Towards a Seamless Provision of Multimedia Course Material

Ian D. Benest, *University of York, UK*

SUMMARY

This paper describes the functionality and user-interface issues associated with the design of an automated system that provides a visually based online index to multimedia information. This generic software has been used to provide 'one-stop' access to a substantial corpus of teaching material required by students undertaking one module of an engineering degree. In particular the paper addresses the metaphorical access to and provision of teaching material, non-linear access to that material, user-interface issues that interfere with the learning process, and the multifarious nature of the material needed for such courses. It concludes that there is a need for a hypermedia system that presents an integrated view of the material with graceful navigational dynamics. It should enable the student to organize interrelationships between that material according to need, while retaining the 'published' structure.

INTRODUCTION

University teaching is still very much based on the modern equivalent of talk-and-chalk with the chalk being replaced by pre-prepared overhead slides that are also copied and distributed to the students prior to the lecture-theatre presentation. In many respects, university teaching has not properly embraced information technology in quite the way it might have been hoped.

Students (in a computer-orientated department) overwhelmingly (Barker and Benest, 1996) demand that lecture-based teaching be retained and not be replaced with computer-based methods; a surprising result when the popular media would have us believe that being taught by a computer is everyone's preferred method. Even distributing copies of all the slides is not preferred by about half the students; some preferring to be more active for at least some of the time during a long lecture by having to copy down notes from the projector screen. Producing the slides with a desktop publishing package and then providing those slides online for the students to print out at their expense hardly constitutes exploitation of information technology; it is merely a computerized photocopier. If information technology is truly to be exploited then it must offer more assistance in the learning process than paper can alone. And it must offer more than is provided by traditional lectures.

Embracing information technology

Providing electronic paper with a myriad of links ostensibly designed to mimic the associativeness of the human mind only complicates the user-interface; it leads to feelings of being lost, of being unsure that all material has been found, and of general untidiness that requires unnecessary effort to manage. Thus, the adoption of hypermedia does not automatically produce a better and easier-to-use product than a paper-based book.

If the learning process is to be enhanced, then information technology must be harnessed to provide a presentation that is superior to that available from paper-based materials. For example, a paper-based book describing how an electronic circuit provides its function, is a poor means of conveying that information. It is particularly poor when the textual description is over the page from the diagram and it refers to each transistor element that must be visually found in the diagram. Online, the user can see the

Innovations in Education and Training International
ISSN 1355-8005 print ISSN 1469-8420 online
http://www.tandf.co.uk/journals

diagram all the time, have each element identified (with an arrow for example) and an accompanying audio narrative removes the need to divert attention to the textual description. Unfortunately one of the problems of providing this enhanced presentation is the inordinate amount of time (hundreds, maybe thousands of hours) (Marshall *et al.*, 1994) needed to construct the material; time that cannot be justified in an Institution where there is an accepted, and imposed, research culture. Added to this is the inability of most lecturers to recognize poor user-interface design; the course developer often assumes that if the latest technology is used, it must result in a satisfactory interface – a similar policy to that adopted, it seems, by those in the computing industry.

Live lectures are not an ideal forum for learning, but they are well understood (and simple to appreciate) by lecturing staff. Live lectures provide a meeting place for daily social intercourse with others taking the same module. They provide a focus for interrogating the lecturer directly, in contrast with the remoteness of electronic mail. Lectures oil the learning process in preparation for private study where the bulk of the learning does take place. So if live lectures can be accurately mimicked and easily specified without the lecturer having to know how to design the user interface, then the lecture metaphor is a useful way of exploiting information technology.

Online lecture delivery

An online lecture (Benest, 1997) is the mimicry of a live lecture, with an audio narrative synchronized with such temporal events as: slide changes, revelation of information, animation (showing, for example, data-flow), the narrated construction of a diagram, and so on. Currently the lecture is both specified (Benest, 1994) and delivered through a hypermedia system known as the Book Emulator. The Book Emulator attempts to mimic accurately the essential features of a real book, while exploiting the possibilities that information technology can provide. An example of an online lecture delivered through the Book Emulator is given in Figure 1.

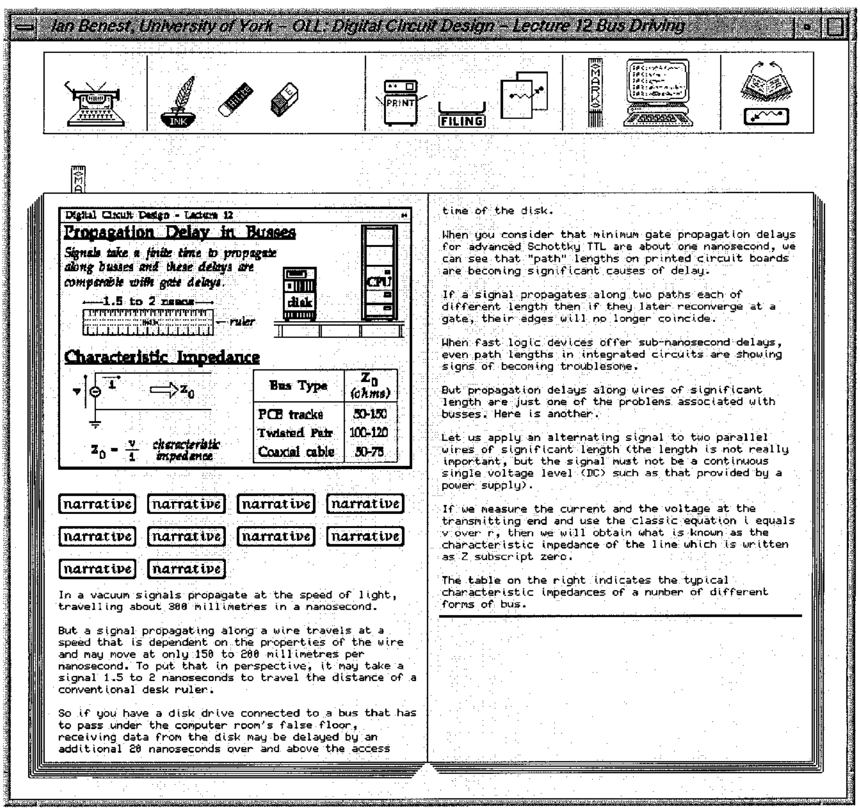

Figure 1 *An online lecture presented as a book of slides*

Each slide is located at the top of the left-hand page. Selecting the slide, *plays* the whole of that slide. Beneath the slide are narrative buttons which *play* the individual spoken fragments. Any text can be included; what is shown in Figure 1 is the script, allowing the student to speed-read through the narrative. The text is removed when the slide is *played* so as to encourage the focus of attention to be on the slide.

