
UNIT 3 MULTIMEDIA AUTHORIZING TOOLS

Structure	Page No.
3.0	Introduction
3.1	Objectives
3.2	Multimedia Development Tools
3.3	Features of Authoring Software
3.3.1	Integrating Multimedia Elements
3.3.2	Script Language Programs
3.3.3	Icon-based Programs
3.3.4	DLLs for Extending Features
3.3.5	Supporting CD-ROM or Laser Disc Sources
3.3.6	Supporting Video for Windows
3.3.7	Hypertext
3.3.8	Cross-Platform Capability
3.3.9	Run-time Player for Distribution
3.4	Authoring Tools
3.4.1	Authorware
3.4.2	Everest Authoring System
3.4.3	Icon Author
3.4.4	ImageQ
3.4.5	Macromedia Director
3.5	QuickTime
3.6	Hypertext
3.7	Applications of Hypertext
3.7.1	Computer Applications
3.7.2	Business Applications
3.7.3	Educational Applications
3.7.4	Entertainment and Leisure Applications
3.8	Elements of Hypertext
3.9	Review Questions
3.10	Summary
3.11	Answers/Solutions

3.0 INTRODUCTION

In the previous two units, we have discussed about the definition and components of multimedia and its applications. A multimedia production need to be interactive. In this unit, we will try to answer the question how multimedia production can be made interactive. Please note that discussions on the software used for development of each component of multimedia have not been discussed here. Several tools are used for developing interactive multimedia. These are:

- authoring systems
- authoring languages
- generic languages (e.g., Fortran, Pascal, C)

The term authoring results from the fact that the person creating the learning material on the computer is often called the author.

Authoring systems are generally menu-driven systems intended to be easy to use by the multimedia developer. They do not require the trainer to get involved with program code. Such systems endeavour to provide a complete, comprehensive

authoring environment where whatever multimedia developer wishes to do is either selected from a menu or entered into the computer in a straight forward, well-prompted way.

Authoring languages are specially designed programming languages, and have some distinct advantages over generic languages. They have been created specifically for the training/educational field, and as such provide appropriate commands and structures to cater for the types of presentation and interaction involved in CBT.

In this unit, our main focus shall be on the authoring systems.

3.1 OBJECTIVES

At the end of this unit, you will be able to:

- categorize the various multimedia development tools;
- discuss in detail the various multimedia authoring tools;
- describe the working and importance of QuickTime and its relationship with other authoring tools;
- discuss the need of hypertext and how it is accomplished in multimedia; and
- discuss the various elements and applications of hypertext.

3.2 MULTIMEDIA DEVELOPMENT TOOLS

A multimedia development environment is the glue that combines different media elements-images, sounds, text and animation into a coherent project that the user can explore. Although programs such as Photoshop and Infini-D enable the user to create bit-mapped graphics or three-dimensional illustrations, they don't allow to add buttons or data fields. This really limits the productions created by the user, and that's why he'll nearly always want to have access to some kind of development environment such as HyperCard, Director. etc.

The presentation tools available for multimedia development can be divided into three categories:

- 1) Slide Presentation Environments
- 2) Interactive Presentation Environments
- 3) Authoring, Environment

Programs such as HyperCard, Director etc. add another element to be multimedia developer's arsenal i.e. scripting. Scripting really means programming, because you write "scripts" using a programming languages that the development environment understands. The scripts, or programs perform various functions such as importing a text file or sorting a list of words.

With scripting it is possible to create very complex interactions between the user and the presentation. For example, by clicking. a single button, the user might inquire about a subject and then find all the information available on that subject in the presentation.

The authoring, environments can be used to create slide presentation or simple interactive presentations but what sets them apart from the other classes of programs is the ability to write scripts and create complex interfaces.

3.3 FEATURES OF AUTHORING SOFTWARE

An Authoring Software is the main production tool for multimedia. Just a layout program for printing, an authoring software brings all media elements together and compose them so that they look, and sound right. The main features of an authoring software are:

- 1) Integrated Multimedia Elements
- 2) Script language programs
- 3) Icon based programs
- 4) DLLs for extending features
- 5) Supporting CD-ROM or laser disc sources
- 6) Supporting Video for Windows
- 7) Hypertext
- 8) Cross-Platform Capability
- 9) Runtime Player for Distribution

3.3.1 Integrating Multimedia Elements

It is essential for an authoring software to support a wide area of format with different media elements. The authoring software must be supported by add-on boards and peripheral devices to play multimedia. The final feature of authoring is to build interactivity.

- 1) Authoring programs provide developers with a greater degree of control over each multimedia element than presentation programs.
- 2) Authoring programs are geared towards education and training applications as well as business presentations.

3.3.2 Script Language Programs

Authoring Software offers the ability to write scripts for software to build features that are not supported by the software itself. Script language programs create multimedia presentations from a series of programming style commands linked together in a word processing type script.

The script language programs are powerful programs but can be difficult to learn. These programs are not readily transported to other hardware platforms.

3.3.3 Icon-based Programs

Although a multimedia project is a big task, makers of authoring software try to make it easy by using Graphical Icons as the control features of the authoring package program. The main features of Icon based programs are:

- i) Icon-based programs are easier to learn than scripting programs:
- ii) Editing individual elements in a presentation requires only clicking on its icon.

Icon-based authoring. programs are object oriented development programs where individual multimedia elements are represented by icons rather than script commands.

3.3.4 DLLs for Extending Features

In order to author a software, we talked of all multimedia devices becoming available. Scripts can be written to provide compatibility between hardware and software. Script programs are a powerful option in authoring needs to go beyond the software built-in features.

Advanced authoring programs can call on special features included in the windows Dynamic Linking Libraries (DLLs). Many add-on boards and peripheral devices include specialized DLLs upon installation. They can be called on for special functions that may not be available in the authoring software program. Specialized programming languages such as C++ allows users to create their own DLLs for enhanced functions.

3.3.5 Supporting CD-ROM or Laser Disc Sources

Storing arrays of audio and video on hardware is often not practical to get the quality and speed as might be desired. Authoring software allows full control of Laserdisc and CD-drives to integrate audio, video and computer files.

