Reading and Writing the Electronic Book

Nicole Yankelovich and Norman Meyrowitz
Institute for Research in Information and Scholarship, Brown University

Andries van Dam, Department of Computer Science, Brown University

The differences between the print and electronic media lead to the definition of a set of capabilities necessary for multimedia electronic document systems.

Juxtaposing "electronic books" and hard-copy books creates a useful framework for critically examining the computer's role in document preparation and presentation. Although electronic document systems are not directly comparable to paper books, the two nevertheless serve a number of common purposes. Both are used as sources of information, as learning devices, and as mechanisms for communication between people who are distant in time or place.

By considering the strengths and weaknesses of paper and electronic documents, it is possible to formulate a set of capabilities that electronic document systems should possess to maximize the advantages of the electronic medium and overcome some of the disadvantages inherent in the print medium. After outlining these general capabilities, the article describes four document systems developed or under development at Brown University that illustrate many of the different necessary functions.

Print medium

Scholars, or "knowledge workers," rely heavily on print media, even though electronic creation and dissemination of information is possible with today's technology. In some cases, this reliance on print is part of a long, ingrained tradition, but in other cases, print is still simply the most appropriate vehicle, either because electronic document systems are impractical to use or because they do not meet a particular objective as well as does paper.

The most important fundamental property of books is that they are static. Once printed, a book cannot be altered except by reprinting, and at no time do readers have the opportunity to change or manipulate its contents. The static nature of books is both their biggest asset and their most serious shortcoming.

A review of the assets and shortcomings of books is helpful in establishing a list of capabilities essential for high-quality electronic document systems (see Table 1 on the next page).

Electronic medium

Electronic document systems have their own advantages and limitations. In some cases they are more powerful or appropriate than paper books for meeting the range of information needs of scholars within the university community. In other cases, books are more useful.

0018-9162/85/1000-0015$01.00 © 1985 IEEE

Table 1. Print medium: advantages and disadvantages.

CHARACTERISTIC	ADVANTAGES	DISADVANTAGES
Integrity of information	Historical value Never inaccessible because of unreliable hardware	Readers can never alter content Readers cannot customize information Cannot conform to user preferences (e.g., type size, margin width)
Physical entities	Portable Allows browsing and exploring Allows annotation and underlining Aesthetically appealing	Limited to 2-D information Limited to static text and graphics Costly to reproduce for quickly outdated information Often hard to locate specific information
Static	[illegible]	Cannot handle sound or motion Difficult to create multiple indices
Advanced technology	Well-defined and accepted standards Typography, graphic design, and photo reproduction refined fields High-resolution print and graphics Easy to read	Joint authorship [illegible] Rekeying text is error-prone

Advantages. Theodore Nelson and Douglas Engelbart were among the first to articulate the benefits of electronic document systems.[1,2] In the early 1960's, they recognized that computers were well suited to helping scholars and others create *connectivity*—webs of related information. In the print medium scholars often mark up books, articles, and papers. When a phrase or illustration sparks a connection to an idea in another book in a scholar's mind, he or she writes that connection, or "link," down *next to* the phrase or picture that sparked the thought. Providing footnotes, references, and word glosses in books is an author's way of making annotations or explicitly indicating connections between his or her writing and other documents, schools of thought, and definitions. These standard devices provide readers with pointers to additional reading and information sources that will enrich the understanding of the subject matter of books so annotated. Creating webs of information and adding to them are integral to all scholarly work, and in the domain of scholarship these webs are commonly called *literatures*. Nelson defines "a literature" (as in "the scientific literature" or a field of discourse) "as a system of interconnected writings."[3]

George Landow of Brown's English Department sees connections and the act of following links as crucial to education. In his teaching, he is particularly interested in helping students see links between the literature they read and such things as art, politics, philosophical thought, and religious doctrine.

Colleagues can easily view one another's documents (if given permission), send and receive personal electronic messages, and jointly edit the same document without leaving their own workplace.

Nelson and Landow both stress the importance of *observing existing connections*, which can be done by studying "the literature," and *making new connections*. Electronic document systems help scholars both create connections and follow those made by others. Because electronic books allow flexible organization of material, they provide authors and readers with a greater degree of freedom than printed books. Explicit connections—"links"—allowing readers to travel from document to document (as one does with an encyclopedia) or from one place in a document to another in the same document can be made effortlessly by authors, thus fostering the creation of *information webs*. With the electronic medium, readers are not obliged to search through library stacks to look up referenced books and articles; they can quickly follow trails of footnotes without losing their original context.

Linking scholars together—intercommunicability—is an essential aspect of connectivity. Electronic document systems running either on a multiuser, time-shared system or on a series of networked workstations allow authors and readers to communicate with one another in a number of ways. Colleagues can easily view one another's documents (if given permission), send and receive personal electronic messages, and jointly edit the same document without leaving their own workplace. These types of communication capabilities may foster "on-line communities"[2,4] of researchers or students and enhance the ability of scholars to make meaningful connections.

Perhaps the greatest advantage of electronic documents over paper ones is their ability to handle many more graphic elements. By combining a variety of media, electronic books can provide not only static images, but also dynamics (e.g., computer animations and computer-controlled video sequences), interactivity (e.g., ability to move objects, change and edit objects, and change states), and sound (e.g., computer-generated or audio disk recordings). These features all help in creating better *audiovisualization*. For example, a biology student might be able to rotate and slice a three-dimensional model of a plant cell while reading related material, or a theater arts professor might have students examine video recordings of theatrical productions in parallel with blocking diagrams.

Electronic document systems can be useful tools for visualizing the structure of the information web as well as visualizing the individual concepts or processes contained in that structure. Not only can they allow scholars to make and follow links, but they can also provide a diagrammatic overview of the web of connections. Different "maps" of information webs can be generated to illustrate the connections that exist in a body of material. Some types of maps, however, are easier to generate than others. For example, it is possible to create a map of a reader's path through a document corpus or a diagram of all possible links from the reader's current position, but as the number of connections and quantity of information increases, so does the difficulty of generating maps of the entire information web. Since most readers cannot readily understand a diagram with hundreds of crisscrossing interconnections, the problem of distilling or summarizing the information must be addressed. In addition, authors may often make circular references, causing even more complexities in graphically representing the web of connections.

The paper medium does not allow a reader to alter the contents of a book. Electronic documents, however, are dynamic in the sense that both authors and readers can customize the material contained within a corpus of documents. For example, in a military setting, an author may want to provide complete access to certain information to those with the appropriate security clearance and only partial access to all others. If working with paper, this author would be forced to publish separate books for each constituency. Readers also may want to "filter" —limit access to—the information. A literature student, for example, might want to look at only critical commentaries on Shakespeare's plays, while an acting student might be interested in solely the original texts. Filtering permits readers to select only information they consider pertinent. A set of selection criteria or *attributes*, somewhat similar to index terms in the back of a book, can define relationships between information blocks or can identify structural components. For instance, a lawyer might want to apply a filter that would display all cases mentioning the name "J. Smith," to examine all cases that "support" a particular decision, or to view the first sentence of every major decision in a given area.

A set of selection criteria or attributes, somewhat similar to index terms in the back of a book, can define relationships between information blocks or can identify structural components.

