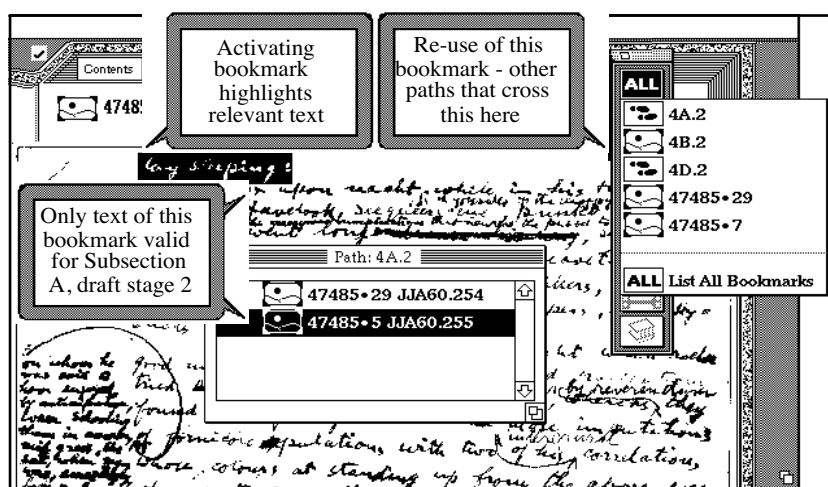


Figure 10 - navigating to a bookmark can invoke context-sensitive behaviours. In this case, following the path for *Subsection A, draft stage 2*, causes the relevant text on the holograph to be highlighted. Note that this bookmark is also used for *subsections B, and D in draft stage 2*, indicated by the *Lister* tool. Selecting the *List All* option causes the pop-up list to be copied into a temporary path-like window.

Bookmark behaviour can be specified two ways: on the bookmark itself, the default behaviour, so when the bookmark is activated this default behaviour is performed; and on a navigation link, when the author¹⁶ is presented with all behaviours the target bookmark can perform and selects the required behaviour for this navigation action. In the latter case, the navigation-related behaviour overrides the bookmark's default behaviour when the bookmark is actioned. Unless otherwise specified, behaviours are local to the target bookmark. Behaviours can be system wide, in which case they are available to all bookmarks. With the advent of Applescript™ and Hypercard 2.2, IntelliText now offers access to the complete Macintosh environment, so the range of possible behaviours is immense including invoking aspects of other scriptable applications (e.g., electronic mail, spreadsheets, drawing and painting programs, etc.).

3.4. Paths and Subpaths

¹⁵ Potentially, any piece of white space on any page can be used for any purpose, e.g., a name, an address, a shopping list, part of a subsection, etc. Research has to attribute each piece of text to its meaning or usage.

¹⁶Note, the term author in this context is used as a functional description describing any user creating, amending or deleting bookmarks or signposts. That user might be the book's author or any reader with appropriate privilege.

™ Macintosh, Applescript and Hypercard are trademarks of Apple Computers.

WORKING DRAFT ONLY

The complex interweaving nature and re-use of pages in the notebooks are almost impossible to model without use of an intervening concept, the *path*, to manage the linking structure¹⁷. The use of the path metaphor to guide navigation enabled each page of the notebook to be re-used on many paths without necessitating its duplication, as in the paper-based version of the Archive. The use of path-specific behaviours enabled the reader to be focussed on that part of the bookmark relevant to their context and/or to be presented with suitable visualisations of the bookmark's content.

The notion of path as implemented in IntelliText is akin to the path through a forest, it can cross other paths, has places of interest on the way, and are created, and thus should be followed, for specific reasons. As already indicated (Figure 10), a bookmark can be 'an interesting place' on many paths (i.e., many paths can cross a page) and a path can be implemented by use of sub-paths (Figure 8). The sub-path has analogies in the real world, one can build a road from Sydney to Adelaide by linking together, i.e., re-use, all the bits of road in between, the sub-paths. This enables the author of an IntelliText book to create many and varied structures to cater to differing requirements by re-using existing relevant structures. The use of sub-paths does however, add extra requirements on visualisation strategies (see below).