So if lectures are also available online, students can view them prior to the live event and view them again when some aspect of the lecture was missed or was found to be too difficult to absorb at the time.

Online lecture specification

The online lecture concept evolved from the need to provide a mechanism that enabled lecturers to specify content and the order in which it was to be presented, without them having to be concerned with the timing and consistency issues required of the user-interface. In other words the skills of the lecturer and the skills of the computer are exploited and complement each other. A key aspect though was the provision of a specification mechanism that was relatively straightforward to use. This was based on a schematic drawing facility illustrated in Figure 2.

On the right is the drawing (specification) book, and on the left is a library book of symbols. The user selects a symbol and it is made available in the drawing book attached to the mouse cursor. Pages can be turned over before the symbol is placed in the drawing book. The same drag-and-drop method is used to create online lectures including the incorporation of audio narratives. Animation can be specified by creating individual slides which, when run together, provide dynamic characteristics. Normally the synchronized speech is captured from a pre-recorded audio tape (recorded in quiet surroundings). It is attached to a link that is dragged-and-dropped into place like any other symbol. The narrative order is specified by the relative location of the audio links on the slide in a left-to-right, top-to-bottom order.

Lectures remain at the heart of engineering teaching. If information technology is to be embraced, then it should at least include an accurately mimicked implementation of those lectures before more-interactive learning support is added. In other words, the piecemeal incorporation of interactive learning support will only be attractive (without inducements) if they appear within a physical context that includes the bulk of the module, where the need to gain familiarity of the location of material is already rewarded. Furthermore, learning support systems can draw on (link to) the online lectures providing either extended remedial material or an introduction to what is to be learnt. That physical context would be satisfied by a 'one-stop' index to all the information and interactive facilities

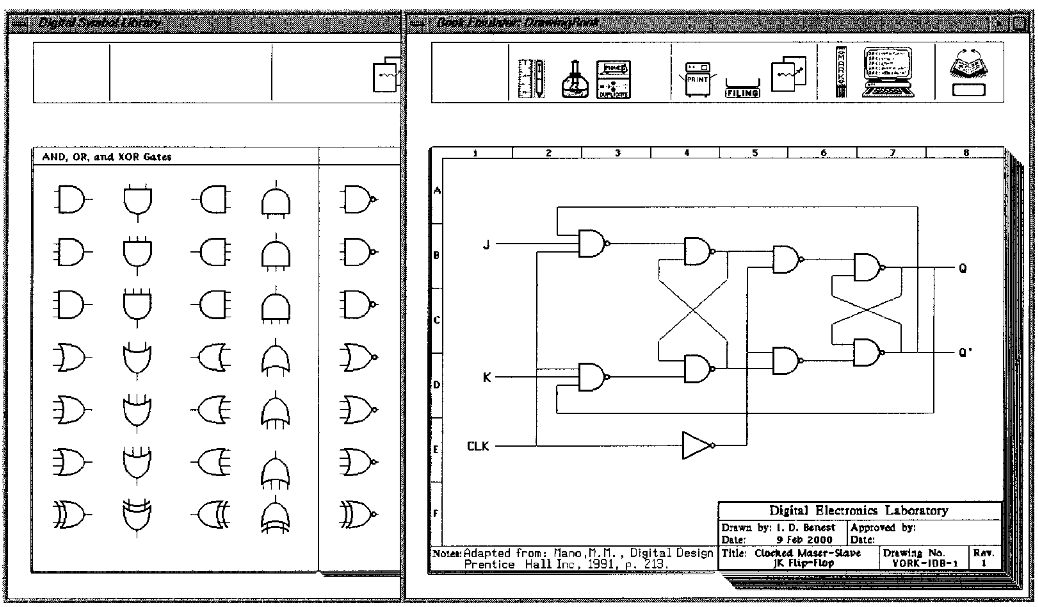

Figure 2 *A drawing book (right) with a symbol and font library book (on the left and behind)*

that are required to support a course module. This index should look similar (though not restricted) to that provided by the contents section in a paper-based book.

Rationale for an index

In addition to the online lectures engineering students commonly need: hard copies of notes, access to simulators and other interactive learning support material, manuals, device data sheets, and past examination papers. An index would provide an integrating view on to the electronic versions of those multifarious *documents*. It should provide a shallow hierarchy that reflects the structure and organization of the material. It should not exhibit a spaghetti-like structure that requires cognitive effort to maintain location (and which induces irritation when material is moved, and forgotten links become invalid).

Browser interfaces for an index

Popular browser technology provides two methods for information presentation and both offer difficulties for navigation and browsing that necessarily interfere with the learning process.