Authoring programs support direct control of CD-ROM drives, video and laserdisc sources. This is provided through the MCI command language.

Multimedia presentations that depend on large amounts of sound or video clips require excessive amounts of hard disk space. The presentation system can be configured by having a CD-ROM drive as the sound source and a laserdisc player as a video source. Both the pieces of hardware are controlled by the authoring software in a seamless manner.

3.3.6 Supporting Video for Windows

Video stored on hard disk is sometimes the right media for your project. Authoring software has the capability to support most multimedia elements like video for windows.

The latest version of Authorware Professional from Macromedia permits the loading of video clips from video for windows and animation clips from Autodesk.

3.3.7 Hypertext

Words can be made into buttons that will take you to cross-reference in a program. Hypertext capabilities can be used to link graphics, some animation and offcourse other text.

Hypertext programming interfaces allows users to select individual "Hot Words" on a screen which are then linked to additional pages and layers of text. The help system of windows is an example of hypertext.

Hypertext systems are useful when presenting and referencing large amounts of textual information. Some hypertext authoring programs provide graphical links for animation, sound and video as well as text.

3.3.8 Cross-Platform Capability

Authoring software can even package a multimedia project to play on other computer types with appropriate planning. This works successfully to broaden your potential audience.

An important consideration with authoring programs in their capability to be altered for presentations on Macintosh or other hardware platforms. Some authoring programs are available on several platforms and provide tools for transforming and converting files and programs from one to the other.

Care should be taken when considering cross-platform development because of significant performance differences in hardware peripherals and devices.

3.3.9 Run-time Player for Distribution

Run-time software is often included in authoring software to explain the distribution of your final product by packaging playback software with your content. Advanced authoring programs provide specialized packaging and run-time distribution for use with devices such as CD-ROMs.

Care must be taken when using run-time players for advanced authoring systems that all specialized DLLs, run-time modules for other programs and non-system fonts used in the presentation are packaged together when the final product is distributed.

3.4 AUTHORIZING TOOLS

Authoring can be described as creating highly interactive applications in which the information can flow in both the directions i.e. from application to user and from user to application. Multimedia authoring tools has helped in creating higher-quality audio and video applications with very little expertise.

Authoring does not require a team of programmers. Instructors, corporate communications, and content specialists with no previous programming experience should be able to learn and use the authoring environment.

At large, authoring programs could be grouped into the following two distinct types:

- i) Those that relied entirely on point and click, and
- ii) Those that required simple scripting.

Multimedia authoring tools can be evaluated by performing certain tests which include the creation of non-interactive multimedia, computer-based training (CBT) applications, catalog creation and even authoring for the world wide web. The advantage of using object-oriented programming has been inculcated in the authoring products like Everest Authoring System offers object instancing, while Media Verse and Oracle Media Objects allows to add new methods and events to existing objects. Supplied templates in Authorware and Media Verse saves the time in implementing specific functionality.

In order to create effective multimedia, one needs tools that can handle a wide range of authoring tasks, including interactive title development, CBT, interactive catalogs/kiosks and web authoring.

The CD-ROM's interactive title development task focuses on the flexibility and power of screen design tools and support for managing memory requirements and playback data rates. Also the ability to create cross-platform applications looked for.

Creation of Computer-based training (CBT) stresses wizards, templates, and other tools designed to make it easier for non programmers to generate applications. Other features included the ability to interface with a database either proprietary or via ODBC for tracking student performance, and special tools designed for creating exercises.

As the ultimate goal of interactive catalogs/kiosks is to sell things, so the products that offer robust database-handling tools would be best suited for processing sales transactions.

The Web authoring task involves generating an application that can be played by a helper application from within a web browser, or something as complex as a stand alone application designed to stream data from the server over TCP/IP connections and onto the user's browser without making the user download the application.

The following subsections discuss a few important authoring tools:

3.4.1 Authorware

Macromedia's Authorware is a dependable industrial-strength environment that allows multiple people to contribute to an application. Artists can load a library with media, programmers can provide templates for complex interactions, and interface designers can put everything together. And one non-programming designer can even do the entire job alone.

Authorware 3 allows authoring of programs in identical environments on either the PC or the Macintosh and produce runtimes for either. Authorware's icons hold a wealth of programming information that never requires the user to think like a programmer.

Some icons in Authorware can manipulate media on the screen. They can display and erase graphics, move or animate objects, play AVI and Quick Time digital movies as well as WAV sound files, and access videodisk and videotape players.

The new framework icon automates hypermedia interactions and provides a default navigation structure. With a 10-page document in rich text format (RTF), it is easy to import it into the first "page" of framework using Authorware. Authorware can create a new display page for each hard page break in a RTF (Rich Text Format) file. Eight buttons are used to move between pages, pop up a pick list of pages, search documents and exit the framework.

Authorware 3 has added text styles that you can apply to any text (labels, menu options, scrolling text boxes).

Designing in Authorware is also allowed. Authorware elegantly provides full data integrity for its naming of icons and variables. Change a variable name and it changes everywhere it's used.

Authorware can run timers, change the flow in response to a user, keep score, and access laserdisc players via dialog box.

Although Authorware includes ODBC to enable connectivity to and from databases, it falls short of enabling proper database interactivity and reduces each fetched record to a single text string.

Also, even though, Authorware ships with a full assortment of buttons, dialog boxes, sliders, gauges, scrolling text boxes, and data entry boxes, it lacks such features as the critical list and combo boxes that are used in the other programs to create the order screens.

Authorware rewards an open mind and a willingness to tackle its different approaches by giving non-programmers accessibility and allowing them to produce complex interactivity with a minimum of programmer like thinking. Authorware's built-in knowledge of interaction, testing and scoring makes it a mature and bullet proof environment for creating kiosks or CBTS.

3.4.2 Everest Authoring System

Everest Authoring System 1.5 is best suited for the developers of CBT applications who seek a powerful but easy-to-use windows-based authoring environment. As far as CBT environments go, Everest (a Windows 3.1 application that also plays back on Windows 95) features strong multimedia support, enough object orientation to improve the productivity over competing products, and great tech support.

Everest's nearly seamless multi-paradigmatic approach includes such features as visual programming with icons, direct manipulation of interface elements and procedural programming. Everest has successfully integrated these elements making applications development easier.