Nicholas Negroponte's idea of "idiosyncratic systems"[5] goes a step beyond filtering. He suggests that electronic documents adaptively display not only *what* material readers wish to see, but also *how* they would like it presented. For example, one reader might indicate a preference for as much graphic information as possible, while another might like the same information but with a more even mix of graphics and text. Negroponte's notion of an idiosyncratic document system can be generalized by creating a knowledge base containing profiles of both the readers and the information within the system. Rather than being powerful multimedia repositories for information, such systems would become "assistants" by not just presenting information, but also by carrying on a dialog with the reader. Although research on "automated authoring" is underway at Brown, MIT, and elsewhere, future knowledge-based electronic document systems are beyond the scope of this article, which focuses on present and near-term issues.

As with hard copy manuscripts, it is possible to preserve multiple versions of a single electronic document. Authors can save a document at any point in the creation process and can produce hard copy renderings of the same information (at least all text and static graphics).

Finally, the electronic medium can aid dramatically in the updating and dissemination of information. In many cases, editing an electronic document (using interactive editors for text, graphics, music, etc.) is far more efficient than making changes to a printed book. In addition, the cost of dissemination (in terms of both money and of natural resources) one day may be greatly reduced by the advent of national networks and high-density storage devices (e.g., diskettes, video disks, CD ROMs, etc).

Disadvantages. A major shortcoming in most electronic document systems developed using current technology is their failure to provide adequate

098

information about where readers are in a document. Readers of paper books can always tell if they are "at the end of the book" or "three-quarters through it." If electronic books were merely linear sequences on a computer screen, then a two-dimensional gauge or a simple numbering scheme would suffice. However, because they are nonsequentially organized and the "middle" for one reader might be the "end" for another; a reader can follow link after link and feel disoriented.

Although it is possible to store documents in various stages of revision, the electronic medium does not encourage one to do so. With current text-editing systems, authors must have a sense of history to consciously save "old" versions of documents created with interactive editors. Even when the versions have been saved, it is difficult to see where changes have been made in a document that always looks "clean" no matter how drastically it has been revised. Authors working in teams also find it difficult to notice changes made by coauthors when edited versions contain no "markup" symbols and none of the visual cues offered by color pens and handwriting styles in the print medium.

For those who must rely on hard copy of an electronic document for some purposes, a linking structure can be seen as a disadvantage, as printing a branching document in a linear fashion poses both technical and conceptual problems.

Aside from the issues mentioned so far, there are some disadvantages to electronic document systems that arise from the limitations of hardware. Many people complain about eyestrain from working at a computer, even with high-resolution graphic display screens; others are attached to the "feel" and aesthetic appeal of bound volumes. Cost is still a major limiting factor to the widespread use of electronic document systems. High-resolution displays and computer systems powerful enough to run the document software are still expensive and not at all portable, especially systems that run on hardware capable of displaying both color graphics and video on the same screen.

In short, electronic document systems using today's hardware and software offer substantial advantages over paper books in providing aids for connectivity, audiovisualization, dynamics, customizability, interactivity, and rapid information retrieval, but also have a number of drawbacks in providing spatial orientation, historical tracing, joint editing, visual clarity, portability, and cost. While these limitations are not intrinsic to the electronic medium, they are problems that must be considered in the development of current and next-generation electronic document systems.

The first essential capability of a good electronic document system is to provide a means for promoting the connection of ideas and the communication between individual scholars.

Desirable features for multimedia electronic document systems

An analysis of the aforementioned pros and cons suggests a number of design features that will maximize the advantages and minimize the current disadvantages of using the electronic medium for document preparation and presentation. To meet the needs of scholars and other knowledge workers, the minimal set of capabilities incorporated into an electronic document system should include tools for

(1) promoting connectivity,
(2) promoting audiovisualization,
(3) creating and revising documents,
(4) browsing, searching, customizing, and retrieving information, and
(5) preserving the historical integrity of information.

These capabilities are explained in detail below and represent capabilities and functions for the current or next-generation electronic document systems. These systems, in effect, are databases composed of text, graphics, dynamics, interactivity, and multimedia components, in contrast to future electronic document systems that will surely be founded on knowledge bases. Stephen Weyer and Alan Borning's Electronic Encyclopedia project represents one interesting example of a prototype knowledge-based system.[6]

Tools to promote connectivity. The first essential capability of a good electronic document system is to provide a means for promoting the connection of ideas and the communication between individual scholars. These capabilities can be conceived of as a set of tools for creating a *hypertext* structure, the underlying framework of all electronic document systems developed or under development at Brown University.

The term *hypertext,* coined by Nelson,[7] denotes nonsequential writing and reading. Both an author's tool and a reader's medium, a hypertext document system allows authors or groups of authors to *link* information together, create *paths* through a corpus of related material, *annotate* existing texts, and create notes that point readers to either bibliographic data or the body of the referenced text. Hypertext can allow the creation of an automated encyclopedia of sorts: readers can browse through linked, cross-referenced, annotated texts in an orderly but nonsequential manner. For example, to learn about Greece from an ordinary encyclopedia, students must look up "Greece" and then retrieve volume after volume to find the host of other articles cross-referenced there (Acropolis, Plato, Crete, etc.). In a hypertext-based encyclopedia, however, students would browse through connected articles simply by touching the computer-equivalents of an encyclopedia's "See also . . ." references.

As early as 1945, a non-computer-based hypertext system was envisioned by Vannevar Bush, who pictured a "memex" device in which individuals would store all their books, records, and communications, form trails through them, and rapidly retrieve specific information contained within them.[8] In 1967 and 1968, Nelson and researchers at Brown University developed an early hypertext-based text-editing system[5] inspired in part by Bush's vision and in parallel with Douglas Engelbart's pioneering editing system, NLS (oNLine System). NLS, now called Augment, allows text and line graphics to be organized in a hierarchical outline structure upon which one can superimpose a network of links that point to discrete blocks of information in the document.[2,9] More recently, Donald Thursh at the University of Illinois College of Medicine used the hypertext concept to create a "Living Textbook of Pathology"[10] in the PLATO environment.

Researchers at Xerox's Palo Alto Research Center (PARC) are currently developing a sophisticated hypertext system called "Notecards" that runs in the Interlisp environment on a powerful graphics-based workstation.[11] Although the system is still under development, it already provides a large number of linking facilities, maps of the information web, and integration with graphics.

By extension, we use the word *hypermedia* to denote the functionality of hypertext but with additional components such as two- and three-dimensional structured graphics, paint graphics, spreadsheets, video, sound, and animation. With hypermedia, an author can create links to complex diagrams, texts, photographs, video disks, audio recordings, and the like.

MIT's Spatial Data-Management System (SDMS)[12] and the interactive automobile repair manual[13] are also examples of existing systems with hypermedia components. Digital Equipment Corporation's Interactive Video Information System (IVIS) is one of the first systems commerically available that allows full integration of text, graphics, computer animation, and video on the same display screen, using a specialized programming language for authoring.

To facilitate distribution of documents to a broader audience than a scholar's immediate community, multimedia document systems should provide tools for electronic dissemination.

To summarize, the basic capabilities implied by the terms *hypertext* and *hypermedia* include linking together discrete *blocks* (e.g., word, paragraph, text document, graphical object, spreadsheet cell, and video frame) to form *webs* of information, following different paths through the information webs, and attaching annotations (special types of links) to any block of information. Typically, different information blocks are created using separate *applications* or *editors*. A hyper*text* document system allows authors to link together only information blocks created with a single application, a text editor, while a hyper*media* document system provides linking capabilities between heterogeneous blocks created with different applications such as a painting program, a chart package, or a music editor. Early systems force the links to be essentially programmed, while newer systems provide interactive link creation as a fundamental component.