The first method is that of providing a continuous sheet of electronic paper, which is scrollable using a scroll-bar 'wedged' within a shaft restricting it to vertical movement. Positional cues for information can only be relative to a signpost such as a diagram. Remembering cues such as 'top of left-hand page' as is inherently available in a paper-based book, are not available. Scrolling the material amounts to moving the content of the document within the window. This makes visual search tiring because the eye has to track a variable rate of scroll and then fly back to the top or bottom of the window to catch the next block of information. A particular difficulty arises when the user is concentrating on the window content and the mouse 'falls out' of the shaft. Users must re-direct their attention to the mouse cursor, and having brought it back into the shaft and the mouse button pressed, the window content jumps in an unknown direction. This causes users to 'lose their place'. The scrolling handle reflects, counter-intuitively, the length of the electronic sheet; the larger the handle the shorter the sheet. The scrolling handle is necessarily small (if the content is long) and is therefore difficult (in the Fitts's Law sense (Murata, 1996)) to select. Furthermore, if the sheet is very long, one pixel movement will scroll more than a window-full of content, thus missing part of the information. The scroll bar is moved as if moving the window over the text, and yet it is the text that moves on the screen, not the containing window. Scrolling is endemic in modern window-based systems and unfortunately it is becoming an accepted *norm*; but it is a means of accessing and browsing material that does not synergize well with human performance.

The second common method of presentation is that of the paged display that provides both absolute and relative positional cues. But in order to go to the next page, the user has either to select a small target in the content of the document or make the long trek to the small 'back button'. This back button is located remotely from those buttons that take the user forward. Navigation requires that the eyes divert from the content to the buttons and then back again. While clicking is easy, pointing is not. This mechanism provides, overall, a high index of difficulty (Murata, 1996). Usually, the change of page takes the form of blanking the area and then drawing the page in front of the user. The constituent elements of the page often appear on the screen at random locations and cause the eye to move to the latest item. In effect the eye instinctively jumps around the screen until the update is complete. The repeated need to focus on navigation, the colour change flashing and the incessant eye movement all serve to interfere with the reading and skimming process. Furthermore, structuring information using this form of presentation leads to deep hierarchies within which the user can become lost.

An alternative method of presentation, encapsulated in the Book Emulator, is the imitation of a real book and its page turn. Both absolute and relative cues are naturally available. A page turn is animated such that the direction of travel is properly cued. (It is interesting to note that the page turn has to be animated in the opposite direction to that of travel in order for the book metaphor to hold.) Two of the three mouse buttons are used for going forwards and backwards in the electronic book; when these are continuously pressed the pages are flicked and the user is truly able to browse, much like a paper-based book. But the content on each page remains static during travel. The index of difficulty (Murata, 1996) is almost zero, amounting to the movement of a finger from one mouse button to the next. There is no movement of the mouse to select a tiny button on the screen. Of course the mouse cursor must remain within the (large) containing window. The third mouse button is used for selection, but its use for navigation is not nearly so important as it is in more conventional browsers.

THE INDEXER

The indexer is a piece of software that generates the index book. The index is specified by a plain text file where each line in the file refers to an item or set of items to appear in the index.

The specification file

Generally each line of text specifies an item to which a link is made in the index book; the exceptions include the various titles such as that of the index book itself, and table titles that are placed at the top of each page to provide structure. Generally each link specification includes a three-line title with each line separated by a semicolon. The exceptions to this are those links made to books (suitable for the Book Emulator) where the title and authorship are automatically extracted instead from the book's directory. Unlike the online lectures where all material needed (except that which is only available on the web) is included in the book's directory, the indexer only makes links and does not, in general, copy the material. An index exploits the same browser as that used for the online lectures and an example is shown in Figure 3.

The three-line title area is selectable and invokes the link. To the right of the title, is a small area that is completed by the indexing software identifying the type of document to which the link points. To the right of that is a speckled square box. When this box is selected, a question mark appears; when selected again a tick appears; and when selected again the speckled effect returns. It is up to the user how they interpret this