The Everest development environment opens with multiple views of an application. In Everest's application creation, the user starts off with a library and name an initial screen. Dragging a layout object icon from the toolset onto that screen provides the basis for the first screen. The layout object, visible as an icon on the Icon Script, also occupies the Attributes Window, in which the user can choose a background colour on bitmap, select a screen size, and specify upto 34 more properties.

Also interface and media object icons can be dragged onto the screen; like list and combo boxes, edit boxes, text displays, bitmap placeholders, sliders and gauges, buttons of all types, video and audio, OLE and animation.

These objects can be made to move around, visually resize them and their properties set by clicking on them in either the Visual-Screen or the Icon Script. The placeholder objects can be left empty or their contents can be specified immediately.

Everest's object orientation, which supports object instancing but not subclassing, increases developer's productivity. For example, if suppose we dragged a button object, dropped it in the Visual Screen and sized it for use as a Back control button to return to previous screens. In the Attributes window, we name it, set the bitmap used for pressed and released, and assigned it an "event code" - in this case, 33-that it would generate when pressed. And we did all this simply with drag-and-drop, point-and-click, and fill-in-the-blank. Then comes the fun part. In the Attributes Window, we set save as object to true. On the next screen, we dragged in another button off the Toolset. With one simple choice of a pull-down menu in the Attributes window, we made this new button an instance of the Back button that we saved as object, and the newly created button inherited all its properties.

Once the screen design satisfies the user, Everest has provided an easy method of scripting navigation and control flow.

Overall, Everest offers a charming environment in which to create CBT applications. Additional modules from third-party vendors can infuse Everest with data handling facilities to access databases and utilize rich text format. The base product can access only its proprietary database, which can read to and write text files.

3.4.3 Icon Author

Icon Author, an authoring tool requires the use of an icon-based flowchart for building an application. It still maintains a strict separation between an application's structure and the actual content and still ships with a number of ancillary programs that help to edit and manage content.

The role of Smart Object editor, has become central to the creation of Icon Author applications. Smart Object Editor assembles individual object such as imported pictures, sound files, video clips, animations, database links, text push buttons, tables and list boxes into coherent pages layouts. The appearance and behaviour of these objects can be controlled by changing the properties associated with each one.

Icon Author handles multimedia objects with aplomb. The sound, movie and animation files are not only easy to be imported but it also takes less effort to control them. For example, adding VCR start and stop buttons to a movie object requires just activating the control bar property in Smart Object Editor.

Creation of CBT application or authoring a CD/Kiosk presentation requires importing and manipulation of data. Icon Author has the ability to access databases via Microsoft Open Database

Connectivity (ODBC) drivers. Once the connection to a database is established externally (via the ODBC Administrator), the database object is simply inserted into a Smart Object Page.

3.4.4 ImageQ

ImageQ is another multimedia authoring software package. Although, it is a windows 3.1 application but it can also play back on windows 95, create slide show presentations well and provides a way to distribute them easily. It lacks the intuitive programming language and strong interface necessary for truly productive multimedia applications development.

ImageQ presentations comprise a series of background images and associated control scripts that add user interface controls, overlay images, and underlying programming to each slide. It uses a multiple-document interface to manage one or more presentations simultaneously.

The slide strip window for each presentation contains either thumb nails or file information for each slide image and provide the means of navigating through the presentation at design time. Although the images in the slide strip are, miniature versions of the background images they do not show any of the user interface element that the user has added such as button and list boxes. The user must create these through code, which ImageQ will interpret only at runtime-a major deficiency that other authoring system (such as Icon Author and Director) avoid by offering design tools to arrange interface controls visually without code.

While ImageQ does not provide the robust slide design tools that competing products include, it does import a wide variety of graphics formats.

Once the images have been imported into an ImageQ presentation, they can then kept in their native formats or saved as .BMP or .HKF files.

Whenever images are imported into an ImageQ presentation, it involves only creation of thumb nails and slide references, the images themselves remain as separate files. The list of filenames and the control and transition settings are stored in a file with an IMQ extension, which serves as the main file for the presentation.

ImageQ's programming language is awkward to work with, because it uses unconventional syntax and event handlers. For example, ImageQ uses lines like "add X to Y", put 150 into X for adding the two values in a single statement and assigning the value to 150 to X, respectively.

Though ImageQ lacks design and programming tools, it does well with distribution. It is easy to build portable presentations as stand alone .EXE files that includes the ImageQ runtime. ImageQ is really a presentation package in an authoring system's clothing. Its real strength lies in ability to create traditional slide shows with prebuilt images and distribute them with no royalty. Yet the product's lack of interactive design tools and its clumsy programming languages ultimately make it a poor choice for robust multimedia applications development.

3.4.5 Macromedia Director

Macromedia Director 4.04 is better suited to multimedia presentations and kiosk applications than CBT or interactive catalogs that require more database work. This product offers a lot:

- i) a central scoring component that provides precise timing control
- ii) cross-platform compatibility
- iii) strong yet intuitive animation features
- iv) and an extensive architecture to add functionality.

The application in Director's user interface is a movie, the playback screen is the stages, and the various elements in the movie are cast members. A Director movie consists of frames that comprise many separate, individual channels, each of which can hold a sprite or sound, custom palette, and the like. Each sprite is an instance of a cast member, such as a piece of text, a bit-mapped image, or a digital video clip.

As all of the sprites can be accessed during each frame of the movie, so an ultimate time-based control can be achieved.

Director has 48 separate sprite channels available in each frame. Sprites are only copies of the cast members so their look can be modified in a particular frame without affecting the original cast member. Each channel number represents a stage layer so the sprites can be moved between layers by simply suffering them up and down. Some special channels exist to refer to each frame; transition, palette, tempo, script and two sound channels.

One of Director's strongest features is its cross-platform compatibility. The Windows and Macintosh versions of the products are binary compatible so the Director movies can be transferred between them by simply moving the file.

Director imports an impressively broad range of graphics formats. When a file is imported, it automatically becomes a cast member and receives a number. Version 4.04 has the ability to mix sound files sampled at different rates and sizes but it is not possible to mix AIF and WAV files with the audio in digital movie clips.

