BUILDING HYPERMEDIA APPLICATIONS
A Software Development Guide
GARY THOMAS HOWELL
The word Hypermedia, as illustrated in Figure 1.1, is a fusion of Hypertext and Multimedia. It follows that Hypermedia technology is a fusion of Hypertext technology and the concept of Multimedia.

![Hypermedia Diagram](image)

Figure 1.1 Hypermedia—the word.

This chapter will define these two separate entities and the terminology surrounding them and will explain how they come together to form Hypermedia.
The term Hypertext was first coined by Theodor Holme Nelson, recognized ideologist of Hypertext and author of "Xanadu" (a Hyper-text engine), in reference to a radically new way of storing and viewing information. Before Hypertext, the most common way to gather information from text was sequentially (i.e., a book is read from front to back). However, when attempting to retrieve information quickly, the text is not read from cover to cover. Generally, you proceed to the index, look up the subject of interest, and go directly to the location of the pertinent information. If you don't find the exact information you are looking for in that location, then go back to the index to find another reference. This goes on until you either find the information or get tired of looking.

With Hypertext, the text is fashioned in multiple layers. In reference to the previous example, the pseudo-index would be the top layer and the locations of the information would be at the next level. The text could be ordered in a pyramid-like structure (see Figure 1.2), or in something a little less structured (see Figure 1.3). The point being, the hierarchy of information is designed to fit the application, not some predefined layout.

![Figure 1.2](image_url) Standard pyramid hierarchy of data.
Generally speaking, Hypertext applications are written to run in a windowing environment (i.e., there can be multiple windows or text viewing ports open at any given time). Applications are mouse driven; there is no command line interface. Actually, Hypertext is simply composed of points, nodes, and links (see Figure 1.4).
Located within the text are "hot spots" or "buttons" which provide the links to other related information (points). If you wish to know more about a particular subject, move the mouse over the subject and click. If there is more information on the subject, it is displayed as a point either in the same window or in another window that is popped up. There is always a way to get back to the previous level and the new information can also contain hot spots and/or buttons to allow further investigation into the subject. In theory, there is no limit on the size or location of the document you are searching. Hypertext can link together separate documents and even link over networks if necessary.

From this explanation, it is obvious that with Hypertext you have an automated index built in to the document. Related pieces of information (points) or related chunks of text (nodes) are connected together by pointers to other pieces of information (links).

Points, Nodes, and Links

Points, sometimes called link anchors, are usually the most fundamental piece of information in a document. They consist of either a word or a phrase that can be expanded upon. In order for the viewer of a Hypertext document to visually distinguish between regular text and a point, the cursor (mouse, pointer, etc.) will typically change shape when it passes over a point. It is not essential that all Hypertext cursors change to the same shape when they pass over a point. It is important, however, that within a single Hypertext document consistency is maintained. For example, you wouldn't want the cursor to change to a pointed finger when passing over a point in one part of a document and then switch to another convention, like a circle, in a separate part of the document. If you maintain the same cursor conventions throughout your Hypertext document, the viewers will not become confused or frustrated and will be more likely to use the application as an integral part of their day-to-day activities.

A node is a container for related pieces of information. Currently, the most popular convention for personal computers is that of a note card representing a node. These cards, while not limited in size, were originally designed to represent those 3 x 5 cards some of us used to study in school. As a study aid, you would write down one concept per card and then flip through them in order to commit the concepts to memory. The idea is similar with Hypertext. There is generally one concept to a card or node, but there can be many points within the node which lead to other nodes of related information. You, as the viewer, can simply flip through the cards sequentially, but you are not limited to merely flipping through the cards.
What is Hypermedia? 7

Figure 1.5 Types of Hypertext links.
By using the points within the nodes, you can navigate your way through the information, extracting the details which are important to you.

Links are the glue that hold Hypertext together. Nodes can be linked together in a sequential manner for simple card flipping as described earlier. Points can be linked to other points within a node, or to a separate node. These links allow the nonsequential navigation of Hypertext documents. From a programming point of view, links are simply pointers to a source of information. When a point is clicked upon, the link tells the program what node to present to the viewer or where to go in the current node. Several different types of links are illustrated in Figure 1.5.