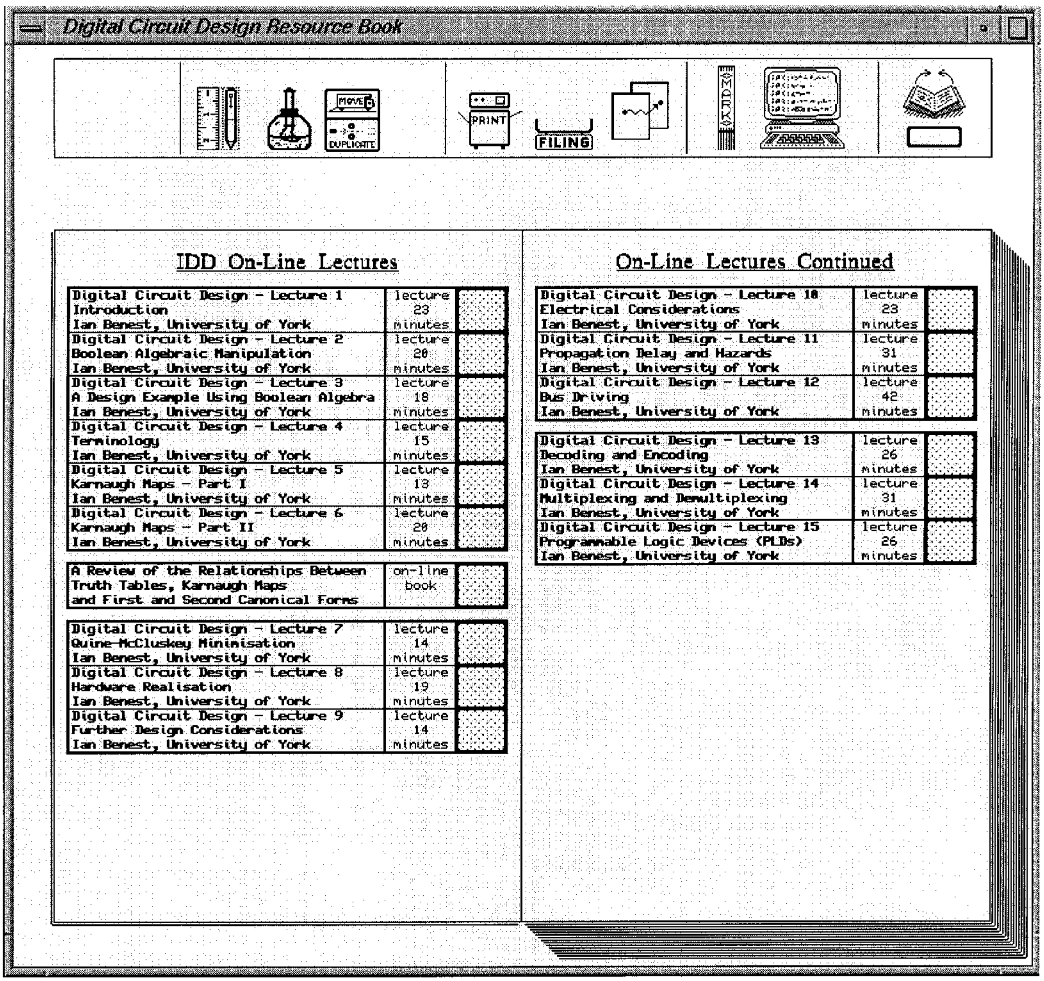

Figure 3 *An indexer book open at the links to the online lectures.*

box. A question mark might provide an indication that they need to return to that material. A tick might mean that all is understood. As well as providing structure, the titles at the top of the page are also used to group generated hyperlinks.

Incorporating online lectures

By specifying the directory name of the online lecture book, a link is inserted into the index book. When that link is selected in the index, the lecture book is started and placed on top; the lecture is automatically started and continues to the end or until the user stops it with a mouse button press. This mechanism encourages the user to concentrate on the lecture selected; there is not the requirement for the user to keep pointing-and-clicking. The approximate playing time for the lecture, generated when the online lecture is created, is provided to the left of the speckled box. When the users are finished with the lecture, they can close it down (i.e. quit) which brings the indexer to the front. Alternatively, a ‘back’ button is made available which instructs the index book to come to the front while leaving the lecture book available and behind. Further selection of that same lecture brings the book immediately to the front. In essence there are two levels of hierarchy, made possible by the provision of a broad menu that is available on many pages.

A single instruction in the specification file causes miniatures of all the slides used in the online lectures (in the order in which they appear) to be made available for selection. An example is shown in Figure 4. Slide text is not readable, but diagrams are easily recognized. Miniatures of supplementary pictures are also included; there are two such pictures shown in Figure 4.

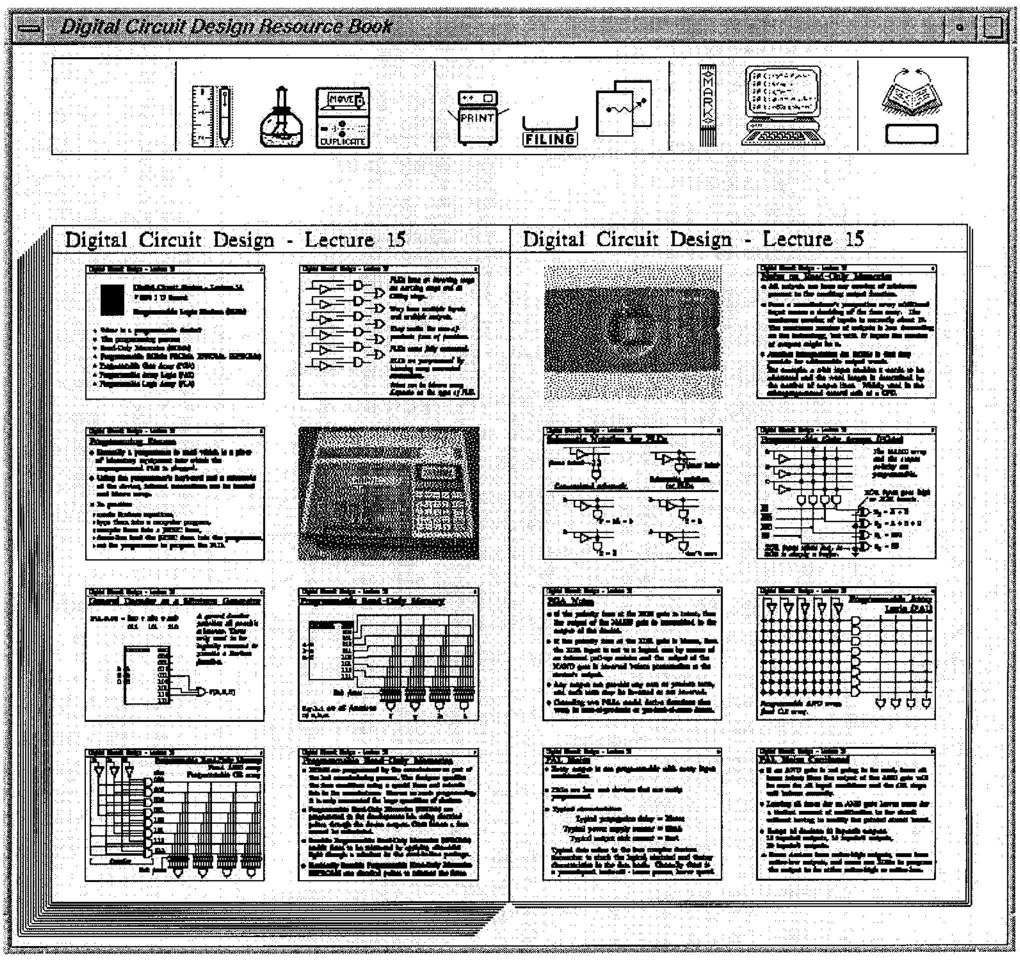

Figure 4 *Access to individual slides in the lectures*

These aid the visual recognition of the required lecture. Selecting a miniature invokes the online lecture that is automatically started at that slide; it stops after playing through that slide. The rest of the lecture may of course be browsed and other slides played.

