Hypermedia-based Requirements Engineering

Peter Aiken
Department of Information Systems and Systems Engineering
School of Information Technology and Engineering
George Mason University
Fairfax, Virginia 22030
703/764-6233

Chapter for ***Advanced Technology for Command and Control Systems Engineering***
(Stephen J. Andriole, editor), AFCEA International Press 1990.

Chapter Overview

CHAPTER OVERVIEW

This chapter outlines a hypermedia-based approach to the requirements engineering portion of software development life cycle. While as yet untested, hypermedia-based tools and techniques seem to offer promising benefits to the analysis and design phases of certain classes of software system development. The first half of the chapter is devoted to an examination of the development of hypermedia as a technology. While its conceptual origin dates to 1945, only recent efforts by software engineers have resulted in tools that begin to approach the functionality of concepts put forth by researchers in the field. These concepts are briefly reviewed. A wide range of software packages have what are labeled as hypermedia capabilities. The next section presents definitions intended to cut through the hype surrounding the evolution of this new technology. The following section presents an examination of the actual capabilities of hypermedia-based systems. These have been summarized as eight features common to systems with hypermedia capabilities. The second half of the chapter is devoted to a discussion of the application of hypermedia features to requirements engineering functions. While not applicable, to all cases each of the eight features can be implemented in a fashion augmenting the development of certain classes of software applications. Specific classes of systems benefiting from the application of hypermedia-based requirements engineering are identified. The requirements engineering process is briefly described in terms of four basic functions: capture, organize, structure, and present. Hypermedia support for each function is outlined. The chapter closes with a brief description of a workstation constructed from off-the-shelf components capable of supporting hypermedia-based requirements engineering.

HYPERMEDIA: AN EMERGING TECHNOLOGY

The Conceptual Origins Of Hypermedia

The conceptual basis for the hypermedia workstation is largely influenced by the work of three researchers. This section describes the contributions of Vannevar Bush, Ted Nelson, and Douglas Engelbart who collectively supplied concepts contributing to the rudiments of hypermedia technology. Three concepts, associative indexing, hypertext, and augmentation technology, initiated a chain of research culminating in current interest in hypermedia technology.

The Importance of "Selection" in Information Retrieval

Hugh G. J. Aitken has said, "Scientists discover new knowledge ... largely in response to problems posed by scientists, and the new knowledge so generated is consumed largely by scientists themselves" [Aitk76, p. 315]. In the July 1945 issue of the *Atlantic Monthly*, Vannevar Bush posed one such problem described at the time as having the potential to alter the "relationship between thinking man and the sum of our knowledge." As Director of the Office of Scientific Research and Development, Bush was concerned with setting a post-war research agenda for the thousands of scientists who were currently applying science to warfare. The area in which he saw the most pressing need was support for the investigator who "is staggered by the findings and conclusions of thousands of other workers - conclusions which he cannot find time to grasp, much less remember, as they appear" [Bush45, p. 101].

In "As We May Think," Bush noted the inadequacy of current means of transmitting and disseminating research results because the published record exceeded the ability of an individual to make use of it. In order to be useful, a scientific record must be consultable. One means of making records more accessible was to reduce the cost of information storage and distribution. Data compression was Bush's solution and, extrapolating current trends, he postulated a time when the cost of reproducing the *Encyclopedia Britannica* on microfiche would be a nickel. Further, the fiche version could be mailed anywhere for the cost of a first class postage stamp.

In Bush's vision of the future, scientists would use an integrated recording system, combining photography and voice recording techniques to create their records. He wrote, "One can picture a future investigator in his laboratory. His hands are free, and he is not anchored. As he moves about and observes, he photographs and comments. Time is automatically recorded to tie the two records together. If he goes into the field, he may be connected by radio to his recorder. As he ponders over his notes in the evening, he again talks his comments into the record" [Bush45, p. 104].

After describing the means of coping with the increasing rate of growth of the scientific record, Bush turned his thoughts to the process of utilizing the information. He identified the "prime action of use" for information as **selection**. Then current means of automating the selection process involved examining every potential piece of information and selecting those meeting the specified criteria. Bush wanted a system more like the telephone exchange, capable of making direct connections between circuits without employing search mechanisms. He criticized the artificiality of indexing systems as being contrary to the way the human mind works.

> "It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain. It has other characteristics, of course; trails that are not frequently followed are prone to fade, items are not fully permanent, memory is transitory. Yet the speed of action, the intricacy of trails, the detail of mental pictures, is awe-inspiring beyond all else in nature." [Bush45, p. 106]

Noting the futility of attempting to artificially duplicate this process, Bush suggested a *memex* as an alternate implementation vehicle. He described memex as a private mechanized system for accessing and filing communications and records. Printed material would be distributed on microfilm, accessible through a reader. Traditional index-based means of accessing the material would be available. More importantly, memex systems would also support automated associative indexing. Users would be able to create unobtrusive buttons in and around the data being viewed. The buttons would represent associative links to other information, which could be accessed by *tapping a button*. Creation of buttons would be as natural as turning the pages of a book. While the human brain is faster and more flexible, the associative indices created and stored in memex would not fade as a function of the passage of time and / or frequency of access,. In addition, humans would benefit from significant increases in the clarity of information items recalled by using the system. Bush predicted more efficient and effective scientific progress using memex and other systems supporting this associative linking process. One of the primary motivations behind hypermedia-based software engineering research is the belief that the ability to create these associative indices will positively effect requirements engineering efforts.