Multiuser access to information is another fundamental capability that must be present to promote connectivity. A group of scholars working together should be able to annotate each other's documents. This means that the system must provide multiple users with access to the same *corpus* of documents. At the least, multiple users must have "read access" to a colleague's documents; at best, they should be allowed to create new links within a colleague's document (either to annotations or to other documents) and to edit the document in a controlled fashion.

The multiuser access problem brings with it extremely complicated issues pertaining to access rights and update consistency. For instance, if user A edits a document and adds a link, do these changes show up immediately in other users' views of this document? Should all users have the ability to add links? Should all users be able to delete links? Can user C delete a document to which others users have linked? Few of the electronic book implementations have tackled this problem on a large scale (Englebart's NLS goes the furthest in providing support over national networks), and it stands as an important research question if electronic books are to be accepted.

To facilitate distribution of documents to a broader audience than a scholar's immediate community, multimedia document systems should provide tools for electronic dissemination. Currently, this would include automatic transfer of finished documents to typesetting equipment or the incorporation of powerful electronic mail or conferencing systems that make use of national and international networks. In the future, it would be desirable to subsume separate electronic mail or conferencing systems under a powerful hypertext/hypermedia system that allows the linking of documents across long-haul networks and that encourages the creation and exchange of references as opposed to files.[1]

Tools to aid in creating audiovisualization. While a text-editing system with hypertext capabilities is useful, it can be argued that the advantages of a paper book still equal or outweigh the advantages of such computer-based text systems. However, when an author can add visuals and sound—not possible in paper books—the scale clearly begins to tip in favor of the electronic format; tools for creating visual and audio components are equally as important as those for promoting connectivity. The music faculty at Brown, for example, believe that multimedia systems could revolutionize the teaching of music. With present teaching techniques it is hard for music students to make the connection between the written notes and the musical sounds without one-on-one instruction. Integrating a music editor, a piano keyboard, and a synthesizer would allow students to *hear* every note they *see* as they write or play the notes. A text-only document system—which might include many of the essential ingredients for connectivity—does not provide adequate tools for producing visual and aural aids. There is, however, a large continuum between text-only and full multimedia systems. A two-dimensional structured graphics editor (such as MacDraw for the Apple Macintosh) might be adequate for creating simple static illustrations, and a low-level graphics package might be all that is necessary for generating maps of the information web; however, more extensive multimedia capabilities are required to produce dynamic electronic documents.

There is a spectrum of media that can be included in an electronic book. Static text, structured graphics, bitmap images, charts, and graphs fall at the low end of the spectrum. The inclusion of animation, computer-generated sound, and audio and video recordings adds a richness to electronic document systems that is impossible to recreate with paper media. Moreover, each medium in such a system is subject to some level of *interactive control*—methods for readers to move, change, and manipulate the material rather than just view it. For instance, readers may want to manipulate graphical objects, as in the biological cell rotation example, or they may want to alter or experiment with animation sequences, seeking answers to "what if" questions. Readers can even become characters in a dynamic story by influencing the progress of the plot.

Authors and readers alike need visual cues that will help them determine where they are in the web of information, and also need graphical means for organizing and reorganizing their material.

As mentioned earlier, it is easy to become disoriented in a complex electronic information web. Authors and readers alike need visual cues that will help them determine where they are in the web of information, and also need graphical means for organizing and reorganizing their material. Tools that promote spatial orientation can include schematics of the information web, maps indicating all possible path options at a given time, and diagrams of specific paths a reader has already taken. For example, MIT's Spatial Database-Management System displays a "world view," an overview of Dataland with a "you-are-here" marker.[12]

For maps to be extremely useful and readable, they must be able to represent varying levels of detail. To get a general idea of how a body of information is structured, it is best not to display a detailed "road map" when a "globe" is all that is needed. To create global views, document systems must have facilities for summarizing, compacting, and extracting the essence of the stored material. However, once readers are in the midst of the web or already familiar with its overall structure, they will require more detailed road maps that show all the links in a given subsection.

For each reader, the system should always save an encoding of the current position, viewing parameters, and past travelling information—*the document state*—so that each reader of a document may pick up where he or she left off and not be forced to re-create the exact links selected and searches made to return to the current position in the information web.

The effectiveness of visuals, whether in conjuring up concrete images of complex concepts or providing a map of the information web, depends largely on the quality of the final images. Several system features have an impact on the quality of the graphics: tools that promote good graphic design, high-resolution display screens, and color. People who are not professional graphic artists may not be able to produce professional-looking images; however, with features like rulers, gravity grids, automatic justification, alignment, and some simple design rules, they should be able to create reasonable visuals.

Tools for creating and revising documents. Since electronic documents are mutable entities that can be interacted with and modified, they must contain tools not only for presenting information, but also for *creating* it. These tools can take many forms, from batch text processors to interactive paint programs, chart packages, or music editors. Most desirable, of course, is a system that provides *direct manipulation*: such systems allow authors to interactively create, edit, and format documents directly in the view in which they will be presented on the screen. This provides the ability to make well-designed documents interactively and is very close to what-you-see-is-what-you-get (WYSIWYG) editing, which refers to on-screen views that appear as close as possible (given the limitations of screen size and resolution) to the way the document would appear if printed. In these direct manipulation systems, users can be presented with one of two interface paradigms: *procedural* or *declarative*. In the procedural interface, the reader specifies exactly *how* formatting should take place, putting in typeface, margin, line length, and leading information explicitly to arrive at a desired presentation image. In the declarative interface, the user specifies exactly *what* document entities and logical structures are desired (numbered points, chapter headings, indented quotations), and the system automatically formats those entities based upon a separately supplied style sheet specifying formatting rules to be applied interactively. The declarative system removes the responsibility for formatting from the author; all that the author needs to do is specify the document entities. For text and two-dimensional graphics (free-hand structured graphics, charts, diagrams, etc.), direct manipulation editors of both types are commercially available for small computer systems; however, user-level editors for three-dimensional image creation exist only on expensive computer-aided design stations, and those for real-time animation exist only in experimental laboratories.

The creation of data can be handled in substantially different ways. Some systems have separate tools for authoring the document and for presenting it to readers. Many frame-oriented CAI programs fall into this category. Instructors (authors) are given the freedom to create frames of information and link them together sequentially or nonsequentially (by indicating branch points). Students (readers) may have the ability to interact with the lesson and sometimes to browse through the frames, but they are not permitted to alter the links or add their own connections. Futhermore, the systems almost never allow student readers to collaborate, share ideas with other students, or comment on each other's work. In some cases, the authoring tools are resident on large time-shared computers, while the final document is presented on small stand-alone microcomputers. Ideally, authors and readers should have the same set of integrated tools that allow them to browse through other material during the document preparation process and to add annotations and original links as they progress through an information web. In effect, the boundary between author and reader should largely disappear.

Ideally, authors and readers should have the same set of integrated tools that allow them to browse through other material during the document preparation process and to add annotations and original links as they progress through an information web.