Incorporating documents

Miscellaneous documents such as those in Postscript or HTML formats can be linked. The index generator automatically selects the correct viewer by inspecting the content of the file. These documents can be pictures, movies and audio files. Only one text processing file format is currently supported; that is 'troff'. All necessary pre-processors for troff are called prior to display. Troff files that have multiple sources are copied into one file, which is then stored in the index's directory. Similar strategies would be included for other text processing formats if required.

But the plethora of external viewers necessary to handle the numerous file formats leads to chaotic screens that have to be managed by the user; it is a common fault in hypermedia systems (e.g. De Roure and Hall, 1997; Yankelovich *et al*., 1988). Many of them have complex user-interfaces that are only trivially compatible with other viewers. They usually take an inordinate amount of time both to spawn in the first place and to get them to display page 317 (for example) in a user-manual or data book. This time delay causes a breakdown in the manipulative dialogue that leads to a temporary memory loss of the reason for the selection. Some formats are secret and so cannot be incorporated in a general search strategy. Some formats are continually being enhanced which leads to downward incompatibilities. Some will only display diagrams properly if the page has been magnified (interconnecting lines in circuit diagrams are often missing, for example). Exploiting external viewers is the antithesis of seamless provision.

Thus, a bitmap viewer has been created which displays bitmap images of documents; navigation in this viewer is similar to that of the Book Emulator. It displays the document almost immediately and can be remotely controlled to switch documents so that only one viewer (rather than multiple Postscript viewers) is running at a time.

Printing provision

As with viewing miscellaneous documents, a link can be provided for the documents to be printed. HTML documents require a World-Wide Web browser to be run first, and a message is sent to the browser for that document to be printed. If the browser was already running with another document then that document is reloaded after the print instruction has been queued. The whole process is slow. When online lectures are generated a Postscript booklet is automatically created in a file with a particular name. A single indexer instruction exists that causes the software to look for all book instructions in the index specification file. If a book contains that particularly named Postscript file then a printing link is inserted in the index book. A document will only be sent to the printer once per session; a tick is placed in the speckled box to indicate that the print request has been sent. The tick is persistent (though the user can change it).

Picture, audio and movie libraries

The indexer can build separate library books for pictures, movies and music, and the presentation is similar to that shown in Figure 4. Of course pages of pictures can also be included with the other links to module material.

When a link is to be provided to a picture or to a movie in an index book, a monochrome miniature of the picture or the first frame of the movie is automatically created and resized. Each miniature has a fixed aspect ratio that ignores the aspect ratio of the original. This allows for a tidy appearance in the index book while still enabling the picture to be recognized. If an attendant audio file exists then when the user selects the picture, that audio file is played; a short delay is inserted to account for the slow display of colour images. The audio would usually be a short narrative explaining the picture, but it could be music from an audio CD with the recognizable cover design as the picture.

Since an index book could contain links to many hundreds of pictures, there is a possibility that the screen could become visually chaotic if very many pictures were selected and moved about the screen by the user. The user would need to perform a good deal of screen housekeeping. This would amount to selecting a tiny square in the top corner of the picture, letting the menu drop down, and then selecting the quit button for every picture on the screen. This is very tedious, so picture files are allowed to remain displayed for about ninety seconds, after which they are automatically removed from the screen.

The Book Emulator supports the use of audio file indexes where the audio is speech. The indexes consist of file offsets for both the start and end of each sentence in the file. This enables (for example) the audio to be paused with the space bar and then when pressed again to release the pause, the audio is 'wound' back to the beginning of the current sentence. The audio index is created automatically when an online lecture is generated.

Bibliographic access

Bibliographic indexes may be created providing access to many volumes of many journals. Again, such links can also be provided within a module index. These links cause a special program to be run that takes references from a journal volume, contained in a single file in 'refer' format, and makes up a book containing a series of front (or back) pages, one for each issue. An example is shown in Figure 5.

Each 'front page' contains the title and list of authors contained in that issue. Each title is made selectable causing the abstract to be displayed on the opposite page. At the same time, the full reference is placed in an editor's paste buffer and in the Book Emulator's own paste buffer; a copyright notice is included. Behind the front (back) pages, the abstracts of each reference are included, one per page. When there is room, the user can make a short note underneath the abstract.