Animation features which is one of Director's strongest suit allows to set the sprites in motion differently.

Director lets the non-programmer arrange cast members frequently and put them in motion, but robust movies need some coding. (Director's language, Lingo, uses English like phrases for commands, but getting used to it takes some time). Director has been proved to be very useful for implementing linear presentations but weak for complex programming.

Though Director can create complex multimedia events such as CBT, a bit of programming experience is still required to pull them off. For time line-based interactive titles and demo CDs, however, Director is an exceptionally useful tool that exploits a logical metaphor to achieve spectacular results.

3.5 QUICK TIME

Quicktime is a software, developed by Apple, that supports time-based media on the Macintosh. It has also been ported for windows environment. An example of time based media is video-a sequence of images that are displayed on screen to create the illusion of motion. The images are time based because to retain synchronization with a sound track, the images must be displayed at the correct time in relation to the sound track.

The Figure-1 show how the different development environments relate to one another, to Quicktime and to programming an application form scratch.

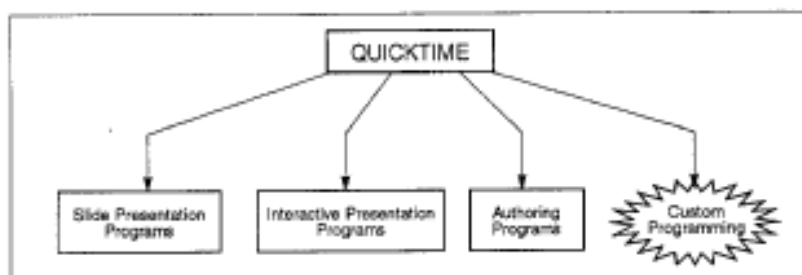


Figure 1: Relationship of QuickTime with different multimedia developments environments

It is probably noticeable from the figure that the QuickTime is off on its own with a circle pointing to each of the other areas. That's because QuickTime is not really a development environment like the others. It more closely resembles a data type, just as PICT is a data type for graphics, and AIFF is a data type for sound.

Though QuickTime can be used to create movies and may even replace the slide presentation programs for some job, it is far more likely that QuickTime will be used to provide animation or video from within one of the development environment.

Importance of QuickTime

QuickTime adds support for time-based media to the basic Macintosh system architecture. The most common media that require this kind of support are video, animation and sound. Sound has been supported on the Macintosh since its introduction, but it's important that QuickTime supports sound because QuickTime can synchronize sound tracks with video tracks.

QuickTime provides a synchronization between the animations that would otherwise play at different speeds when played on different machines.

QuickTime stores video digitally-that is, it converts the frames of video to digital images that are stored on the computer's hard disk. When the video is "played" these images are read from disk and displayed on the computer screen. The advantage of storing the video this way is that the computer can very quickly and easily access other information on a hard disk.

Another advantage is that once the frames are stored on the hard disk they can be manipulated using a number of different editing applications.

The disadvantage of storing video this way is that the computer has to do a lot of work to display the video sequences.

How does QuickTime Work?

To achieve digital video, the computer needs two things:

- 1) The computer must be capable of handling video's large amounts of data. This is accomplished by streaming in which the computer displays one frame as it reads the following frame from the disk.
- 2) The computer needs some kind of timing mechanism that keeps video and sound synchronized. Timing is particularly important when the people are displayed

speaking on the screen as without synchronization their lips won't move in synch with the sound. QuickTime fulfills both needs. QuickTime has many features but the two most important are its Compressors and its timing mechanism. The compressors enable the program to reduce the frames to a manageable size, so the computer can stream the information from the disk. The timing mechanism ensures that multiple tracks maintain synchronization.

Without compression, the large amount of data would prevent even the fastest hard disk from playing a movie-by the time the computer had read the first from the disk, the same for displaying the frame would have passed. Even with compression, QuickTime may not be capable of playing all the frames in the movie. QuickTime handles this problem by dropping frames. A 20-frame movie might play at only 10 frames per second on a LC, in which case QuickTime plays every second frames in the movie. QuickTime always tries to maintain the quality of the sound.

3.6 HYPERTEXT

Although the term "hypertext" is relatively new to many of us, it was actually coined in 1965 and an article predicting hypertext was published in 1945.

Hypertext is nonlinear or nonsequential text; there is no single order that determines the sequence in which the text is to be read. That is, the text is organized so that one can easily jump around from topic to topic.

Although, hypertext is best brought to life on a computer, it can be found in simple paper documents too. A paperback novel and the front page of a newspaper are the two examples of text out of which the newspaper page is hypertext because one can easily jump from topic to topic if one wants more depth on any one topic. While, the novel, on the flip side, does not encourage to skip around. The novelist assures he will be reading in a fixed sequence reading one page right after another. Figure-2 illustrates hypertext with an example.

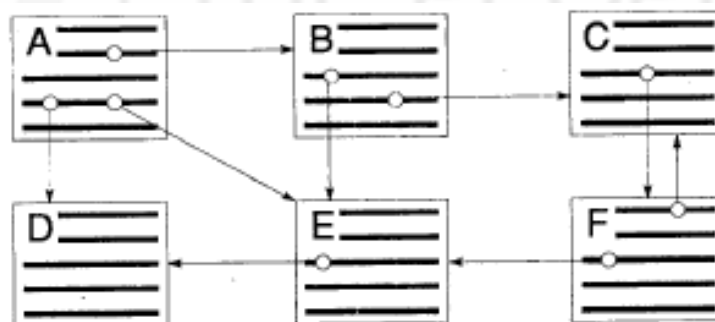


Figure 2: Hypertext Structure

Assumes that one starts reading the piece of text marked A. Instead of a single next place to go, this hypertext structure has three options for the reader: Go to B, D or E.

Assuming that one decides to go to B, then to C or to E, C and from E, one can go to D. Since it is also possible to go directly from A to D, this example shows that these may be several different paths that connect two elements in a hypertext structure.

Hypertext presents several different options to the readers, and the individual reader determines which of them to follow at the time of reading the text. This means that the author of the text has set up a number of alternatives for readers to explore rather than a single stream of information.