It is important to consider not only the quality of the authoring tools, but also how these tools relate to one another. On one end of the *integration* continuum, documents containing different types of information (e.g., text, music, spreadsheet) are created with separate application programs, and the "electronic document" is comprised of many linked documents, each of a single medium. A little farther along on the continuum is the "cut, copy, paste" paradigm. Here, it is possible to copy bits between heterogeneous applications (e.g., paste a spreadsheet into a text document). This paradigm thus provides some integration, but loses important semantic information (all numbers in the speadsheet are now treated as text strings). The next logical step in the continuum may be called "reference copy, paste." At this level of integration, a spreadsheet that is copied and pasted into a text document retains its "spreadsheetness." If the original spreadsheet document is updated, all copies, or *instances*, of the spreadsheet are automatically updated (typically today, only data from certain applications can be used in a reference copy action).

At the highest end, a fully-integrated multimedia document system would allow reference copying and pasting of all data types and would include a context-sensitive cursor. As the author moved the cursor over the spreadsheet, all spreadsheet editing functions would be made available; as the cursor moved over some musical notation, all music editing functions would automatically be activated; and so forth. The Macintosh desktop environment[14] is one example of "cut, copy, paste" integration, while office systems such as Lotus Jazz offer several special cases of "reference copy, paste" integration.

It is nearly impossible to retrofit independently written applications into an integrated environment. Achieving the highest degree of integration requires either (1) monolithic applications that understand every type of data structure that one might want to use, (2) a universal data structure (UDS) such that any application can convert between the UDS and its own data structure (an extremely difficult problem that is prone to losing semantic information in the conversion), or (3) attaching functionality not to an application, but to the particular en-

102

tities that are the focus of attention. In such an object-oriented system, "objects" are self-contained entities encapsulating information about all possible operations that can be performed on them, thus allowing reference copying between all applications that share the same object library and object memory. The third alternative shows the most promise but is still an open issue, given the difficulty of creating a memory that allows efficient multiuser, concurrent access to very large numbers of objects, not to mention the cultural, sociological, and economic problem of getting the programming profession to switch from conventional programming to object-oriented programming.

Tools for browsing, searching, personalizing, and retrieving information. Database functions underlie all electronic document systems. Thus a useful document system should take advantage of this underlying structure and provide authors with a set of searching and filtering features, including the ability to search using a single criterion (*simple searching*), to search using multiple criteria (*Boolean searching*), or to customize the presentation in other ways.

The same criteria used for searching can also allow authors to restrict access to documents to specific readers and to groups of readers. Similarly, readers can selectively view the contents of a corpus so as to eliminate material that is not currently relevant. By providing methods for authors to associate *keywords* with any discrete block of information, a document system can filter the information for readers using these terms. Newspeek, a system developed at MIT for creating personalized newspapers, is a good example of information filtering. Readers submit lists of keywords that describe their interests. As stories come over the AP and UPI wires, the system filters stories according to each reader's list and prepares a custom newspaper for each individual.[15]

Keywords can also be associated with links to facilitate searching and browsing. A link always represents a relationship between two items. Attaching a keyword (or multiple keywords) to a link allows authors to explicitly name or define the existing relationship. Keywords associated with links might include such terms as "example" (meaning *A* is an example of *B*), "supports" (meaning *A* is an argument that supports *B*), or "child" (meaning *A* is the child of *B*).

Filtering can also be accomplished by associating attributes other than keywords to blocks and links. Different structures within documents can be named so that information can be filtered according to level of detail. An outline illustrates the point best. In NLS and Brown's FRESS system, for example, each level of an outline is identified as a separate structure, allowing readers to expand and contract the outline to see more or less detail. In text documents, sentences, paragraphs, and subject headings can all be identified as separate structures; however, it is considerably more difficult to make such distinctions in other media.

One method of encouraging electronic editing of documents, particularly if several authors or editors are working together, is to provide markup tools.

Using keywords and filters for finding specific information, hiding unnecessary detail, or tailoring the format of the material illustrate how the electronic medium allows readers to personalize the presentation of information in ways that are not possible with paper books.

The maps mentioned earlier not only provide a visual summary of the information space, but in fact are critical for browsing and information retrieval. The maps show the connectivity between documents rather than the content of the documents themselves. If these maps are supplemented with capabilities allowing users to travel by pointing to specific sections of the map, the map becomes a convenient way to move quickly through the web via the connections. This traveling can be done with the goal of discovering, for example, an appropriate detailed document or even the pattern of connections between information in the web.

Tools for preserving historical integrity of information. An electronic document system should allow authors to preserve a historical record of the creation of a document. Periodically saving versions of the document is one possible option; saving a keystroke/button push record of all updates is another. Finding viable methods for saving each draft version of a document has been one of the major research tasks of the Xanadu project initiated by Nelson.[1] Clearly, allowing authors to print their material at any time is an essential feature, particularly if the author prefers editing on paper.

One method of encouraging electronic editing of documents, particularly if several authors or editors are working together, is to provide *markup* tools. Annoland,[16] a hypertext-based system developed by Richard Burton, John Seely Brown, and others at Xerox PARC, includes an option that causes the delete operation to overstrike the target text with x marks rather than removing them from the screen. Authors can temporarily "turn off" the editing symbols and see a clean copy of the document, perhaps with insertions by other individuals appearing in a different font. Editing symbols and edit marks identified by users are two more examples of attributes that can be used for filtering information. Markup features allow an author to immediately pick out the places where changes to a document have been made; these are particularly useful when combined with filtering so that an author can see only one editor's changes at a time (just as one would be able to distinguish different editors' marks by pen color or handwriting). As multimedia document systems become more widely used, markup techniques will have to be developed for editing nontextual information.

Survey of existing systems at Brown University

Researchers at Brown University have been experimenting with electronic document systems since the late 1960's. We have chosen to describe several systems developed or under development at Brown to illustrate concrete implementations of many of the features described earlier, and we encourage the reader to extend these illustrations by examining other important work such as NLS,[2] Xanadu,[1] the systems of MIT's Architecture Machine Group,[5,12,13,15] Xerox's Notecards[11] and Annoland,[16] the Living Text,[10] and the Electronic Encyclopedia.[6]

Although each of Brown's four systems described below have markedly different user interfaces, they all have an underlying hypertext or hypermedia structure. The first and oldest of the four systems, FRESS, handles primarily text, while the current development project, Intermedia, should provide a high level of integration for a host of multimedia documents. The descriptions below highlight important or unique features of each system.

FRESS. FRESS (File Retrieval and Editing System), solely a hyper*text* document system, allowed authors to create links within any text document or among any number of text documents. FRESS was essentially a text-only system (links could be made to some graphics when using an IMLAC PDS-ID display) and was developed under the VM/CMS time-sharing system in the late 1960's as a successor to the Hypertext Editing System.[7] The system contained important "navigational," or linking, facilities. It also was designed for multiple users, although it did not support concurrent updating of a single file.

Authors created FRESS documents using a full, interactive text editor with batch formatting. They could insert a marker at any point in a text document, which became the *source*, and "link" that selection to any other *destination* point in the same document or a different one. FRESS had two types of links: *tags* and *jumps*. A tag—a one-way link—indicated a connection to a single element such as an annotation, definition, or footnote. When the reader pointed to a tag with a light pen, the associated text appeared in another window on the screen for reference while the reader remained in the main document. Unlike a tag, a jump—a bidirectional link—indicated a path to another document. By following a jump, the reader was transferred from one document to another, and up to seven windows could be used to display documents simultaneously. By inserting links, authors could create paths through a large number of documents for themselves or others to explore. Since cross-reference markers (destinations of links) were displayed in the text, readers could backtrack through a sequence of links, retracing their steps through a path.