By adopting the path-in-a-forest metaphor, IntelliText enables an author to create a navigation network and not the more restrictive form, a navigation hierarchy or tree, although these limiting cases are available if required. Thus the network of paths allows for multiple disjoint entry points, doorways, into the electronic space, each entry point catering for different requirements.

4. Issues for Future Research

The prototype work to-date has examined the methods and technologies required to build an electronic manuscript. We have indicated, above, aspects of the IntelliText software that was used to trial our ideas and specific issues related to the use of Finnegans Wake. In this section we will describe some issues for future research. The issue that is present in all this future work is the meaning and use of context. The problem facing this research is that context is, by-and-large, an unknown quantity¹⁸.

4.1. Progress Indicator

Electronic editions suffer, in our estimation, by not supporting many of the cognitive stimuli that readers are used-to in reading paper-based manuscripts. This contributes to the lost-in-hyperspace syndrome associated with many non-trivial implementations of electronic manuscripts. One simple cognitive clue as to where we are is provided by the number of pages on the left and right of the book, i.e., are we about two thirds of the way through, or one quarter, etc. This stimulus can, and usually, is easily implemented using a progress indicator, like a slider on the scrolling parts of a window or a progress bar often observed when executing long functions.

In IntelliText the user can select the *Progress* tool which displays a progress bar in a *windoid*¹⁹. The issue here is not what is the best visualisation, although that is obviously very important, but what does the stimulus indicate in an environment where the reader's interactions with the physical objects are mediated by a logical level, mediation not present in paper-based books where no logical level is present and the reader interacts directly with the physical form, and a logical level enabling a reader to build their own 'books' by re-using some or all of the physical objects in any alternate

¹⁷ Without such management, indiscriminate linking will result eventually in every object being linked to every other object.

¹⁸ The author Jansen is currently producing a literature review on context, its meaning and usage. Draft copies are available by contacting him directly.

¹⁹ A *windoid* is a window always floating on the top of all other windows and hence always visible until closed.

WORKING DRAFT ONLY

sequence. IntelliText provides unskilled readers with front and back covers to minimise getting lost, but these objects define a logical book identifying its start and end, not a physical object nor its real boundaries. Cognitively, and spatially, what should the progress bar indicate, and what does this indication mean, when opening a *path* or *note* bookmark especially when the bookmark's behaviour is to display its text view like any other page in the book and not a more specialised visualisation? Should it indicate their physical location which may well be beyond the back cover? What does it actually mean to be 'beyond the back cover', given that there is no real analogy? Sliding-bar type of progress indicators work very well for list or sequence processing, but there remain many questions as to their semantics in multi-dimension volume or network processing.

4.2. Visualisation

One major lesson from this collaborative effort is the requirement for complex visualisation methodologies and techniques. Simply visualising the contents of a bookmark using a page-like display will satisfy naïve readers but will quickly frustrate expert readers. The use of navigation-based behaviours assists in addressing this problem but we believe more needs to be done.

The technique of visualising paths, and sub-paths, as a sequence in a window is highly unsatisfactory for long or complex paths. Our simple tests indicated that users will get easily lost in a nest of paths and sub-paths each appearing in their own seemingly unrelated window, and thus we require techniques for visualising paths not as just sequences, but as complex structures. Map metaphors may play a large part in addressing this issue.

Overall, the context facilities of IntelliText do help but only to a point and usually more so for an experienced IntelliText user. Naïve users will require time to become experts, not just in the domain of the book, but in the philosophy of IntelliText itself. Visual art techniques will also play an important role in addressing these issues²⁰.

4.3. Multi-user Support

The current philosophy in IntelliText is to mimic a real book. In this sense IntelliText is a single-user tool, just as each book can only be read by a single user at a time. Obviously, multi-reader paradigms based on client-server architectures are possible, and many tools offer this architecture. However, we feel there are still many problems facing reader support in these types of architectures, ownership being just one of them. IntelliText allows a reader to become an author by annotating their books, adding new information to existing information, creating new bookmarks, deleting bookmarks they have created, and changing the navigation structures to suit their requirements. Translating these activities into multi-book environments has resulted in formal protocols for library functions whereby, for example, defacing or destroying a books causes instant wrath from their custodian, the librarian. Similarly, other functions may have to be curtailed or modified in electronic multi-book environments. Context will play a major role in determining who will be allowed to do what and when.