The same is true of footnotes in traditional printed texts, since readers have to determine upon reaching the footnote marker whether to continue reading the primary stream of text or to branch off to pursue the footnote. Therefore, hypertext is sometimes called the "generalized footnote".

When hypertext is implemented on a computer system there is potential for giving readers more flexibility in jumping around from topic to topic. The text for each topic may be stored in a special area of computer memory and then easily accessed. In the hypertext literature, these "areas of computer memory" are referred to as "notecards", "cards" or "nodes". Whatever be the size of these nodes, each of them may have pointers to other units, and these pointers are called links. The number of links is normally not fixed in advance but will depend upon the content of each node. Some nodes are related to many others and will therefore have many links, but have no outgoing links of their own. Sometimes, such nodes without further links are called leaf nodes.

Figure 1 also shows that the entire hypertext structure forms a network of nodes and links. Readers move about this network in an activity that is often referred to as browsing or navigating rather than just "reading" to emphasize that users must actively determine the order in which they read the nodes.

A hypertext link connects two nodes and is normally directed in the sense that it points from one node (called the anchor node) to another (called the destination node).

In the early days of hypertext, nodes contained only textual data. Now nodes can contain various kinds of data: graphics, audio, video, computer-animated images, film clips of animated scenes, digital sound or other kind of information. The term hypermedia may be used, when referring to systems that include a wide variety of node types.

Hypertext is now being used in more and more areas. Educators are making good use of hypertext and making learning difficult subjects more and more accessible. While reading some instructional material on a computer screen, the user may come across several technical terms on the screen that are highlighted (shown in a different colour, say, yellow letters on a green background or black letters on a white background). One can instantly branch off to a new window of information a definition or elaboration of any of these highlighted words. This enables a comfortable learning with a well-designed hypertext system.

3.7 APPLICATIONS OF HYPERTEXT

This section highlights some of the current applications of hypertext.

3.7.1 Computer Applications

Since hypertext is a computer medium, it is natural to use it in connection with computer oriented applications. Hypertext can be used to prototype the user interface for almost any other computer program as most initial prototyping consists of linking together screen designs and presenting them to the user in an order determined by simple user actions.

a) Online documentation

Online documentation may be the most natural of all hypertext applications. Hypertext is the obvious method for helping users in the situation when they require

good access tools to help them retrieve the sections of the manual that are relevant to their current needs.

b) User Assistance

Hypertext provides a mechanism for integrating several forms of user assistance, including the online manual, an introductory tutorial, an online help system, and even the error messages. Because users only get error messages when they are in some kind of trouble, the error messages are a prime candidate for providing users with assistance.

In an integrated user assistance facility based on hypertext, it would be possible for the user to link from an error message to the location in the help system that gives further assistance on the problem. Suppose if the user's difficulty is not the error situation in general but a single incomprehensible word in the online message, it may become possible to link from that word to the location in the online manual where it is defined with the help of hypertext. Also if the user wanted further assistance than could be provided by the help system or the manual then it may become possible to link further, to the appropriate location in the tutorial component, to get a computer-aided instruction lesson.

This type of integrated user interface does not exist in current computer systems and can be made possible with the hypertext facility.

c) Software Engineering

The hypertext has a great potential for providing links among the large number of specification and implementation documents that are produced during the software development life cycle. For example, it would be possible to start from a requirements document and link to that part of the design specification that meets a given requirement. One could then link from the design specification to the actual code to see how that design is implemented.

A development organization would require to follow a software engineering methodology supported by an integrated set of computerized tools in a complete CASE (Computer Aided Software Engineering) environment, in order to derive full benefits from this form of hypertext links among the various documents in the software lifecycle.

It is also possible to use hypertext in a less lifecycle-oriented approach by including linking facilities in structure-oriented editors for program code. For example, it is possible to click on a variable to get to see its definition and associated comments, or to link from a procedure call to opening a window with the text of the procedure. The Smalltalk code browser links related pieces of code in somewhat this manner.

3.7.2 Business Applications

This section concentrates on "mainstream business", but quite frankly there are not yet all that many business applications of hypertext in place to be used every day for real profit. But there are potential applications that are currently being investigated by several companies, and there are also some systems in real use.

a) Dictionaries and Reference Books

Several dictionaries and large reference books have been converted from a traditional paper form to a hypertext format. A hypertext dictionary from scratch has been generated for the children's dictionary shown in Figure-3.



Figure 3: Children's dictionary with hypertext features

For pure environment the user can click on the picture of the dog to have it replaced by another kind of dog. There are hypertext links to "dog words" and to other pets, as well as the links to other words from the definition. All these features encourage the child to explore the dictionary and learn new words.

One of the biggest advantages of hypertext encyclopedias and dictionaries, compared to a printed work, is that they can show moving images and play sound. For example, Microsoft Bookshelf contains a dictionary with sound recordings of how the words are actually pronounced.

Printed dictionaries for deaf users portray hand and facial gestures through line drawings with arrows to show movement, so multimedia dictionaries are clearly a superior format for communicating this dynamic language. Also, the computer format has added benefits such as the ability for less experienced users to view a sign at much slower speeds than normal conversation.

b) Law

Hypertext has two main applications in the legal field. The first is to support legal research and is mainly used by professional lawyers. The second is support of legal document creation and management and is used by both lawyers and people who wish to avoid paying legal fees.

Legal texts are filled with cross-references and are therefore well suited for hypertext support. Typically, a lawyer's brief might refer to several court rulings, each of which would refer to further rulings and to the applicable statutes. Having each of these references available online saves the lawyer much time in researching the law. A special aspect of law is that new statutes or higher court ruling may invalidate earlier decisions, meaning that something may be legal one day but illegal the next.

This time varying nature of the law has two consequences for legal hypertext systems. The first is that a lawyer who accesses one of these overruled cases would better be informed that it is no longer a valid law. Legal information services achieve this purpose by inserting bi-directional links from the new law to any earlier nodes referring to the same issue. Lawyers can then follow the links in the opposite direction in a process known as Shepardizing after the printed books that originally supplied

these inverse links. The second implication is that lawyers will sometimes need temporal scrolling of the law to discover what the rules were at some earlier point in time when the action under consideration took place. For example, if a client is taken to court over an old income tax return, the defense should be based on the tax rules for the year in question.