FRESS allowed authors to attach "keywords" to links, thus providing a facility for readers to filter the information through which they wished to browse, and allowed users to name blocks of text for subsequent references and searching.

FRESS allowed authors to attach "keywords" to links, thus providing a facility for readers to filter the information through which they wished to browse, and allowed users to name blocks of text for subsequent referencing and searching. With these facilities, a student reader could choose to see only annotations left by the professor, examine only those links that led to literary criticism of a poem, ignore for the moment all the comments written by classmates, or select all poems written by a certain poet or on a certain subject.

FRESS was used in production for more than a decade at Brown for both on-line documents and hard-copy manuscript production. Two educational experiments—one in a course on energy and one in a course on poetry—were conducted using hypertext "corpuses" created in the FRESS system.[17]

For the functions FRESS provided, it had two major drawbacks. First, students in the experimental courses often did not create many of their own links because the commands necessary to establish links were somewhat complex. Second, the system provided no spatial cues, and readers found it difficult to remember where they were in the information web.

To summarize, FRESS provided a set of essential tools for promoting connectivity, and also provided facilities for customizing an individual's view of a hypertext corpus. Audiovisualization, dynamics, and integration of applications as described above are not relevant to this early system.

The Electronic Document System. Far more modern than FRESS, the Electronic Document System, completed in 1982,[18] represents Brown's first hyper*media* system. The system, which is made up of three subsystems (the Picture Layout System, the Docu-

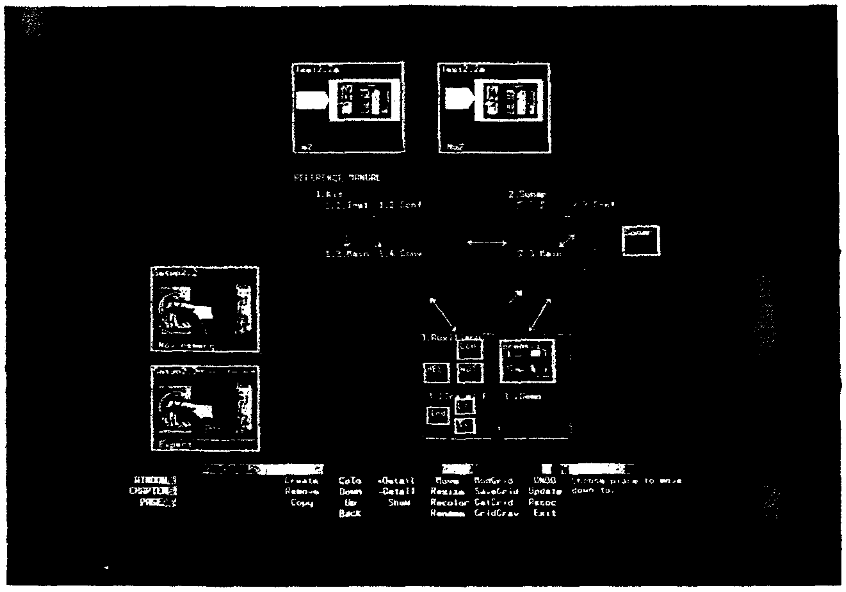


Figure 1. The Document Layout System. Authors can create the document structure (pages imbedded in nested chapters) and display arrows that represent keyworded links. (Photograph courtesy of Steve Feiner)

ment Layout System, and the Document Presentation System), was developed on a DEC VAX 11/780 and uses a Ramtek 9400 color display screen (1024 by 1280 resolution) plus a data tablet. The Electronic Document System's high-resolution color graphics capability, its graphic representation of the information web, and its ability to incorporate animation sequences set it apart from the earlier text-oriented FRESS system.

Briefly, graphic designers or artists use the Picture Layout System, a structured graphics editor, to "draw" images that may include text. Before, during, or after graphics for individual pages are produced, the author uses the Document Layout System (DLS) to construct the web structure of the document. Making use of multiple windows, authors can create blank pages and chapters and fill the pages with pictures created with the graphics editor. As necessary, authors link pages together by creating sensitive areas on the pages called *buttons*. Authors will often reserve one window in DLS to display the overall structure of the document—chapters nested as deeply as desired within chapters, pages nested within chapters, and keyworded links connecting the pages. Special keyworded windows can also be created so authors may see only links labeled with a particular keyword (Figure 1).

In the Document Presentation System (DPS) readers select buttons with a pointing device to traverse links created in the Document Layout System. Actions other than "get new page" can be associated with buttons. Most notably, buttons can activate dynamic actions such as triggering animation sequences (animation scripts are created outside the system using DIAL[19]), exiting from the system, turning on graphic overlays, or activating the annotation facility.

To date, two prototype maintenance-and-repair manuals have been created using the Electronic Document System: one for a computerized numerical control system and one for naval sonar equipment. A reader traversing through the sonar manual, for example, either selects choices from a menu or picks the "NEXT" button located at the bottom of every page to follow a path through the document.

An author uses the Document Layout System to predefine a number of different paths for different classes of readers. In this particular document, the author has created "novice," "intermediate," and "expert" paths, and by using keyworded links has been able to further customize each reader's experience depending on the choices each reader makes and actions that have occurred during the current session. Specifically, if a novice wishes to see an explanation of power supply failures, he or she is automatically given a step-by-step procedure to follow, while an expert requesting the same information is given a much briefer checklist of trouble-shooting procedures.

The most interesting aspects of the Electronic Document System include the graphical representation of the underlying information web provided for authors and the two forms of "maps" provided for readers. The first type of map provides readers with a history of their path through the document in the form of a "timeline" with miniatures of the pages visited and allows them to select any page's icon and return to it. The second type of map helps to orient the reader spatially. In the center of the "neighbors" screen, the system displays a miniature of the reader's current page. On the left side, it shows miniature pictures of all the pages that the reader could possibly have come from, on the right side, all the pages to which the reader can currently go (Figure 2). Again, all icons may be selected as the "next" page. While this feature is useful, it falls short of helping the reader to visualize the entire system of connections in the document, as can be done in the Document Layout System.

Another interesting feature included in the graphics editor is a "reference copy, paste" capability. If the need arises to use any drawing more than once, it is possible to create "instances" of the drawing (which can be scaled) in any number of other Picture Layout documents. Editing the original drawing will cause the changes to appear in all instances of it.

The separate drawing, authoring, and viewing facilities, and the absence of multiuser capabilities, three-dimensional modeling capabilities, and a text editor in the suite of applications that make up this system are severe disadvantages, as is the considerable computing power necessary to allow the system to run at a reasonable speed. On the positive side, the Electronic Document System permits both readers and authors to perform all operations graphically without requiring a text editor or a special command language for specifying links.

BALSA: an introduction to nuclear arms issues: BALSA (Brown Algorithm Simulator and Animator) is not in fact a general-purpose electronic document system in our sense; rather, it is an environment designed to facilitate the creation of computer science educational software. It has a number of powerful features for animating complex algorithms such as sorting, searching, binary tree creation, etc.[20] BALSA is included here because an educational program created within the BALSA environment for a course on arms control has many of the features of an electronic book.