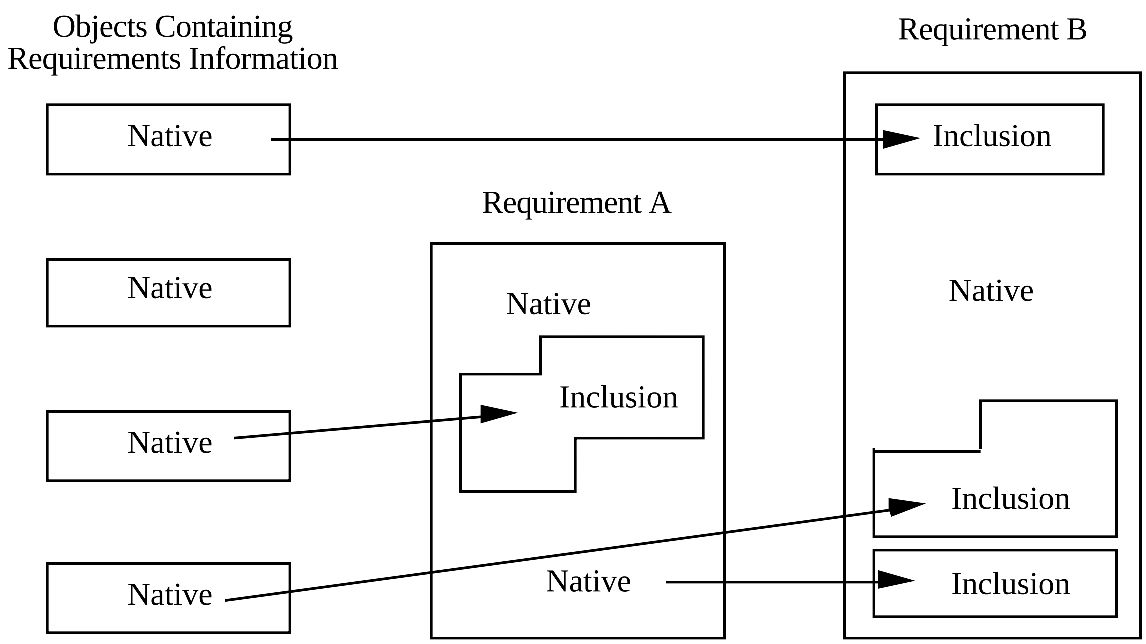
Hypertext

In the fall of 1960, Ted Nelson was a graduate student at Harvard taking a course on computers in the social sciences. He allegedly was astounded that automated support for organizing notes did not exist in forms that he considered useful. This useful form evolved into what he termed *hypertext*, meaning non-sequential reading and writing [Conk87]. Since the 1960's, Ted Nelson has been advocating use of a hypertext-based approach to creating and preserving knowledge and with this purpose in mind, developed Project Xanadu.

Documents in hypertext systems consist of *original text*, created specifically for the document in question and *inclusions*, information originally created or included in other documents but now part of a new document. (Figure 1 - Hypertext Document Creation illustrates the use of hypertext to create system requirements.) Based on this model, Nelson's original concept has evolved into a grand vision for hypertext as "a repository publishing network for anybody's documents and contents, which users may combine and link to freely" [Casa88, p. 24]. To fulfill his vision, Nelson has also created

- methods for keeping track of the immense quantity of links generated by authors as they create and include text objects, and
- a document numbering system suitable for tracking the very large and growing collection of documents existing on distributed systems and for tracking the origin of all inclusions [Nels88, p. 225].

Figure 1 - Hypertext Document Creation



(Adapted from [Casa88])

Nelson's vision of hypertext has had a significant impact on the development of hypermedia technology. As hypertext has evolved into the primary vehicle for implementing the associative indices described by Bush, his status, originally that of a maverick, has changed to that of a visionary. Nelson has stated that products based on *Xanadu* concepts and running on Sun workstations should ship during 1989 [Casa88].

Augmentation Technology

A 1962 summary of Doug Engelbart's research to date, *Augmenting Human Intellect: A Conceptual Framework*, completed the conceptual basis for hypermedia technology. Inspired by J. C. R. Licklider's concept of human-computer symbiosis [Lick60] and by the concepts developed by Bush and Nelson, Engelbart saw a potential usefulness for man-computer synergism [Casa88]. He defined augmentation technology as interactive computerized methods or devices capable of extending native human sensory, mental, and motor capabilities [Enge62].

In 1968, Engelbart implemented some of his ideas in a system called NLS (oN Line System) which has since evolved into its present form, Augment, the information repository for ARPAnet [Casa88]. NLS was the first implementation of hypertext and was quickly extended to include a number of other features such as integrated computer-to-computer communications and conferencing facilities, electronic mail and other support for group collaborative processes, integrated text and video imaging, the mouse, and the first windowing system, and the first context-dependent Help system. Engelbart collectively referred to these efforts aimed at providing consistent support for knowledge workers as *augmentation technology*.

> "By augmenting man's intellect we mean increasing the capability of a man to approach a complex problem situation, gain comprehension to suit his personal needs, and to derive solutions to problems. Increased capacity in this respect is taken to mean a mixture of the following: that comprehension can be gained more quickly; that better comprehension can be gained; that a useful degree of comprehension can be gained where previously the situation was too complex; that solutions can be produced more quickly; that better solutions can be produced; that solutions can be found where previously the human could find none." [Engl62]

Augmentation technology helped to define the thrust of modern support-based technology. In general terms it defines the nature of support required by the knowledge worker [Engl62]. Hypermedia support for requirements engineering is a direct extension of augmentation technology. Functions supported by the hypermedia workstation are memex-like functions tailored to provide support for requirements engineering tasks. Much of that support is focused on providing support for the requirements engineer's ability to create associative indices while working with requirements information.