Another legal application is the use of multimedia visualizations by trial lawyers to illustrate their version of events during a trial. Since visualizations are so compelling, juries can often be convinced that something has really happened because they saw it happen (on the computer).

c) Trade Shows, Product Catalogs and Advertising

Many kinds of advertising and communication to customers can be improved by hypertext. For example, one can attract attention at trade shows by having a computer with hypertext information about one's products.

Hypertext can also be used to provide information about an entire trade show and help people find those exhibitors that would interest them.

One of the intrinsic advantages of hypertext in an advertising context is the general ability of hypertext to provide access to large amounts of information but to show the user only those small parts that interest him or her. This property of hypertext is important for applications like product catalogs. A hypertext product catalog can reduce the complexity of choosing among a large number of options by showing only those that are relevant for the individual customer. It can also offer help in placing the order and might even place it by an online connection to the vendor. A hypertext catalog could also include an option to remember what products the user ordered the previous time, thus making them especially easy to reorder.

Traditional types of advertising are certainly possible in hypertext, either by sending customers a disk or by making promotional materials available on the internet. Compared with paper publishing, where a majority of the pages in many newspapers consist of advertising, hypertext provides great potential for saving resources while making more advertising content available to those readers who express an interest in a certain ad and follow its link to the supplementary materials.

3.7.3 Educational Applications

Many hypertext systems have been produced specifically for educational use. Hypertext is well suited for open learning applications where, the student is allowed freedom of action and encouraged to take the initiative. For example, the Interactive NOVA hypertext allows the student to browse through a set of biology information and see those parts that interest the student or make sense in the context of a current assignment. On the other hand, hypertext may be less well suited for the drill-and-practice type learning that is still necessary in some situations.

Foreign Languages

The linking abilities of hypertext are ideal for the learning of foreign languages. Hypertext can provide automatic access to dictionaries through implicit links from any text. A student who does not know English very well might still be able to understand material in Intermedia because of its ability to link to an explanation of any word. Hypertext also enables students to view two parallel versions of the same text: An original version and a translation.

The Video Linguist is a hypermedia system that teaches a language by showing clips of television broadcasts from a country speaking that language. The advantage of this approach are that TV shows are fun and motivating and that they reach the culture of the country in addition to the language. One major problem with learning languages is that native speakers of many languages tend to speak in a very agitated and fast manner which makes their utterances hard to understand for foreigners. But of course the goal of learning French is to understand the way the language is actually spoken in France and not the way a teacher may speak it in class. Therefore the Video Linguist initially plays the original sound track from the broadcast version of the show.

If there is a part of sound track that students do not understand they can utilize the hypertext facilities and link to a version where the same words are spoken more slowly. If they still don't understand it, they can follow a link to a version where each word is spoken v-e-r-y s-l-o-w-l-y and clearly. A final hypertext feature in the Video, Linguist allows the student to click on any word in the subtitles and get its dictionary definition.

3.7.4 Entertainment and Leisure Applications

Hypertext provides several opportunities for pure enjoyment. Although, there has been very little research conducted in this area, but still there has been some pioneering research as well as the first few commercial examples.

News, Newspapers, and Magazines

News delivery is an obvious candidate for online services since events can be reported as they occur instead of having to wait for the scheduled publication time of a newspaper or magazine. Online versions of large newspapers help the articles to be accessed around the clock. Some added value derives from search capabilities where one can find, for example, all news stories about a certain company over the last month before deciding whether to invest in it or how to solicit it as a customer. Also information filtering can be used to design personalized electronic newspapers with exactly those newsstories that are of interest to the individual reader.

The ability to link between articles from several magazines is a value-added feature of the hypertext versions of the magazines. Even the discussions between readers is a value-added feature for the hypermedia versions of the magazines: if somebody is reading, say, an article about an upgrade to a spreadsheet application. that person is likely to be interested in the experience other readers have had when installing the upgrade.

3.8 ELEMENTS OF HYPERTEXT

This sections gives a quick overview of the basic elements of hypertext which are as follows:

- 1) Nodes
- 2) Links
- 3) Annotations
- 4) Buttons
- 5) Editors
- 6) Browsers
- 7) Trails

Built-in programming languages

1) Nodes

Hypertext is a network of nodes. A node is a collection of data organized around a specific topic and related or linked to another body of information. In a network, each node is linked to some other node.

In different hypertext systems, different terms are used for nodes. In HyperCard, a card is a node. In HyperPAD, the node is a Pad.

In many hypertext systems, a node is a screenful of data. Nodes can be categorized in different ways and there can be different kinds of nodes.

The Figure-4 illustrates the node reference within the screens.