The program, "An Introduction to Nuclear Arms Issues," organizes a body of information around a *timeline* (Figure 3).[21] Each event on the timeline is linked to an expanded textual explanation of the topic, one or more graphic images that are sometimes animated or interactive, and lists containing suggestions for further reading, possible essay topics, footnotes, and definitions of technical terms. Students are able to browse through the topics on the timeline in any order,

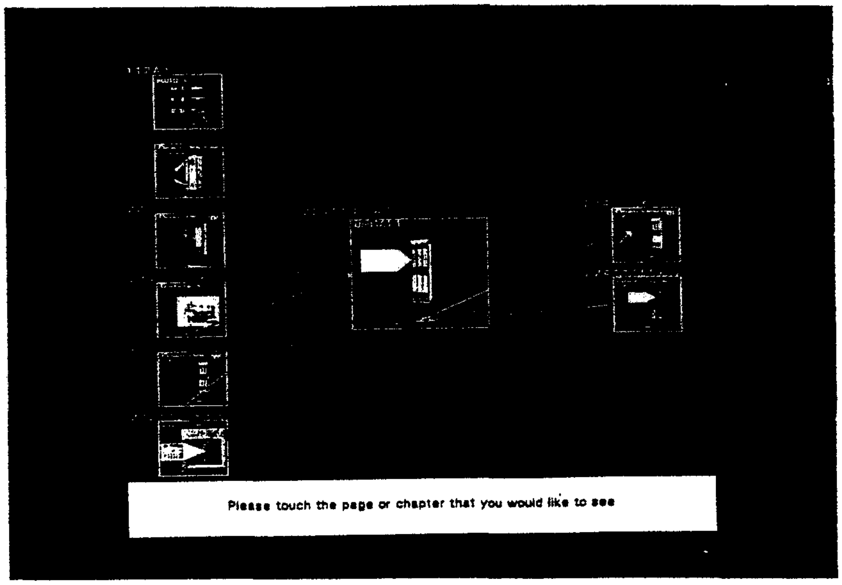


Figure 2. The Document Presentation System. In the "neighbors" display, the center page is surrounded by miniatures of its predecessors (on the left) and successors (on the right), each of which is nested in a box representing its chapter. Arrows emanate from the center of buttons on the center page and its predecessors and terminate on the pages they can access. Touching a page makes it the current page. If there are more neighboring pages than will fit on the screen, arrows will appear at the bottom to scroll the side columns. (Photograph courtesy of Steve Feiner).

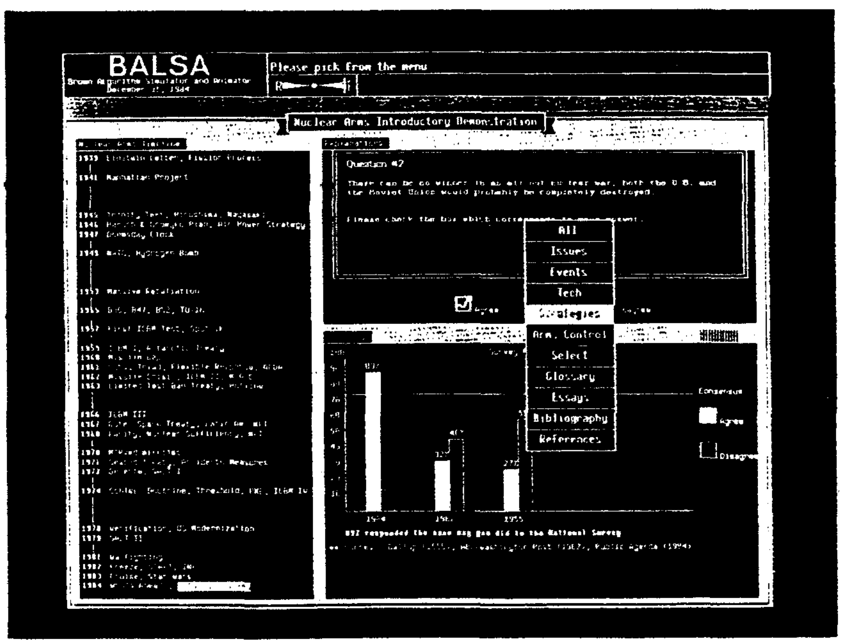


Figure 3. "An Introduction to Nuclear Arms Issues" runs in the BALSA environment. The "National Survey" topic is highlighted on the timeline and the pop-up menu shows the five filters as well as supplementary information that is available for all timeline topics.

106

or choose from five informaton filters to customize the information presented to them. A student wishing to review only technical information about weapon systems would select the "TECH" filter from the pop-up menu. Hitting the return key would then cause the student to travel sequentially through all blocks of information related to weapons systems. As the student progresses through the material, the current topic is highlighted on the timeline to provide spatial and historical orientation.

Several events on the timeline are linked to interactive material. A student who selects "National Survey" from the 1984 position on the timeline, for example, is presented with a series of survey questions to answer. After answering each question, the student is provided with (linked to) a graph indicating how his or her answer compares to the national sample and a block of text discussing the particular statistic.

The major difference between the nuclear arms program and the two systems described previously is that this program provides no user-level authoring tools. A programmer is required to add the text, create the graphics, program the interaction, and edit the timeline. The timeline structure, however, has provided one useful model for structuring a body of related information. Unlike the history and neighbors diagrams in the Document Presentation System, the timeline provides readers with an overview of the entire information web.

Intermedia. Intermedia, currently under development at Brown's Institute for Research in Information and Scholarship, is a multimedia system that will ideally provide most of the major capabilities desirable for a good electronic document system. An assumption underlying the design of Intermedia is that users will be most likely to take advantage of the system's capabilities if they can create links as part of their regular work with spreadsheets, word processors, graphics editors, or other media. Therefore, Intermedia is not a separate application, but rather a framework for a collection of tools that allow authors to make links between standard types of documents created with heterogeneous applications.[22]

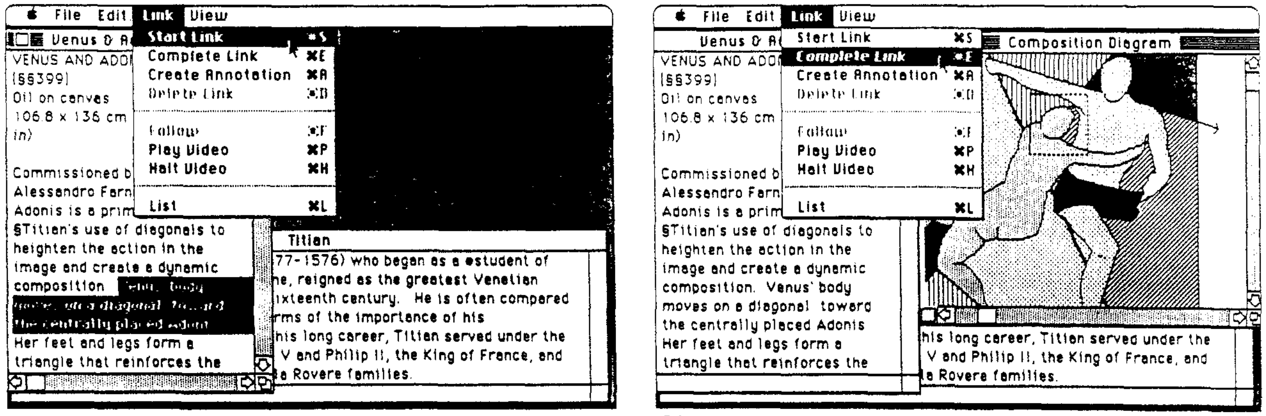
The standard type of user environment supports an integrated application domain at the user level by providing the basic ability to cut or copy text (and sometimes graphics) from one application to another.