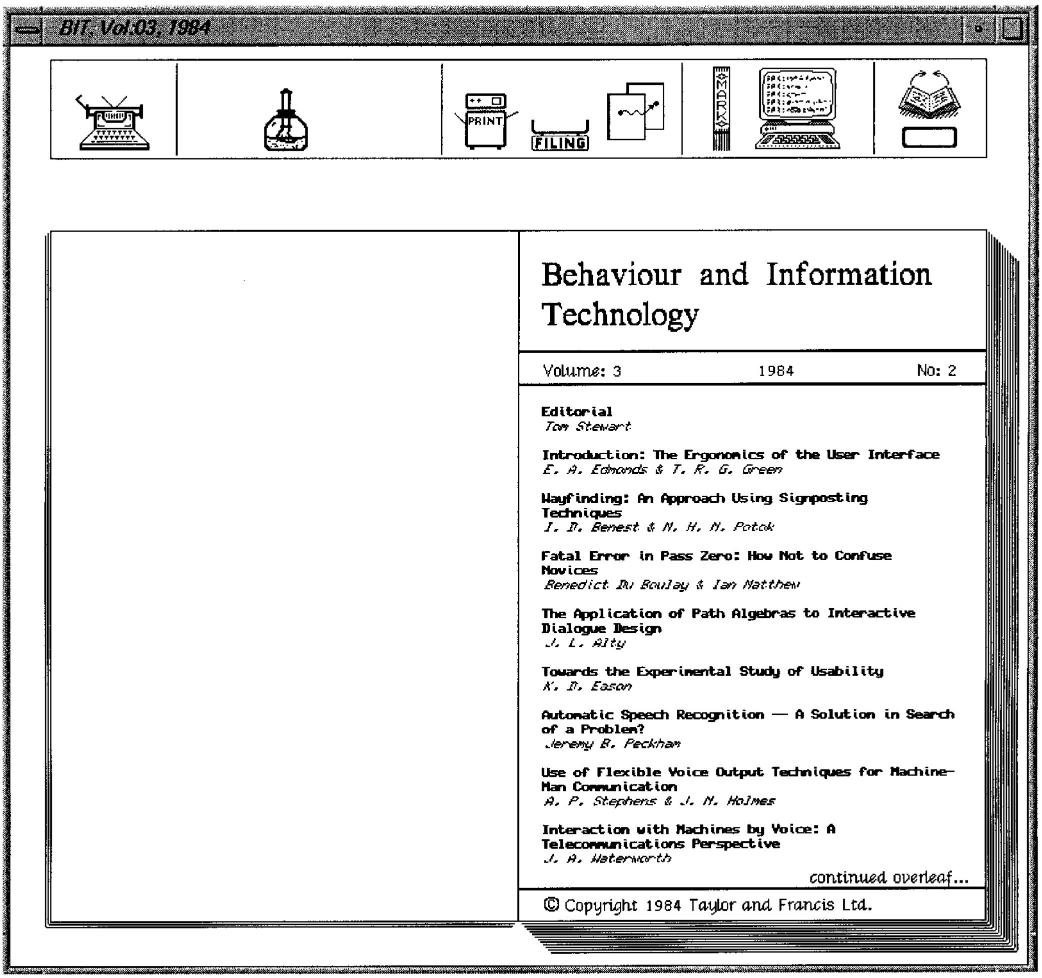

Figure 5 *A book of 'front covers' for a journal volume. These references were taken from Perlman's HCI Bibliography (Perlman, 1991)*

Access to learning support tools

Any runnable program or executable script can be selectable in an index book. Usually such programs would control their own windows and might access a remote server on which the main program was executed. Thus, links to any intelligent tutoring system or simulator could be incorporated in an index book.

Installation

Once the lecturer is satisfied with the online lectures, each lecture book directory is placed in the public area in the filestore. A book consists of a public set of files contained in a directory. A related directory is automatically created in the user's filestore when the Book Emulator is first run by that user with that book. This private directory contains such information as bookmarks, annotations, schematic drawings, etc created by that user in that book. The index book links are specified with absolute addresses to each document. Once the index book is generated it is also placed in the public area.

EXPLOITING DYNAMIC HYPERLINKS

Dynamic hyperlinks in the Book Emulator

The indexer provides areas on each page that, when selected with the mouse, cause information to be presented either visually or aurally through the most appropriate browser. These hot-spots thus provide 'static hyperlinks'. The Book Emulator also has the ability to support 'dynamic hyperlinks'. The anchors for these dynamic hyperlinks are graphical symbols that can be copied any number of times and can be dragged-and-dropped anywhere within the current book (turning the pages if necessary) and dragged-and-dropped into any other book. When (long) selected, the information to which the hyperlink points is presented using whatever viewing software is most appropriate.

When the index book is generated, the static hyperlinks (except those for hard copies) are programmed to generate appropriate dynamic hyperlinks that are automatically placed in a buffer, which is available to all other books. To retrieve the dynamic hyperlink, the user must select the glue-pot (paste-pot) symbol located above the book within the Book Emulator's containing window. This causes the hyperlink symbol to be attached to the mouse cursor and it can be placed where the user wishes within a book. The link is retained in the glue-pot so that when the glue-pot is selected again another copy of the link is attached to the mouse cursor. Note that there is no tedious dialogue to form the link as was required in Intermedia (Yankelovich *et al.*, 1988). 'Form filling is never a convenient option' (Carr *et al.*, 1998). If the link is actually placed in a book and the information becomes inaccessible, the symbol remains with a skull and cross bones on top. If the information has changed since it was last viewed or heard by that user, the symbol is greyed. Dynamic hyperlinks link to any information, however it is displayed. A feature of the dynamic hyperlinking mechanism is the ability for them to be logically grouped with other dynamic hyperlinks; it is up to users if they wish to group dynamic hyperlinks visually.

Support for the visual search of dynamic hyperlinks

The dynamic hyperlinks can be put in any book accessible through the Book Emulator; in particular a notebook or a book designed to help the user personally categorize the links. An example of the latter is shown in Figure 6. This supports the important notion of spatial hypertext (Marshall and Shipman, 1995) that enables users to form associations between disparate document parts. The Book Emulator allows the user to connect lines between links or visually segregate links into groups in order to assist the user in remembering relationships. Different pages can provide further segregation.

If the symbol were the same for all dynamic hyperlinks, then finding the one that was required would cause the user to point-and-click many times in order to find the one required. This form of tedious interaction, which is often misnamed as browsing, is commonly required when navigating the World Wide Web.

So the anchor symbols vary; where the static links were pictures (of online lecture slides, pictures, movies), the same symbols used for those static links are also used for the dynamic link symbols. This is designed to help the user remember the original selection. When such a symbol is attached to the mouse cursor, it is possible to cycle through (in either direction) the other symbols in the same logical group. This means that if the user has kept a link to slide three of lecture five, then any link to any other slide in lecture five can be obtained by cycling through; when selected the lecture book is opened at that new slide. For picture and movie library index books, the page titling instruction determines the grouping of these dynamic links. Thus if Matisse and Rembrandt are separately headed, the dynamic hyperlink groups are also separate. Then only those