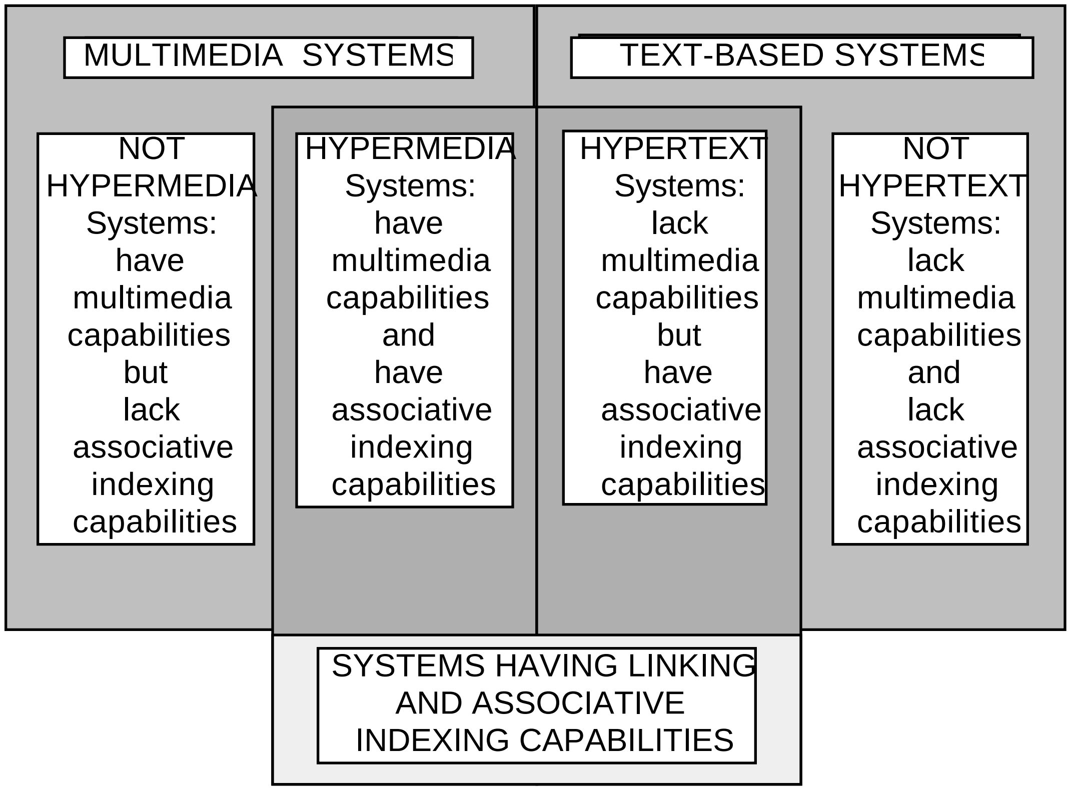
Hypermedia Defined and Differentiated

Currently, the literature presents a bewildering number of often conflicting definitions for the terms "hypermedia," "hypertext," and "multimedia." (For examples see [Nels67], [Yank87], [Deli86], [Garr86], [Conk87], [Garg87], [Aksc88], [Casa88], and [Smit88].) This section presents a set of definitions in hopes of clarifying the terms used in this chapter. Figure 2 presents one means of conceptualizing the distinctions between hypermedia, multimedia, and hypertext systems.

Hypermedia

Hypermedia systems can be described as multimedia systems with the added capabilities to provide support for the creation and maintenance of associative indices. The most appropriate definition for hypermedia is one by Frank Halasz who describes hypermedia systems as: "information representation and management systems that organize information into networks of multimedia nodes interconnected by links. Each node generally contains a large chunk of 'content,' such as a document, a drawing, or a voice annotation. The links are used to represent interrelations among these nodes" [Hala87, p. 350]. Hypermedia systems contain the following four components: 1) a computer-based data analysis and management system with 2) a high bandwidth human-computer interface and 3) capabilities permitting users to quickly and easily navigate, manipulate, and maintain multimedia data by 4) linking hypermedia objects containing chunks of unique, identifiable information into personalized associative indices.

Figure 2 - The Relationship Between Three Often Confused Terms



Data is stored in hypermedia systems as objects, each containing a variable amount of information stored on a variety of media. The media can be classified into four general categories: text, graphics, motion, and sound. The contents of a hypermedia object can be as small as a single bit of information or it can be quite large (i.e., a digitized two-hour movie occupying 550 megabytes on CD ROM disc). Each hypermedia object is stored in a separate window. Using system utilities, users navigate, manipulate and maintain structures of objects linked into associative indices.

Associative indices are used to maintain data in an essentially unconstrained form. Unconstrained implies that data models created using the workstation are not restricted to customary patterns of data organization. Data models can be created by users as dictated by personal taste. This permits organization of data from different perspectives or in different contexts. Robert Akscyn and Jeff Conklin have highlighted additional differences between hypermedia and other types of systems to help establish boundaries differentiating systems ([Aksc88] and [Conk87]). They note:

- **Multimedia Document Systems** lack user ability to create, manipulate, and maintain links;
- **Windowing Systems** lack database capabilities;
- **File Systems** are databases but generally lack linking capabilities and navigational support;
- **Outliners** have limited support for references occurring between outline entries and provide only hierarchical links between entries;
- **Text Formatting Systems** permit only hierarchical links between chunks of text; and

- **Data Base Management Systems** lack the ability to link multimedia items together and a unified user interface across media types.

Multimedia

Hypermedia systems are often mislabeled as multimedia systems and vice versa. Consequently, a whole array of developmental and commercial, multimedia and hypermedia products share the same assortment of names (see [Aksc88]). Multimedia is properly defined as: "Including or involving the use of several media" [Amer82, p. 821]. Multimedia-based systems generally seek to create a more effective presentation by synergistically combining various media (i.e., text, graphics, video images, etc.) to enhance user interaction with screen events [Brew86].

Hypertext

Hypertext has been variously defined as:

- " … a combination of natural language text with the computer's capacity for interactive branching, or dynamic display ... of a nonlinear text ... which can not be printed conveniently on a conventional page" [Nels67];

- A system permitting "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" [Yank85];

- A facility for machine support of arbitrary cross-linking between (textual) items [Conk86]; and

- Non-sequential reading and writing [Casa88].

However, the most useful definition is to define hypertext systems as similar to hypermedia systems but supporting only text-based objects. Recent efforts on the part of the National Institute of Standards and Technology have explicitly used the two terms synonymously [NIST90].

Features Of Hypermedia-based Systems

Eight noteworthy features of hypermedia systems are described below. Not all features are present in all hypermedia systems. However, many of them are present in many hypermedia systems.