The structure of the Intermedia user environment facilitates connectivity, audiovisualization, creation of material, and complex filtering of information. Its architecture is similar to the window-based desktop environment provided on the Apple Lisa, in which multiple applications run simultaneously. Each application allows the user to create material of different types: a text-editing application is used to create text documents, a music editor is used to create musical scores, and a spreadsheet application is used to create financial models. An application can either be "off" (not being used) or "running," meaning it has opened one or more windows and provided an interface to create or modify material. Multiple applications of the same type (e.g., multiple music editors) can run simultaneously.

The material an application creates is the *document*, a part of which is viewed in the application's window. While the application is running, the document can be updated interactively, and is edited and viewed in the application window, a facility adhering to the principle of direct manipulation. When the application terminates, the document can be stored on disk for later retrieval by the application. The *desktop manager* (also known as the display manager, the graphical shell, the window manager, or the finder in various systems) provides the ability to name, open, close, move, and organize applications, windows, and documents.

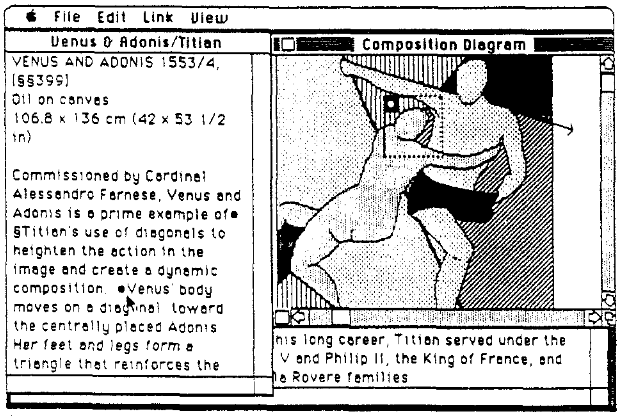
This standard type of user environment supports an integrated application domain at the user level by providing the basic ability to cut or copy text (and some graphics) from one application to another. In copying and pasting, one picks a source selection, chooses the copy command, picks a destination selection, and chooses the paste command to paste the source selection at the destination selection. When the operation is finished, there is a duplicate of the source at the destination, and no semantic tie between the two. The process of creating links mirrors the copy, paste operation—the most familiar command in the standard desktop. One simply picks a source selection, called a *block*, chooses the "start link" command (Figure 4a), picks a destination block, and chooses the "complete link" command (Figure 4b). When the operation is finished, there is a bidirectional tie between the source and destination blocks (Figure 4c) such that whenever a user selects the source block and issues the "follow" command, the document containing the destination is activated, with the destination block highlighted in another window on the screen and the destination document retrieved from storage if necessary.

The applications now under development and the documents they generate include a text processor, presentation-graphics editor, paint-graphics editor, constraint-based animation application, timeline editor, geographic map editor, videodisc access building block, and subject-specific building blocks.



(a)

(b)



(c)

Figure 4. Intermedia Prototype. (a) In a corpus of art history materials, the user has already created a link from a biography of Titian to a detailed article describing one of his works. The user now highlights a portion of text within the article and then selects the "Start Link" command. (b) The user opens the Composition Diagram and selects a portion of the bit map. This selection will become the destination of the link after the "Complete Link" command is issued. (c) A link is established between a selection of text in the article on Venus and Adonis and the Composition Diagram. The program places bullet markers at both the source and destination points to indicate the existence of a link.

In addition to the linking capability, many other features are included to aid in navigation and information retrieval. A user can create *paths*, named trails, through a particular set of links. After issuing the command "start me on the Novice path," a user can issue a "next link" command that will automatically jump to the next item in the path, or the "previous link" command to go to the previous block. A special *system path* keeps track of users' entire Intermedia sessions (like the history facility in the Document Presentation System) so that they can take side trails serendipitously and go back, step-by-step, the way they came.

Keyworded blocks and links are also integral to the Intermedia environment. Keywords applied to links allow the user to attach one or more attributes to a link; later users can apply filters to the document so that only link symbols meeting their filtering criteria are shown on the screen. Similarly, users can filter blocks within a document according to simple or Boolean search arguments. The result of a search can be stored in a *search list* for later perusal.

Intermedia blocks, links, keywords, and paths associated with a set of documents are stored in a *web*. The web is essentially a way of defining a context or database in which a typically large set of related block, link, keyword, and path information is stored. The web allows individual users, groups of users, or, in the extreme, an entire campus to create a shared web of materials from different applications. Two webs might exist that reference the same documents but have an entirely different set of links. For example, the English Department might have a web referencing all of the Shakespearean tragedies along with links pertaining to color imagery in those plays, while the Religious Studies Department might have a web referencing those same plays with links pertaining to religious symbolism.

The system also provides ways to generate diagrams of the information web at different levels of detail. A *global map* gives a high-level view of the web, showing links at a document-to-document level. A *tracking map* provides a continually updated graphical view of the currently active document and its links, and can be viewed at the following four levels of detail: (1) document-to-document; (2) document-to-block; (3) block-to-document; or (4) block-to-block. With each document, one can store filtering and viewing attributes so that a document's map, rather than its contents, can be viewed by the user; the map is generated on the fly. By providing easy-to-use facilities for linking, saving paths, retracing paths, and viewing the information web at several levels of detail, Intermedia promotes connectivity by combining the most significant features of FRESS and the Electronic Document System.

While facilities for visualizing the web structures are provided by the Intermedia framework itself, other audiovisualization tools such as graphics editing, three-dimensional modeling, sound generation, and animation must be provided by individual applications within the framework. For example, a "timeline" application could be included as an additional aid to viewing the organization of the information visually, and an audiodisc control application could be added to allow blocks to be linked to audio segments. Any multimedia component can be included in the Intermedia environment if its design adheres to the standard user interface conventions of the system and if it includes the standard methods for users to identify discrete blocks of information that can serve as source and destination points for links and allows users to create and travel through the links themselves.

The Intermedia user environment, although still at the prototype stage, already provides most of the necessary linking facilities included in Brown's early electronic document systems, including text, line graphics, and videodisc applications. An early version of the final system is scheduled for testing in an English literature course beginning in January 1986.

Multimedia electronic documents ments can be enormously useful adjuncts to the existing teaching, research, and learning tools of scholars, but to do this they must provide facilities that are different from and more powerful than those of paper books. These facilities, which include tools to promote connectivity, enhance audiovisualization, aid in the creation and revision of information, facilitate the search for and retrieval of data, and maintain historical integrity of materials, represent the set of features that the electronic document systems developed at Brown aim to encompass. Not all, however, can be accomplished fully with existing technology.

Until hardware and software become standardized, research and standards in the area of document exchange formats and systems that tailor the electronic document presentation to the type of hardware currently in use are of primary importance.