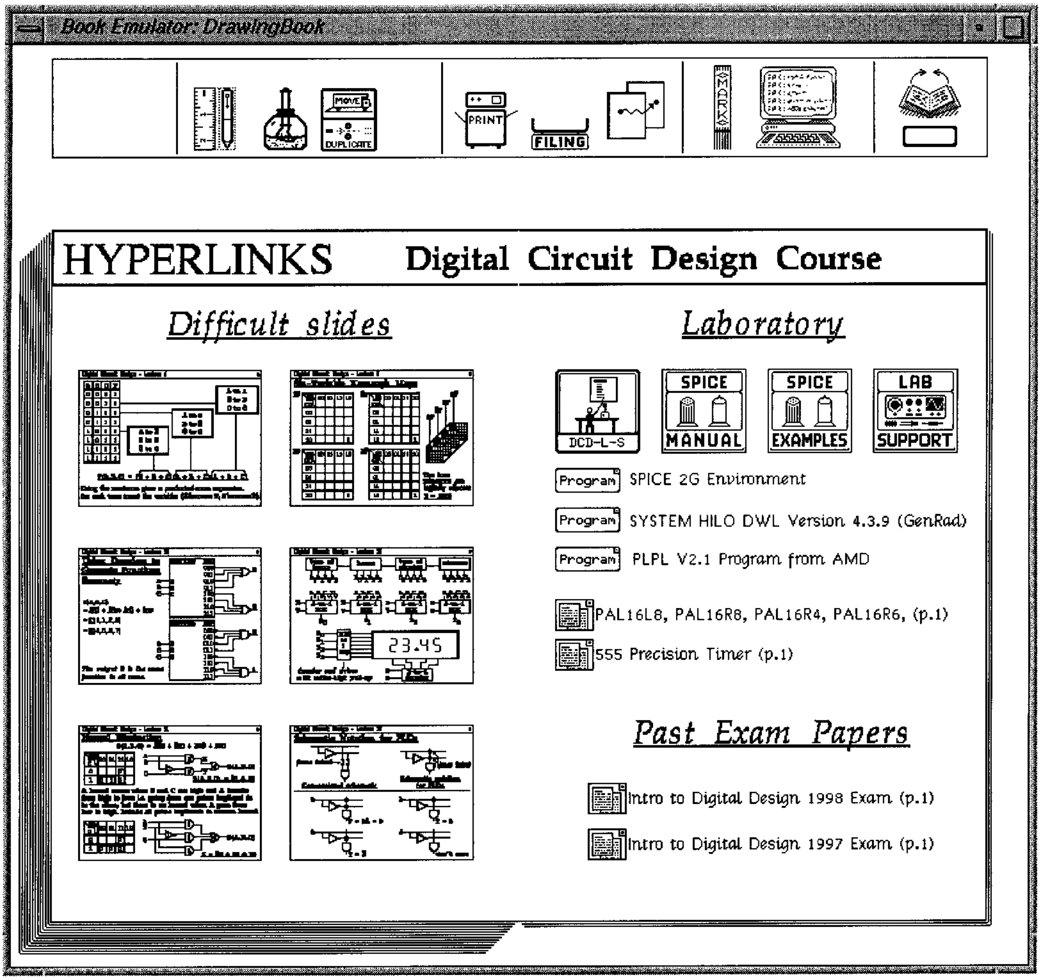

Figure 6 *A book (actually the DrawingBook whose pages can be changed to reflect their purpose) for holding dynamic hyperlinks*

pictures with Matisse can be cycled through on that dynamic hyperlink.

Where the static link is not a picture, a generic symbol is automatically used which still distinguishes between pictures, movies, audio files and documents. Since these are also logically grouped, the symbol may be changed, this time to reflect more accurately the content of the information. For example a GIF file may contain a graph, and the user can change the automatically selected generic picture symbol to one of the graph symbols. These generic symbols (for example the exam paper links in Figure 6) are supplemented with the title of the document so that the visual search is confirmed by reading the title. The initial title is the top line given in the static hyperlink in the index. The title may be edited by the user.

The Book Emulator itself also provides the ability to create a link to the current position in a book; it is placed in the glue-pot from where it may be extracted any number of times into any number of books. It is also placed in the filing tray where it is added to a stack of links from where it may be extracted into other books at a later date. This link creation is interactively simpler (fewer steps) than that adopted by other hypermedia systems (such as Yankelovich *et al*., 1988) and probably more human-orientated; the users may gather up links as they travel, and only later do they file them away. Each book creates a standard visual anchor determined by the author of the book, together with a variety of others (with different information in the symbol or different size and shape); these are grouped and may be cycled through at the time of placement. This variety is designed to aid visual search while flicking through the book; the symbols (which are differently sized) can be placed in margins, at the end of paragraphs, and so on.

SEARCHING

When the index book is automatically constructed, search instructions are installed so that all documents (except those at remote sites accessed via the web) can be searched. This includes image, movie and music descriptive annotations, the bibliography links, the lecture slides and transcripts of each speech file. During a search, the static hyperlinks are highlighted if there is at least one hit in a document. For online lectures, the miniature slides are highlighted if a hit in the slide or in the accompanying narrative occurs. Symbols selected from a symbol library book (such as that shown in Figure 2) may be inserted in the search buffer causing those slides containing that symbol to be highlighted. If dynamic hyperlinks have been placed in the index book, then the files to which they point are also searched and the anchors are highlighted if a hit is found.

The search results are displayed in the index book by highlighting. The pages are automatically flicked through to show the user. A key feature is that the documents found are not separately listed but included in context, making it easier for the user to decide which is probably the most appropriate document to view; is it a data sheet application, or a lecture, or the title given by the user to a simulation exercise, or a note made by the user close to a bibliographic abstract? Users are thus able to utilize their growing familiarity with the structure of the corpus shown in the index book; a feature of seamless integration.

But those files whose formats are secret (e.g. PowerPoint) cannot be incorporated in a 'search-everything' facility; the user is restricted to specific viewers by which those documents can be searched. For example, instead of being able to search the entire corpus of material available, the student would have to point, click, wait, find the search button, execute the search, wait, go back, and then repeat for every document that is stored in a secret format. And how is the student to know which documents are secret in a system that can globally search some documents, but not all? Thus, secretly formatted documents cannot be seamlessly incorporated.