Higher Potential Information Bandwidth in Human-Computer Dialogue

In both directions, hypermedia's multimedia capabilities offer the highest potential amount of information exchange because hypermedia excludes no media as potential input or output. In fact, Alan Kay has noted "the computer, as a medium itself, can be all other media if the embedding and viewing methods are sufficiently well provided" [Kay88, p. 254].

Better Information Access and Exchange Capabilities Through a Unified Software Interface

Hypermedia has been envisioned as a "cassette player for information." This implies anyone is capable of retrieving the information, provided they have access to the appropriate hypermedia technology. Consider the analogy of hypermedia providing an *information* infrastructure similar to the transportation infrastructure existing as railroads, highways, and airports [Atki87].

Non-traditional Linking of Presentation Media

Information stored on CD ROM, video tapes, audio recordings, and other non-traditional presentation and storage media can be incorporated within a collection of data. Individual data items can be represented with multiple complementary media types. The key is to store and present the user with information using the most appropriate format.

Support for Reporting Beyond Traditional Methods

Associative indexing capabilities permit users to create buttons and active screen objects. With these, users can branch to different data objects within a given context without having to access an index or otherwise employ a search mechanism.

Non-traditional Linking of Facts

The direct branching capability is not restricted to jumping around within the same data collection. Branching across conventional subject boundaries is an important capability, permitting users to focus on the content of the data and ignore its organization.

Availability of Concepts Unique to Object-oriented Systems

Object-oriented concepts such as encapsulation, classes, inheritance, and polymorphism provide a number of advantages over conventional systems. Encapsulation of data and methods in objects permits the data itself to know which type of access procedures to invoke in order to accomplish specific objectives. Classes permit data structures and methods to be shared among related groups. Inheritance simplifies aspects of organizing and classifying data items. Polymorphism enables the implementation of a single set of access methods for all types of objects.

Toolkit Availability

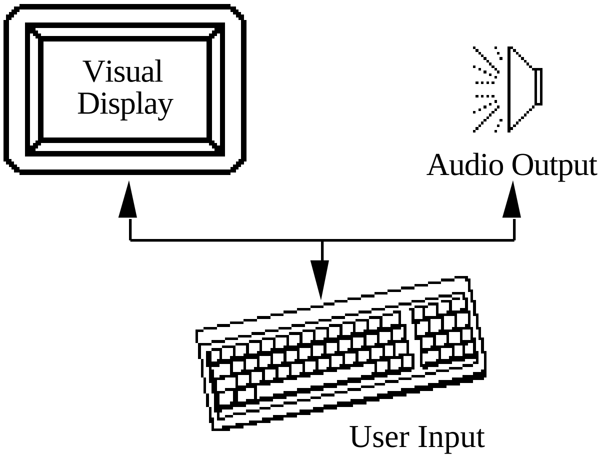
Hypermedia technology offers the ability to create and access a whole toolkit of utilities in a fashion similar to the UNIX Programmer's Workbench concept, but with expanded information handling capabilities.

Increased Interaction Modes

Traditional user-computer interaction consists of a user interacting with a visual display, occasionally supplemented with audio tones. Hypermedia interaction permits more total interactions. (See Figure 3 - Possible Hypermedia Interaction Modes.) This may be particularly important because recent research indicates that the source of a event may be particularly important [Gave88].

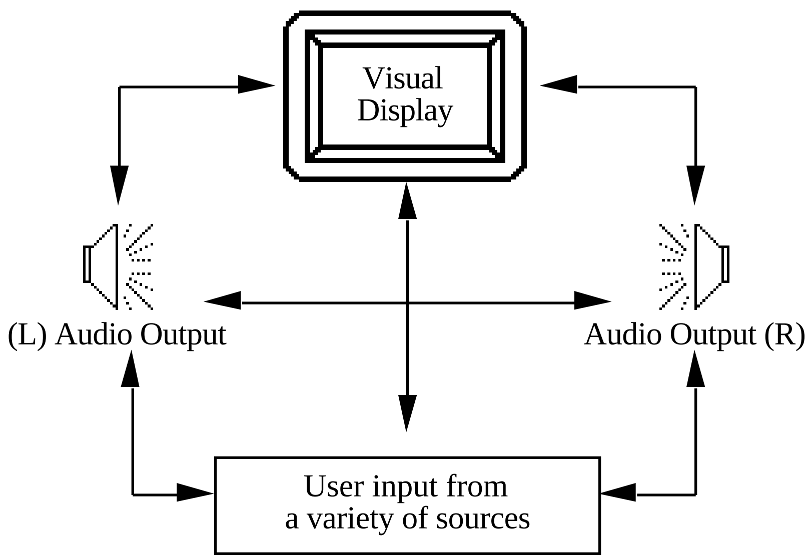
Figure 3 - Possible Hypermedia Interaction Modes

TRADITIONAL SYSTEM



4 possible types of interaction
- Visual display directed to user
- User response to visual display
- Audio response directed to user
- User response to audio channel

HYPERMEDIA WORKSTATION



12 possible types of interaction

- Visual display directed to user
- User response to visual display
- Right audio channel directed to user
- User response to right audio channel
- Left audio channel directed to user
- User response to left audio channel
- Visual display directed to right audio channel
- Visual display directed to left audio channel
- Right audio channel directed to visual display
- Left audio channel directed to visual display
- Right audio channel directed to left audio channel
- Left audio channel directed to right audio channel

(Adapted from [Brew86])

USING HYPERMEDIA TO AUGMENT REQUIREMENTS ENGINEERING

Classes Of Systems Able To Benefit From Hypermedia-based Requirements Engineering