Five areas, in particular, point to the need for further research. The first, concurrent distributed file access over a network, is a problem that must be tackled before sophisticated tools for joint authoring can be included in electronic document systems. The concurrent distributed file access problem is particularly acute in cases when a group of scholars would like the freedom to edit documents in the information web while others are editing, reading, or creating links to and from the same documents. Distributed database techniques will have to be developed and incorporated into multimedia document systems intended for multiple users.

Related to this challenge is the goal of high-level integration of applications. Methods must be found for software developers working separately to create multimedia applications that work together in a consistent user environment. The most promising current technology is that of object-oriented programming.

Third, the art of graphic design for electronic multimedia presentation of information is still young. Today's graphic design specialists concern themselves only with the linear presentation of material, while graphic design for electronic media introduces the elements of change and time. How should a document be presented on the screen if each reader sees a unique sequence of information blocks, and how should multimedia components be combined to present the reader with the pleasing, crisp appearance commonly expected by readers of books?

Fourth, the graphic design issue raises questions of standardization. User interface standards are needed in order to give users a sense of familiarity with the electronic tools, but these standards must, at the same time, be flexible enough to accommodate the full range of applications any one scholar requires.

Last is the problem of interoperability. If the proliferation of hardware and software systems continues, incompatibility may mean that electronic books run the risk of being usable on only a small number of systems. Currently, different readers have available machines with substantially different software, memory, and input and output capabilities. Over the five centuries since the print medium was introduced, a high degree of standardization has been developed that allows books to be used universally. Although it may not take quite so many years to develop a universally accepted format for electronic books, interoperability still looms as a major obstacle to electronic book publishing and dissemination on a wide scale. Until hardware and software become standardized, research and standards in the area of document exchange formats and systems that tailor the electronic document presentation to the type of hardware currently in use are of primary importance.

The pursuit of these goals will be a challenging one. At Brown, by separating application-specific components from an application-independent linking structure in Intermedia, we hope to take a first step toward providing a framework for multimedia systems for the knowledge worker. With the current upsurge of high-quality work in this area at other institutions, we expect the remainder of this decade to be a fruitful one for this important field. □

Acknowledgments

We wish to thank the many researchers at Brown University involved in the development of the four systems described in this article. In particular we would like to single out all of the researchers responsible for developing and enriching the FRESS hypertext system; Steve Feiner, David Salesin, Sandor Nagy, Randy Pausch, Gerry Weil, Imre Kovacs, Charles Tompkins, and Steve Hanson for their work in creating the Electronic Document System and the Sonar Maintenance and Repair Manual; Bob Sedgewick and Marc Brown for creating BALSA and many algorithm animations; Liz Waymire for the development of "An Introduction to Nuclear Arms Issues" within the BALSA environment; and Steve Drucker, Page Elmore, Charlie Evett, Matt Evett, Nan Garrett, Ed Grossman, Bern Haan, and Karen Smith for their contributions to the design of the Intermedia environment. We owe thanks to Helen DeAndrade for creating an art history corpus for the Intermedia prototype system. Lynne O'Brien and Trina Avery deserve special thanks for their insightful comments and careful editing of this article. We are also indebted to our reviewers for raising important technical issues and providing constructive suggestions for revisions.

We would like to thank IBM, the Annenberg/CPB Project, the Office of Naval Research, the Exxon Education Foundation, and the National Endowment for the Humanities for their generous financial support of the various projects described in this article.

References

1. T. H. Nelson, *Literary Machines*, Swarthmore, Penn., 1981. Available from author.
2. D. C. Engelbart and W. K. English, "A Research Center for Augmenting Human Intellect," *Proc. FJCC*, Vol. 33, No. 1, AFIPS Press, Montvale, N.J., Fall 1968, pp. 395-410.
3. T. H. Nelson, "Replacing the Printed Word: A Complete Literary System," *Information Processing 80*, S. H. Lavington (ed.), North-Holland Publishing Co., IFIO 1980, pp. 1013-1023.
4. A. van Dam and D. E. Rice, "Computers and Publishing: Writing, Editing and Printing," *Advances in Computers*, Academic Press, New York, 1970.
5. N. Negroponte, "An Idiosyncratic Systems Approach to Interactive Graphics," paper presented at ACM/SIGGRAPH Workshop on User-Oriented Design of Interactive Graphics Systems, Pittsburgh, Penn., Oct. 14-15, 1976.
6. S. Weyer and A. Borning, "A Prototype Electronic Encyclopedia," *ACM Trans. Office Information Systems*, Vol. 3, No. 1, Jan. 1985, pp. 63-88.
7. S. Carmody et al., "A Hypertext Editing System for the /360," *Pertinent Concepts in Computer Graphics*, M. Faiman and J. Nievergelt (eds.), Univ. of Illinois Press, 1969, pp. 291-330.
8. V. Bush, "As We May Think," *Atlantic Monthly*, July 1945, p. 101.
9. N. Meyrowitz and A. van Dam, "Interactive Text Editing Systems: Part 1," *Computing Surveys*, Vol. 14, No. 3, Sept. 1982, pp. 321-352.
10. D. R. Thursh and F. Mabry, "A Knowledge-Based System for Pathology Education," *Bull. Pathology Education*, Vol. 6, No. 2, Fall 1980, pp. 36-45.
11. F. Halasz and R. Trigg, personal communication.
12. R. Bolt, *Spatial Data-Management*, Architecture Machine Group, M.I.T., Cambridge, Mass., 1979.
13. D. Backer and S. Gano, "Dynamically Alterable Videodisk Displays," *Proc. Graphics Interface 82*, Toronto, May 17-21, 1982, pp. 365-37.
14. *Inside Macintosh*, Apple Computer, Inc., Cupertino, Calif., 1985.
15. W. Bender, "Imaging and Interactivity," *Fifteenth Joint Conf. Image Technology*, Tokyo, Nov. 26, 1984.
16. J. S. Brown, "Process versus Product: A Perspective on Tools for Communal and Informal Electronic Learning," in *Education in the Electronic Age: A Report From the Learning Lab*, WNET/Thirteen Learning Lab, New York, 1983, pp. 41-58.
17. J. Catano, "Poetry and Computers: Experimenting with Communal Text," *Computers and the Humanities*, Vol. 13 (1979), pp. 269-275.
18. S. Feiner, S. Nagy, and A. van Dam, "An Experimental System for Creating and Presenting Interactive Graphical Documents," *Trans. Graphics*, Vol. 1, No. 1, 1982, pp. 59-77.
19. S. Feiner, D. Salesin, and T. Banchoff, "DIAL: A Diagrammatic Animation Language," *IEEE Computer Graphics and Applications*, Sept. 1982, pp. 43-54.
20. M. H. Brown and R. Sedgewick, "A System for Algorithm Animation," *SIGGRAPH '84 Conference Proc.—Computer Graphics*, Vol. 18, No. 3, 1984, pp. 177-186.
21. N. Yankelovich et al., "The Sampler Companion," IRIS Technical Report 85-1, Inst. Research in Information and Scholarship, Brown Univ., March 1985.
22. N. Meyrowitz, "Networks of Scholar's Workstations: End-User Computing in a University Community," IRIS Technical Report 85-3, Inst. Research in Information and Scholarship, Brown Univ., June 1985.