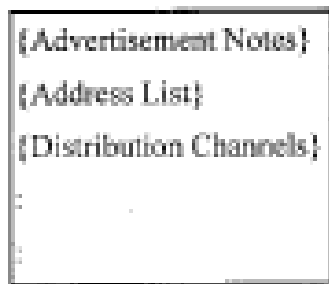


Figure 4A

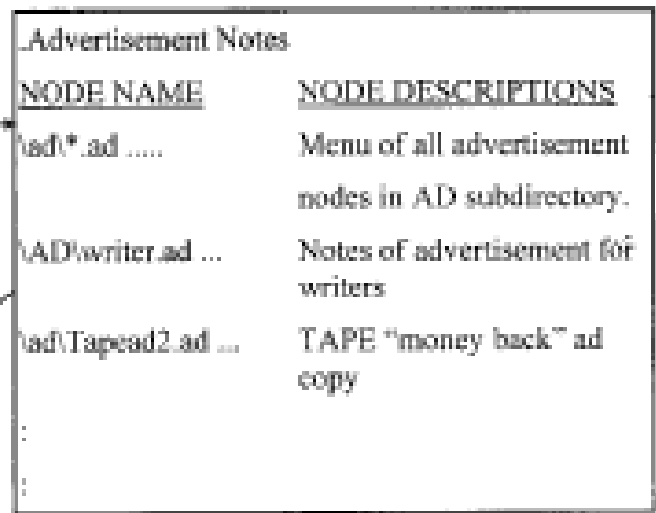


Figure 4B

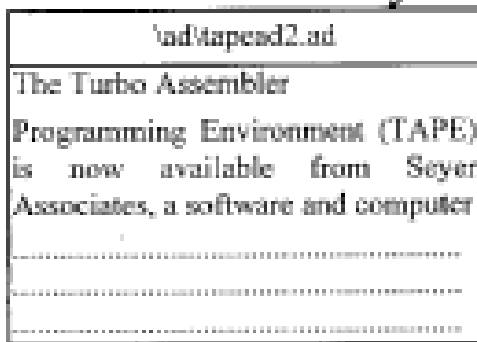


Figure 4C

Figure 4: Network of Nodes

This first node reference is {Advertisement Notes} in Figure 4A. It is called the top-level control text because it contains only references and pointers to other node and it is always enclosed in braces. That is, each item in braces is a pointer to another node. By moving the cursor to an opening brace and pressing F10, the user can instantly branch to the corresponding node in the network as shown in Figure 4B. The second level node in this figure also contains reference to other nodes rather than the content information. So, in this case, it is also a control node-it only allows the users to control their navigation through the network.

Placing the cursor on a backslash and pressing F10 instantly displays the contents of that node as shown in Figure 4C. Here each node is shown as a separate text file.

2) Links

Links are the other fundamental unit of hypertext besides nodes. A link is information embedded in a node that connects it in some way to another node. Links are almost always anchored at their departure point to provide the user with some explicit object

to activate in order to follow the link. The result of activating the anchor is to follow the link to its destination node.

Most links are explicit in the sense that they have been defined by somebody as connecting the departure node with the destination node. Some systems also provide implicit links which are not defined as such but follow from various properties of the information.

A hypertext link has two ends. Even if a link is not bidirectional, there may still be a need to anchor it explicitly at the destination node. Hypertext systems also have plain links, which are just connections between nodes. The advantage of that approach is of course the simplicity of both authoring and reading. There is nothing to do with links excepts to follow them, and that one action can be achieved by a click of the mouse.

In addition to the standard links connecting two nodes, some hypertext systems also have "super- links" to connect a larger number of nodes. There are several possibilities for dealing with having a single anchor connected to several destinations. The two simplest options are either to show a menu of the links or go to all the destinations at the same time. Intermedia uses the menu option and allow users to choose only a single destination. This approach requires good names for the links or destination nodes in order for users to be able to understand their options. Some users of NoteCards have implemented a "fat link" type that opens windows on the screen for all the destination nodes.

3) Annotations

A special link type is the annotation link to a small, additional amount of information. The reading of an annotation typically takes the form of a temporary excursion from the primary material to which the reader returns after having finished with the annotation. Annotations are quite similar to footnotes in traditional text and can be implemented, as Guide pop-up windows that disappear as soon as the user releases the mouse button. Annotations can be accessed through an icon.

Hypertext writers can use annotations in the same way they would use footnotes in traditional text with the exception that hypertext annotations are less intrusive because they are not shown unless the readers asks for them. Many hypertext systems allow readers to add new links to the primary material even if they do not always allow the reader to change the original nodes.

4) Button

A button is a visual cue in a node that alerts a user that a link exists. In other words, a button is a visual representation of a link in a node. By pressing a mouse button, or appropriate function key, the user can cause the system to activate a link to display a different node. Some literature describes a button as a "hot spot" on the screen-a spot that is sensitive. If the cursor (or pointer) is moved over a "hot spot", it will change shape indicating the type of button present.

The buttons in BlackMagic (a hypertext word processor) appear in different colors on screen. There are different kinds of buttons and links. The different buttons are labelled in the annotated screen shown in Figure 5.

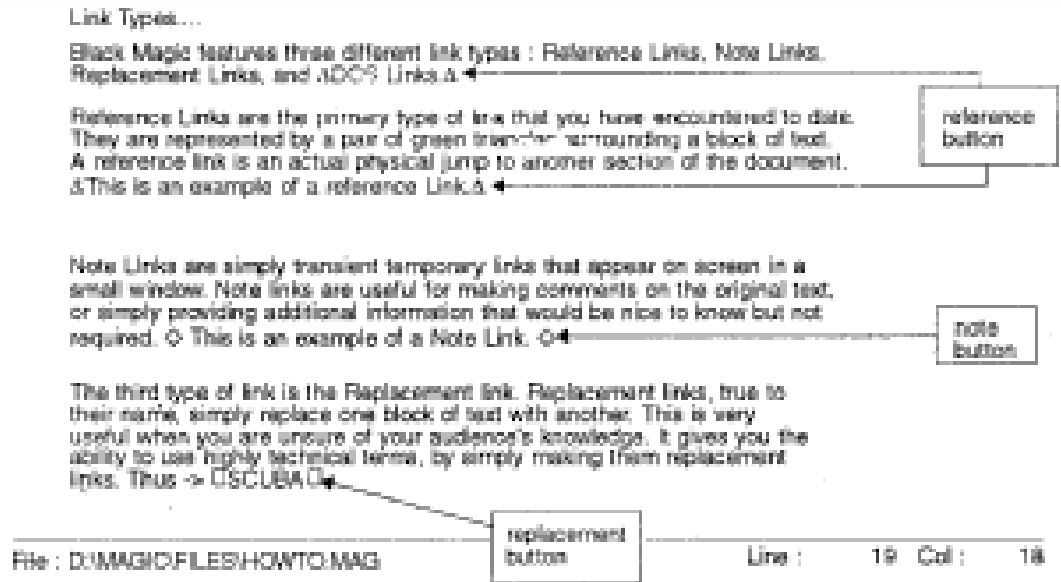


Figure 5: BlackMagic screen with note button

In this discussion, the button includes the diamonds and the text in between the diamonds. When you move the cursor is moved on to a diamond (or on to a text delimited by the diamonds) and F1 is a note node appears in a small window in the upper right corner of the screen as shown in the Figure 6.