4.4. Multi-book Support

In its philosophy of mimicking real books, the current version of IntelliText limits the navigation available to a reader to a single computer file. Links between books can only be currently accomplished if both the source and the target physically exist in the same computer file. Thus, if an author wants to provide complete access to another book, both must be physically in the same IntelliText file. Multi-book support, although already

²⁰ Visual artists continually represent complex structures, i.e., their reality or their view of reality, using two-dimensional visualisation techniques.

WORKING DRAFT ONLY

provided at some level by other systems, requires more investigation from the reader-support perspective and, like multi-user support, the development of formal protocols for defining what readers and authors can and can not do. Context becomes extremely important in this scenario by mapping available functions to required functions.

4.5. Behaviour not Limited to Those Known

We have described the behaviour functions available in IntelliText and their relation to context. The implementation of behaviours in IntelliText is similar to an object-oriented paradigm (Booch, 1986) where an object, a bookmark in IntelliText, can have *local* and *public* behaviours. In the object-oriented paradigm, a source object can only invoke a public behaviour of a target object. In other words, an object can only behave as it knows how to behave. However, it is conceivable to envisage an object taking the contents of another object and doing something with it inconceivable to the target object. In IntelliText terms, a bookmark, knowing how to speak text could take the contents of another bookmark and speak it without the target knowing how to speak. In a general sense, why should a required behaviour be limited to one of the behaviours of the target as long as the target is not altered or defaced? This is a more general approach than that practised by the object-oriented technology but one that implements more aspects of context.

5. Conclusions

The most prominent conclusion that can be drawn from this implementation is that this book would not have been implementable without the use of the *path* knowledge type. This structure facilitated the mapping of complex interleaving and object re-use into a collection of simpler structures able to be (easily?) visualised and followed.

Object re-use required a mechanism for highlighting relevant and appropriate aspects of the object depending on the current context. Navigation-based behaviour provided this mechanism, as well as supporting different visualisations of bookmarks, i.e., audio, etc.

Visualisation of complex structures is a limiting technology to-date. A book, in the traditional sense, supports many complex functions and visualisations that are not easily translated into the electronic medium. Some of the problems are related to the state of the hardware available to-date, whilst most, being cognitive based, require more research. A major problem to be addressed is the paradigm shift in moving into an electronic form. The book paradigm is partly translatable into the electronic medium but we need to be able to recognise when have we gone beyond its capabilities and what should augment it.

Acknowledgments

The work reported in this paper was made possible by a grant from the Recherche Associ e program of the Centre National de Recherche Scientifique (CNRS), in France. In addition, the input and interest shown by colleagues at the CNRS Institut des Textes et Manuscrits Modernes (ITEM) laboratory, especially Ann Marie Basset, is acknowledged and appreciated. Bob Colomb of the University of Queensland, Australia, and Anne Marie Vercoustre of the Institut National de Recherche en Informatique et Automatique, in France, have also contributed to the thoughts and ideas behind, and expressed, in this paper.

References

Booch G, *Object-Oriented Development*, IEEE Transactions on Software Engineering, Vol. SE-12, No. 2, February 1986, pp211-221

WORKING DRAFT ONLY

- Hayman D & Rose D, *James Joyce Finnegans Wake Book III, Chapter 4, A Facsimile of Drafts, Typescripts, & Proofs*, Garland Publishing Inc. New York, 1978
- Jansen B & Bray G, *Context & Knowledge Types Vs Serendipity*, in Proceedings of the 13th International Conference Avignon'93, France, 1993, pp85-95
- 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
- Jansen B, *Context: A Real Problem for Large and Shareable Knowledge Bases*, Proceedings of the International Conference on Building and Sharing of Very Large-Scale Knowledge Bases (KB&KS'93), Tokyo, Japan, 1993
- New Scientist, Technology section, 21 November 1992
- Slezak P, *Situated Cognition: Minds in Machines or Friendly Photocopiers?*, in Proceedings of the IJCAI'93 Workshop on "Using Knowledge in its Context", France 1993, pp111-120