MULTIMEDIA

The media part of Hypermedia comes from multimedia. The different media forms most generally associated with Hypermedia are text, graphics (including drawings and pictures), animation, sound, other computer programs, and moving pictures (see Figure 1.6).

![Figure 1.6 Media forms in general order of cost in producing Hypermedia applications.](image-url)
**Multimedia—Text**

Text is by far the easiest and least expensive media form to use in Hypermedia. There are even computer programs available that can take your existing text files and put them in Hypermedia format. While plain text Hypermedia might seem a little dull, you can liven it up a little with multiple fonts, font sizes, attributes, and even colors if your system permits. But even without these additions, plain text Hypermedia is still functional. In fact, it is more than functional. Plain text Hypermedia is Hypertext which became so popular it spawned Hypermedia. From a legal point of view, text is also the easiest and least expensive media form. Simply use common sense: if you quote a significant portion of a work, then get permission to do so. But remember to give reference to your information source.

**Multimedia—Drawings and Pictures**

The least expensive way to move out of Hypertext and into Hypermedia is by incorporating drawings and pictures into your application. With these, your applications can leave the realm of simple databases and card catalogues and move into the domain of manuals and documentation. As an example, television repair manuals could have schematics and pictures of the actual TV parts as points and nodes in a Hypermedia application. Pictures do, however, tend to be a little more expensive than drawings. You would be required to have a scanner to convert the pictures into an electronic format. And, pictures take up a great deal more hard disk space than drawings, thus requiring larger disks. Legally, drawings and pictures are not much worse than text. Self-created drawings and pictures are obviously not a problem at all. If you are using someone else's drawings or pictures, get permission. Also, still photographs are the first category in which it may cost you to purchase the rights to use the media.

**Multimedia—Animation**

Animation adds something entirely new to Hypermedia: motion. Simple animation can be only moderately more expensive to create than drawings or pictures. And, it can sometimes be done by a single person at a graphics workstation. Large scale, artistic animation, on the other hand, can be extremely expensive, rivaling the cost of motion pictures. From a legal perspective again, self-created animation is free. But if you intend to use someone else's anima-
tion, or even portions of it, you most likely will be asked to pay for the rights to use it.

Multimedia—Sound

Sound is another media form that can add greatly to your Hypermedia application. The type of sound you use can vary a great deal depending on the quality of sound you desire. The cost of using sound can also vary greatly from a few hundred dollars for the equipment to record and playback voices and simple sounds to tens of thousands of dollars for the rights to use original recordings from popular artists. Disk space is another consideration when you are contemplating using sound. Sounds, even when using expensive compression utilities, take up tremendous amounts of disk space.

Multimedia—Computer Programs

The use of computer programs, aside from the one you use to author your application, can add a great deal of utility to your Hypermedia application. You can use these programs to allow users to add their own text and graphics to the application. The computer programs most commonly used for this are word processors and scanner drivers. This utility, of course, comes at a price. You could be asked to pay a great deal for the right to use a commercial computer program within your commercial Hypermedia application. In fact, you may be denied permission. One way around this problem is to inform the buyers or users of your application that in order to use your product most efficiently, they must purchase their own copy of whatever software you prescribe.

Multimedia—Moving Pictures

Incorporating film and/or video into your Hypermedia application is by far the most expensive of the media forms. In fact, without some significant breakthroughs in storage and playback technologies, moving pictures are beyond all but the most wealthy software developers. Assuming that money is not a problem, there are still problems with the quality of the film when played back from a laserdisc due to inferior conversion methods and/or foreign standards.

HYPERMEDIA

Hypermedia is created when you take the previously mentioned media forms and put them in Hypertext format. However, there is
no official Hypertext format. The author of a Hypermedia application can format the flow of information any way he or she wishes. It is important that the Hypertext concept is followed while developing the application and that you consistently use whatever conventions you choose. The Hypertext concept means that your application has definitive points and nodes, and that the links allow the viewer to create his or her own path through the information. The viewer should not be forced to view the data in a predefined manner. In fact, a Hypermedia author can allow viewers to create their own points, links, and nodes within the existing document. With this in mind, the rest of this chapter will expand on what is involved when putting different media forms into Hypertext format.