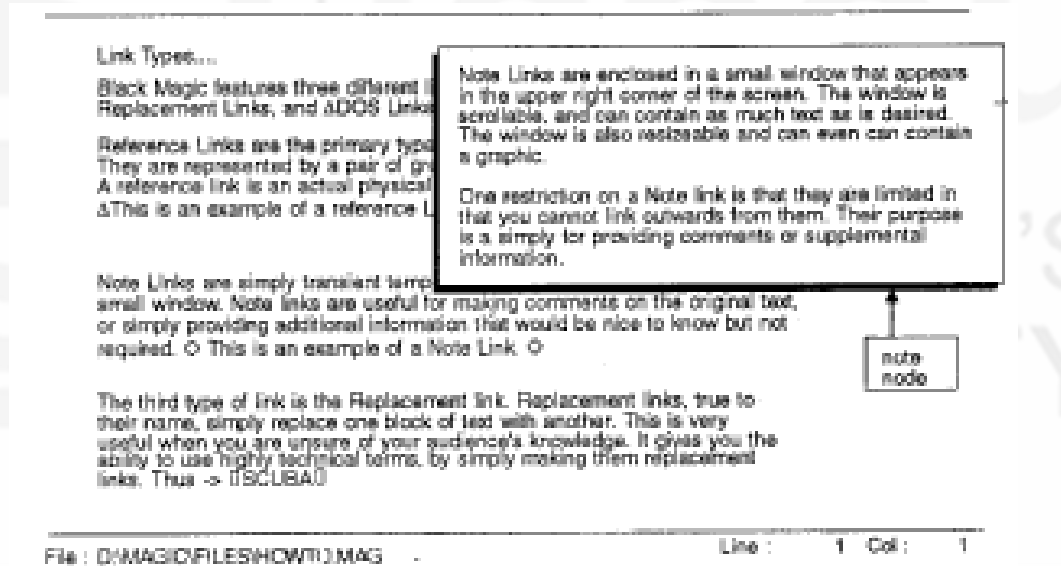


Figure 6: BlackMagic screen with note node

5) Editors

An editor is the part of the hypertext systems that enables a user to create a node and link it into the network. Some hypertext system are net distributed with the editors because they are intended as "presentation-only systems".

6) Browsers

A hypertext browser is a program or subprogram that can display a diagram of a network of nodes. Few hypertext systems for personal computers currently have browsers.

A hierarchical browser has some nodes that are regarded as "parent" nodes which are at a higher level than nodes beneath them. They are sometimes regarded as "child" nodes.

In a hierarchical browser, each of the child nodes are linked to each parent node. That is, one can see what child nodes spring from any given node. If the same child node is connected to two different parent nodes, the child node is repeated under each parent node. In a hypertext network, however, there may not be a hierarchical relationship among nodes. Nodes may be equal and just linked together. In such a case a network browser might be more appropriate. In a network browser each node would be designated only once and lines would be drawn to show all of the links between all of the nodes.

7) Trails

A trail is a record of the nodes that a user has accessed in viewing a hypertext network. Ideally the trail would include "note nodes" created by the user in the process. Some hypothesize that users will blaze exciting, useful or enjoyable trails for others. It might be enjoyable for example, to follow a trail made by others (famous or otherwise) and read their comments on various stories, illustrations, new items and so on.

Trail recording can be a useful tool for refining hypertext networks. If one can study the trails that users take in searching for information or in studying an instructional network, he can spot problems and refine the network.

8) Built-in programming languages

Some hypertext systems have built-in programming languages, so one can customize the system to fit one's specific needs.

A good example of a program that has a built-in programming language is KnowledgePro. KnowledgePro enables to build expert system, systems that can emulate the behavior of human experts within a limited subject area. KnowledgePro allows store a number of rules in a file called a knowledge base using special Knowledge-Pro commands. One can program KnowledgePro to present information and ask questions to help users solve complex problems. KnowledgePro differs from many expert systems in that any word in question or response can serve as a hypertext button. One can program KnowledgePro so that when the user selects a hypertext button, a built-in subprogram is activated. The subprogram may go off on its own tangent-ask questions, branch to other subprograms and so on. Eventually, though, when the user finishes the subprogram, control will return to the node that contained the hypertext button.

Several other hypertext systems have built-in programming languages. HyperCard on the Macintosh is a language called HyperTalk.

IBM LinkWay also has its own script language, as does HyperPAD. With such a language one can do more than just create a network of nodes and create special visual effects, prompt the user for specific input, and process the input-for example, perform mathematical calculations.

Languages like HyperTalk and PADtalk, which works with HyperPAD, are not computer languages in a traditional sense; they are scripts that the hypertext application program reads, interprets and then executes. Still, they qualify as programming languages because they enable you to store a series of instructions that can be activated at the touch of a button.

3.9 REVIEW QUESTIONS

- 1) Which authoring tools is most suitable for online help system on a computer?
- 2) Which multimedia software is best suited for synchronizing sound tracks with video tracks?
- 3) Explain any one business application (other than that specified in the unit) which is an example of applications of Hypertext.

3.10 SUMMARY

In this unit, you have learnt the need for multimedia authoring. (why it is required), features of the authoring tools:

- Authorware
- Everest Authoring System
- Icon Author
- ImageQ
- Macromedia Director

You have seen how is QuickTime related to these relating programs. In addition, you have learnt the working of QuickTime.

By now you have a good background of the fundamentals of hypertext and why is it valuable. In addition, you've learned about typed links and how you might use them for more intelligent processing of hypertext networks. It has also explained about the various applications areas to which the hypertext can be used.

3.11 ANSWERS/SOLUTIONS

- 1) Hypertext
- 2) QuickTime
- 3) Auditing is another natural application for hypertext because it is based on relating information from various sources and checking for consistency. The audit task includes gathering and producing large numbers of documents and linking them together to substantiate the accuracy of the information they contain and huge amount of information get distilled into a single financial statement, so links are needed between the conclusions and the source data. Further more, the audit of an international company involves a large audit team distributed over several countries, leading to several advantages for various forms of computer support like electronic mail and hypertext links among documents produced in different areas of the world.

The hypertext system links the information produced during the audit process and makes it possible to track information from a final financial statement back to where it originated. It is possible to scan original documents from the clients, and it might conceivably the possible in the future to link directly into the client's own computer system.