CONCLUSIONS

The benefits of seamless behaviour in an information base are manipulative consistency where attention to one user-interface should lead to less complexity (i.e. less unnecessary interaction, less visual chaos, and less demand on human memory), and more fluid performance. This enables the user to concentrate on learning rather than fiddling and having to remember idiosyncrasies of the system as a whole. Seamless behaviour exudes 'one system'; no jumping screens, no flashes of colour change. It should capitalize on the users' cognitive needs rather than jarring their perceptual processors.

But people seem to assume that interfaces just have to be unpleasant and, given the information that is available, they simply accept it. They marvel at what can be displayed. Designers of information systems place great reliance on the humans' ability to adapt and accept. This designer arrogance would be unacceptable in media designed to entertain such as a television documentary. Would viewers accept a seven-second scene change in a documentary with a blank screen while they wait? Scene changes are not abrupt; they are almost imperceptibly faded so that there is never, for example, a black to light-grey to deep-blue colour change sequence. Where science fiction seriously depicts access to information, it is always fluid and information is found almost immediately.

Educational information systems are some way from being seamless, but it is a goal to which research should strive. The author believes that the basic file format required should be a bitmap ensuring that the publisher's desired appearance for the document is maintained. Ideally the bitmap should store anti-aliased lines and text for high legibility, and be stored with loss-less compression. Given that teaching material can be enhanced with temporal and sequential information, the basic file format should be compatible with a recognized movie format. Then for long documents, page turns could be animated by the publisher rather than the browser.

The format needs to be able to hold the text contained in the bitmap and include a description of the layout with any incorporated diagrams and pictures. The format should be capable of storing audio narration, which should be supplemented with its transcript keyed in with the video sequence. Multiple audio streams, to support multiple languages each with their own transcript, are desirable. A semantic-based index into the video and audio should be possible, and the viewer should display the file almost instantly at whatever location it was signalled to start. Above all, the users' needs rather than technical capabilities should determine the file format and the access to the interactive functions. For example, showing in a bar chart where music ends and speech begins may be useful to a documentary's producer, but it is not particularly useful to a student trying to understand the underlying content of the documentary. MPEG-7 (Nack and Lindsay, 1999a; b) and MHEG-5 (Echiffre *et al.*, 1998) may go some way to providing the basis for this required format. But the traditional encapsulation in one large marked-up file with links is not a necessary precondition for the new file format.

Educational institutions of course should not rely on the World Wide Web to provide information since it would have no control over the availability of external information; so the material would need to be pre-copied and stored locally and perhaps issued to students on a computer disk. After copying the information, it has to be installed, so conversion to the new format might be necessary.

Search facilities should evolve to become unintrusive guidance mechanisms based less on keyword searching and more on information semantics, with the retention of the best aspects of keyword usage. This is because as the online material becomes greater, so the student needs to know: 'is information on X available in this system?', 'how can information X be found?', has information X already been seen?', 'what is the viewed proportion of the module so far?', and so on.

Perhaps the most difficult item to incorporate seamlessly is the more interactive learning support (in contrast with interactive teaching support) application and the key is to develop such material based on the new file format. It is, of course, inevitable that useful legacy systems will remain and they will have to be simply linked at the expense of bringing possible visual chaos to the screen.

ACKNOWLEDGEMENTS

The author is grateful to Michael Bennett for his automatic audio file indexing software and to Philip Barker for his thought-provoking discussions.

REFERENCES

Barker, P G and Benest, I D (1996) The on-line lecture concept – a comparison of two approaches. In *Learning at a Distance: Developments in Media Technologies*, IEE Colloquium Digest No. 96/148, IEE, London, 9/1–9/7.

Benest, I D (1994) A schematic entry drawing capability in a linearised hypermedia system, *Computer Graphics Forum*, **13**, 5, 293–303.

Benest, I D (1997) The specification and presentation of on-line lectures, *Innovations in Education and Training International*, **34**, 1, 32–43.

Carr, L A, Hall, W and Hitchcock, S (1998) Link services on link agents?, *Hypertext 98*, Pittsburgh, USA, 113–22.

De Roure, D and Hall, W (1997) Distributed multimedia information systems, *IEEE Multimedia*, **4**, 4, 68–73.

Echiffre, M, Marchisio, C, Marchisio, P, Panicciari, P and Del Rossi, S (1998) MHEG-5 aims, concepts, and implementation issues, *IEEE Multimedia*, **5**, 1, 84–91.

Marshall, C C and Shipman, F M (1995) Spatial hypertext: designing for change, *Communications of the ACM*, **38**, 8, 88–97.

Marshall, I M, Samson, W B and Dugard, P I (1994) Multimedia courseware cost modelling, *Proceedings of the European Cost Modelling Conference 94*, Italy, 27.1–27.19.

Murata, A (1996) Empirical evaluation of performance models of pointing accuracy and speed with a PC mouse, *International Journal of Human Computer Interaction*, **8**, 4, 457–69.

Nack, F and Lindsay, A T (1999a) Everything you wanted to know about MPEG-7: Part 1, *IEEE Multimedia*, **6**, 3, 65–77.

Nack, F and Lindsay, A T (1999b) Everything you wanted to know about MPEG-7: Part 2, *IEEE Multimedia*, **6**, 4, 64–73.

Perlman, G (1991) The HCI bibliography project, *ACM SIGCHI Bulletin*, **23**, 3, 15–20.

Yankelovich, N, Haan, B J, Meyrowitz, N K and Drucker, S M (1988) Intermedia: the concept and the construction of a seamless information environment, *IEEE Computer*, **21**, 1, 81–96.

BIOGRAPHICAL NOTE

Ian Benest is a member of the academic staff at the University of York in the UK. He lectures on Digital Circuit Design and User Interface Design.

Address for correspondence: Ian Benest, Department of Computer Science, University of York, Heslington, York, YO10 5DD, UK.
Tel: (+44) (0)1904 432736;
Fax: (+44) (0)1904 432767;
e-mail: ian.benest@cs.york.ac.uk