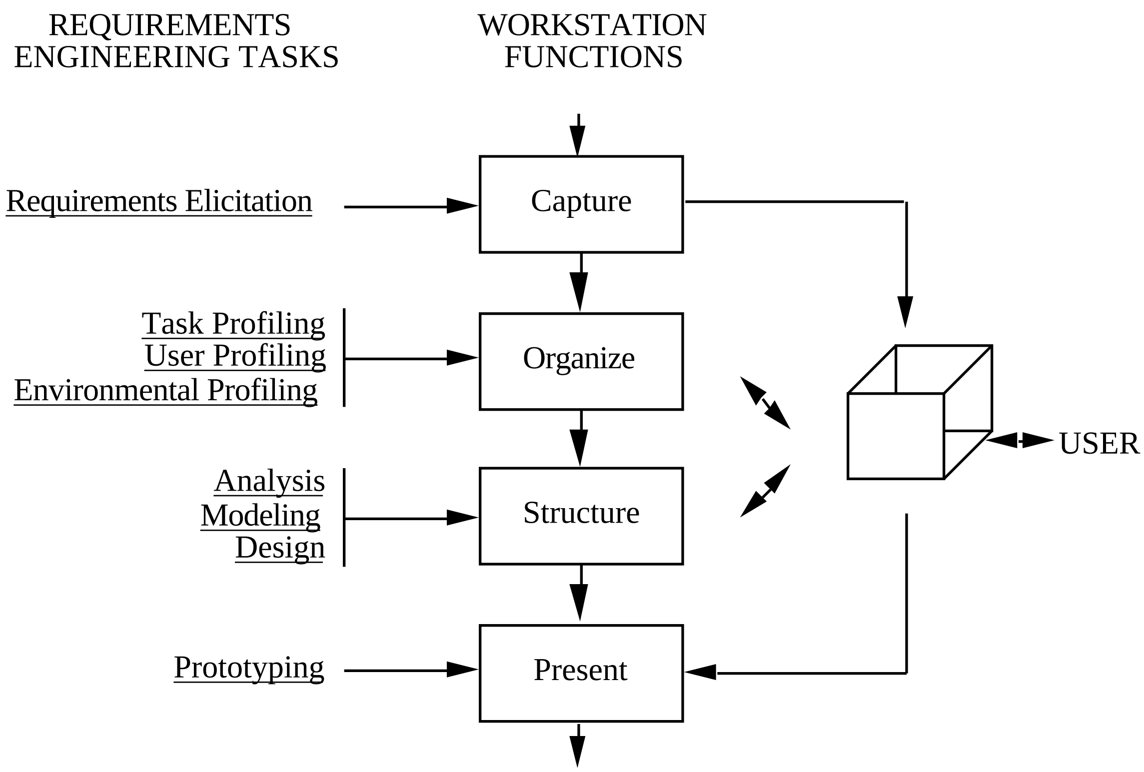
Applying hypermedia-based technologies to a given requirements engineering application typically requires relatively expensive hardware, often including more expensive storage technologies, audio/video playback and recording equipment, and color display technologies. Until the cost of this hardware decreases substantially, it will not be cost effective to use hypermedia in situations where the requirements are straightforward. Thus, hypermedia-based requirements engineering for systems where inputs and outputs are known and can be represented in text-based format, and processing requirements can be specified procedurally will not be cost effective for a number of years. On the other hand, there are several classes of situations where the use of hypermedia will be worth considering. These include:

- software systems where the problem solution cannot be described procedurally;
- software systems developed in an environment where current, text and static graphic-based requirements engineering tools and techniques cannot adequately capture and represent the issues involved (for example, the positions of the various stakeholders, political constraints, or lengthy background information);
- software systems where the design issues are too complex to be represented with static diagrams;
- software systems specifically concerned with sound-based, and/or motion-based requirements; and
- multimedia or hypermedia-based software systems.

Requirements Engineering Functions

The requirements engineering portion of the software development life cycle refers to all development activities occurring prior to system construction. This portion encompasses the most important phases of the software development life cycle: system requirements and system design. These phases incorporate the requirements engineering tasks of elicitation, task/user/environmental profiling, analysis, modeling, design, and prototyping and others. An examination of these tasks resulted in the identification of four basic requirements engineering functions. (See Figure 4 - Mapping Between Conventional Requirements Engineering Tasks and Requirements Engineering Functions.)

Figure 4 - Mapping Between Conventional Requirements Engineering Tasks and Requirements Engineering Functions



The first two functions are associated with the requirements analysis phase of the software development life cycle. They are 1) **capturing** information about the task to be performed, the system users, and any relevant organizational / situational characteristics, and 2) **organizing** the information into an easily accessible form for use in subsequent phases. The other two functions, associated with system design are 3) **structuring** the tasks identified during the requirements analysis into a model of the computer-based solution, and 4) **presenting** the model to the user in order to obtain corrective feedback. Each function is described in more detail below.

Capture

The capture function is concerned with acquiring information about three categories of requirements data.

- **User profiles** portray the users of the application along a number of dimensions including job function, level of computing experience, frequency of access, conditions under which the application will be used, and technical background. The goal is to identify the *shape and size* of the application's users.

- **Task profiles** describe the qualitative and quantitative aspects of the processes, functions, behaviors, and performance of the application. It is important to describe all aspects of the tasks including both automated and manual sub-tasks to gain the most complete picture possible.

- **Environmental profiles** concern specific, relevant or other factors impacting the situation. This dimension captures the *flavor* with which things are done in the host organization. (For example, in the military this translates directly into support for established doctrine.) [See Andr86].

Organize

The organize function is concerned with arranging the captured requirements information into an easily accessible form so specific objects can be accessed throughout subsequent phases of the life cycle. Dallas E. Webster has noted " ... the way in which information is represented is frequently the critical factor in determining whether problems are easy or difficult to resolve, or whether a (application) does or doesn't work. So we must pay careful attention to the choice of representation media and methods to make sure they are sufficient or, more realistically, extensible or adaptable" [Webs88, p. 10]. Hypermedia technologies offer the ability to represent information using the most appropriate media. Once organized, the requirements information forms the basis for the software system design. Coders reference this information to supplement the system design when developing code. And when maintenance personnel approach a problem, they access the requirements in order to familiarize themselves with the system's function(s).