Hypermedia—Text

Text is obviously the basis for most Hypermedia applications. While textual documents can be put in Hypermedia format by hand, there are utilities capable of doing this automatically. And, for large documents, this should be further investigated. However, whether formatting is done by hand, or automatically, you must establish absolute conventions within your application and adhere to them. These conventions are the look and feel of a node, and the point identifier.

Consistency in look and feel of a node includes what the node looks like and how the viewer moves between nodes. A text field should have a standard look and it must be obvious to the viewer if the field is editable. Also, maintain the same button in the same location for moving between sequential nodes.

It is very important to keep the point identifier consistent. You are liable to confuse and frustrate your viewers if points in one part of your application are identified in bold and in other parts they are not different from surrounding text. It is a good idea to double identify points to the viewer. Double identifying involves varying the textual attributes of a point to visually differentiate it from surrounding text. And, have the cursor change shape when it passes over a point. Please note, however, that it can be very annoying for the cursor to change into something different on every node. Find two cursor styles you like and stick with them.

Hypermedia—Drawings and Pictures

The introduction of drawings and pictures can result in tremendous enhancements to an all text Hypermedia application. The key to accomplishing these enhancements is incorporating the graphics completely in the application. Don't limit your drawings and pic-
tures to nodes. Place points on them as well and allow the viewer to interact with the graphics. Point-to-node links and point-to-point links with drawings and pictures (see Figure 1.7) can assist greatly in viewer comprehension. These types of links allow the viewer to establish a pictorial reference to the textual data as a memory aid.

![Point-to-point link](image1)

![Point-to-node link](image2)

**Figure 1.7** Effective links with graphics.

**Hypermedia—Animation**

While animation can add a lot of flash to your Hypermedia presentation, it is difficult to actually incorporate it completely into the application. Animation can make a single node quite memorable for the viewer, but, establishing moving points in animation is nearly impossible. And, those viewers with less than wonderful hand to eye coordination would be quite irritated having to chase animation around the screen with the mouse. Stick to using animation to visually enhance your application. Keep the buttons or points stationary and use them to launch the animation or use the animation as side-
line entertainment on a node which contains otherwise mundane information.

Hypermedia—Sound

Sound in Hypermedia applications has a problem similar to that of animation. The problem, though, is more difficult to solve. It is impossible to click on sound. Even if you could, it would not be particularly useful. Sound, for Hypermedia purposes is limited to enhancement of the overall viewing experience. In some applications, musical perspectives for example, the sounds could actually be the nodal information. But for the most part, sounds are relinquished to background music.

Incorporating sounds other than simple two to three-second beeps and boings into your application brings with it other problems. Sound is the first media form considered which may require special purpose hardware. If you are recording your own sounds, input hardware and sound editing software must be purchased. And, even if you are only playing prerecorded sounds, you may require sound output hardware depending on the type of computer you are using (IBM PC and compatible computers have a sound system which is not designed for much more than beeps and boings). Another problem associated with sound is disk space. Fifteen to twenty seconds of sound recording can take up several megabytes on a hard disk. A portion of this problem can be relieved by sound compression utilities, but some sound quality is sacrificed to free up disk space.

Hypermedia—Computer Programs

Computer program use within Hypermedia applications is generally limited to special purpose input and output programs. There is great advantage to using off-the-shelf software for this purpose, but there are licensing problems as discussed previously. However, if the author of the application wishes the viewer to be able to add to the presentation, these problems can be overcome.

Hypermedia—Moving Pictures

Motion pictures in Hypermedia have similar problems as both sound and animation-only much worse. The technology that is currently available virtually limits video to playback. There is talk that Apple Corporation is about to unveil a new technology which will bring inexpensive interactive video to the masses sometime in 1992. In theory this technology will be available for the Apple Mac-
intosh series of computers and would be ported to whatever platform Apple and IBM Corporation decide upon in their upcoming collaborative venture. However, without this or some other major technological breakthrough, interactive hypervideo is beyond the financial scope of 99.99 percent of software developers and, therefore, will not be covered in this book.