Structure

The purpose of the structuring function is to increase the connectivity of the requirements information while designing a model of the computer-based solution for the host environment. Analysis of requirements information identifies constraints which then become design issues guiding model development. Webster has also stated, " ... the design process can be viewed as successive elaborations of representations, such as adding more information or even backtracking and exploring alternatives" [Webs88, p. 8]. Additionally, an effective environment for the design processes provides support for the required visual momentum and cognitive processes [Wood84] associated with these tasks.

Thus, the primary requirement of the structuring function is the ability to rapidly and easily navigate, manipulate, and maintain individual requirements data items. This reduces delays in system operation and cognitive processes that can cause the requirements engineer to loose the current train of thought, wasting valuable time, and potentially introducing other mistakes to the process. Requirements engineers require integrated tools permitting them to quickly manipulate multimedia chunks into associative indices forming the basis for the application model. They should be able to **directly** combine, divide, link, and redefine the chunks created during the organize function.

Present

The goal of the presentation function is to prevent disconnects from occurring between the representation of the application presented to the user by the requirements engineers and the user's / sponsor's concept of the application. This disconnect occurs when the user gains a different mental interpretation of the application prototype presented by the designer.

Presentation of the model to the user also can take a number of forms ranging from written reports to well-developed interactive prototypes. Written prototypes are easiest to produce but provide the least valuable information in terms of user feedback. It seems more a prototype is based on images, functions, and sounds familiar to the user, the smaller the chance of a disconnect occurring. Another rule of thumb is that the closer a system approximates the actual look and feel [Samu89] of the proposed system, the better the user will be able to correctly assess the system's ability to meet the required needs. As a result of the capturing, organizing, and structuring functions, many sounds and images are already contained as objects in the workstation. These should be incorporated into the prototype.

Ideally, the requirements engineering process repeats until the sponsor and the requirements engineers are satisfied with the correctness of the requirements data, its organization, the model of the system and the prototype. The application of the requirements engineering functions is shown in the context of a prototyping development methodology in Figure 5 - Suggested Requirements Engineering Process.

The information flow during the development of a software application can also be viewed as the result of a series of transformations on real world inputs. The goal of the first transformation is to obtain a text-based representation of the problem at hand. Typically input data is transformed from its representation in the multimedia, real world into a two-dimensional, serial document, although it is often supplemented with a table of contents, indices, and static graphs (i.e., various flow diagrams). The system design is generally made to fit a paper-based media. Thus requirements data enter the system through a keyboard and representations of the design leave the development system through a printer of some sort. (See Figure 6 - Data Transformation During Requirements Engineering.)

Figure 5 - Suggested Requirements Engineering Process

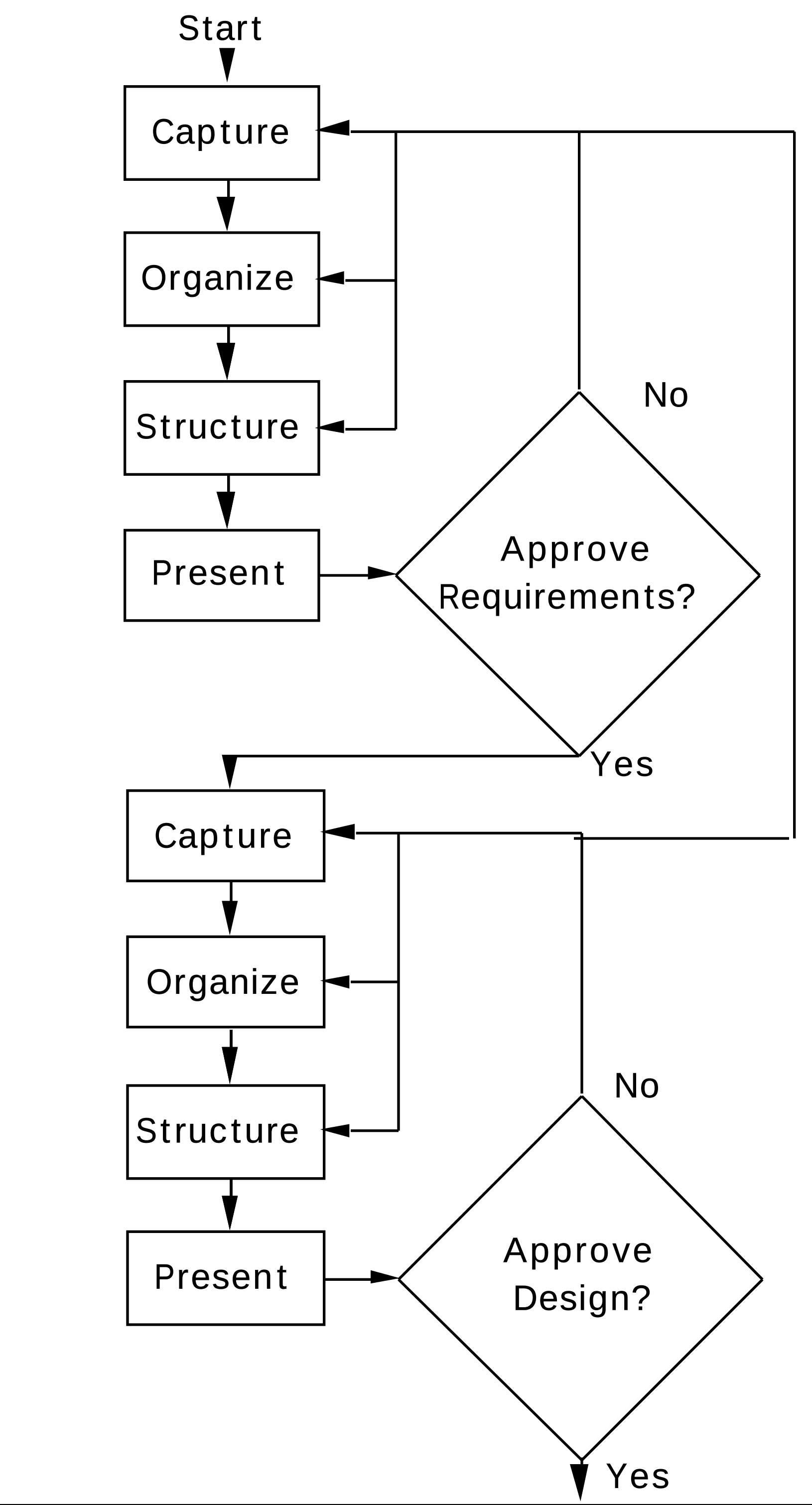
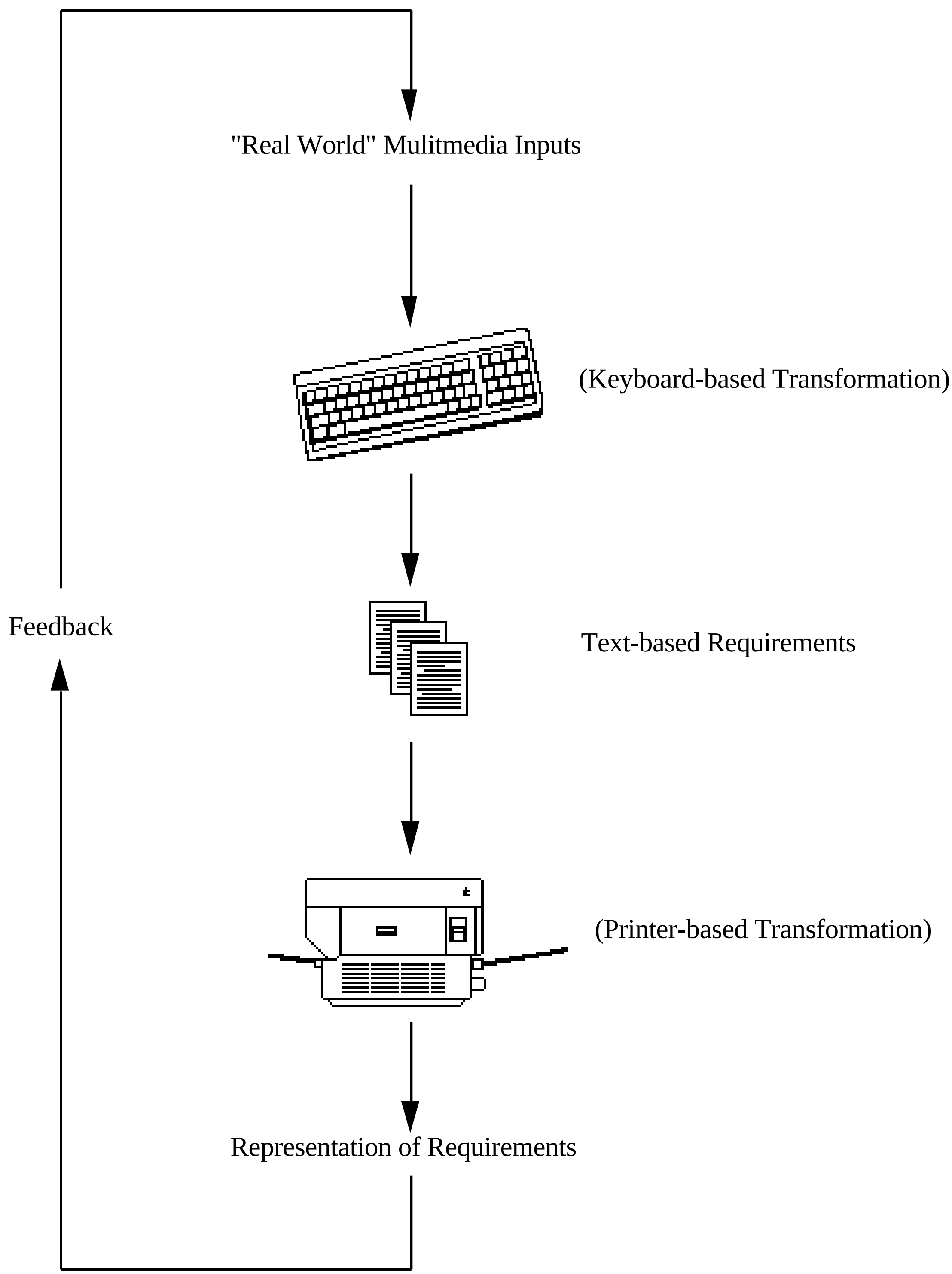


Figure 6 - Data Transformation During Requirements Engineering



HYPERMEDIA SUPPORT FOR REQUIREMENTS ENGINEERING FUNCTIONS

Hypermedia-based tools and techniques offer specific support for each of the four requirements engineering function. Support for each function is described below.

Support For The Capture Function

Hypermedia support for the capture function is concerned with supporting processes used to obtain requirements information. The goal is to increase the amount and quality of information available to the requirements engineer. This is accomplished by increasing the volume of inputs and reducing the transformations required to get the requirements information into a manipulable format. Augmented data capture facilitates the user's ability to capture requirements information from a variety of external sources and to create new data internally by transforming objects from one media to another. Sounds, pictures, and motion-based input data can supplement traditional text and static graphics-based representations of requirements information.

Support For The Organize Function

Hypermedia support for the organize function is concerned with arranging the captured information, within dimension, into easily accessible task, user, and situational profiles. The goal is to assist the engineer to effectively and efficiently group requirements information into object collections. A partial list of useful object collections include objects grouped by:

subject area within media classification,
subject area irrespective of media classification,
requirement,
design constraints,
design issues,
design features,
prototype features.

This process can be supported by

1. augmenting the requirements engineer's ability to transform unstructured objects into objects that can be identified by the search operations executed by the user, and

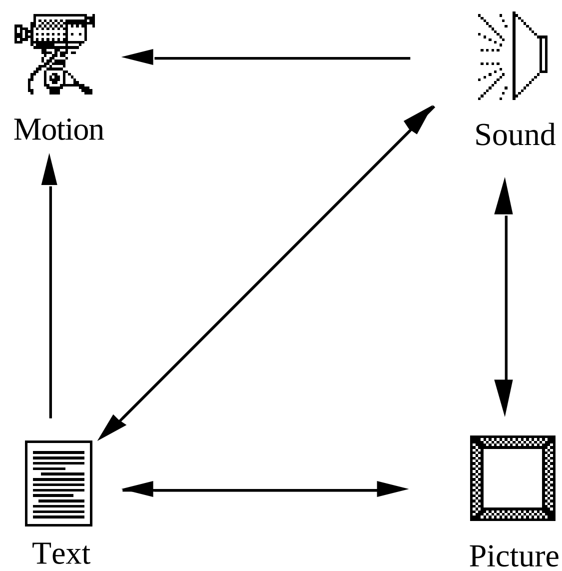
2. providing the ability to quickly and easily manipulate objects. Manipulations supported include combine, divide, re-define, and attach.

Hypermedia can support the use of an unconstrained data model as defined above. Support for the organization function permits requirements engineers to quickly and easily access multimedia objects. Object can be manipulated and redefined as additional objects. Users can combine objects with other objects in many combinations to create the most appropriate representation of the requirements information. (See Figure 7 - Potential Object Combinations.) The system supports the use of many-to-many relationships between objects and augments user efforts to create and maintain those links. In addition, all requirements information is maintained on-line and is accessible through a single interface.

Support For The Structure Function

Hypermedia support for the structure function is concerned with increasing the connectivity of the information by establishing linkages across subject boundaries. The goal is to provide increased situation comprehension and augmented solution development. Two subfunctions supporting the requirements engineer’s efforts to structure requirements information are attach and link. The attach subfunction is used to create composite objects - objects made up of multiple media. The link function supports the process of creating associative indices.

Figure 7 - Potential Object Combinations



Support For The Present Function

Hypermedia support for the present function is concerned with ensuring that the requirements engineer and the sponsor have the same picture of the proposed application. The goal is to increase the communication between the engineer and the sponsor through the use of prototyping strategies. Throughout the prototype evaluation users are presented with familiar sounds, images, and functions. This is accomplished by providing support for:

1. **Human-computer Interface Design** - The workstation facilitates the presentation of prototype interfaces using high bandwidth multimedia human-computer interface. This simplifies the use of realistic interface metaphors.

2. **Program Flow** and **Program Control** - The workstation supports simulation program flow and simulation of the host environment so the application can be observed in its intended environment.

3. **Presentation Sequencing** - This workstation provides interactive prototyping facilities for construction of prototypes closely approximating the look, feel, and operation of the application to be developed.

A HYPERMEDIA WORKSTATION FOR REQUIREMENTS ENGINEERING

Research efforts at the Center for Software Systems Engineering at George Mason University have resulted in the development of a hypermedia workstation supporting requirements functions as described in this chapter. The workstation was assembled using off-the-shelf hardware components and a specially developed software shell. The software shell provides the capability to navigate among, manipulate and maintain, requirements information existing on various media. The shell enables users to create and modify hypermedia objects, and links between objects. The workstation is based on a Macintosh IIx computer. On-going research is evaluating the potential contribution hypermedia-based tools and techniques embodied in the workstation can make to the requirements engineering process [Aike89].

REFERENCES

Aike89 Peter Aiken, "A Hypermedia Workstation for Requirements Engineering," Ph.D. Dissertation, School of Information Technology and Engineering, George Mason University, Fairfax, VA., 1989.

Aitk76 Hugh G. J. Aitken, *Syntony and Spark: The Origins of Radio*, New York: John Wiley & Sons. 1976.

Aksc88 Robert Akscyn, *Tutorial: Design and Use of Hypermedia Systems*, ACM/SIGCHI Conference on Human Factors in Computing Systems, May 15, 1988.

Amer82 *American Heritage Dictionary*, Second Collage Edition, Boston: Houghton Mifflin Company, 1982

Andr86 Stephen J. Andriole, *Handbook for the Design, Development, Evaluation, and Application of Interactive Military Decision Support Systems*, Marshall VA: International Information Systems, 1986.

Brew86 Bryan Brewer, "Compact Disc Interactive Audio," in (Steve Lambert and Suzanne Ropiequet, editors) *CD ROM: the new papyrus*, Redmond, WA: Microsoft Press, 1986, pp. 273-294

Bush45 Vannevar Bush, "As We May Think," *Atlantic Monthly* July 1945, pp. 101-108.

Casa88 Lou Casabianca, "Ted Nelson Hypermedia Magician," in *Hypermedia* Summer 1988, p. 23-25.

Conk86 Jeff Conklin, "A Survey of Hypertext," MCC Technical Report No. STP-356-86.

Deli86 Norman Delisle and Mayer Schwartz "Contexts -- A Partitioning Concept for Hypertext" *Conference of Computer-Supported Cooperative Work - Proceedings 1986*, Austin TX, 1986, pp. 147-152.

Enge62 Douglas Engelbart, *Augmenting Human Intellect: A Conceptual Framework* Summary Report, Stanford Research Institute, on Contract AF 49(638)-1024, October 1962, 134 pp.

Garg87 Pankaj K. Garg and Walt Scacchi, "On Designing Intelligent Hypertext Systems for Information Management in Software Engineering", *Hypertext '87 Papers*, pp. 409-433.

Garr86 L. Nancy Garrett, Karen Smith and Norman Meyrowitz "Intermedia: Issues, Strategies and Tactics in the Design of a Hypermedia Document System" *Conference of Computer-Supported Cooperative Work - Proceedings 1986*, Austin TX, 1986, pp. 163-174.

Gave88 William W. Gaver, personal communication 8/26/88.

Hala87 Frank G. Halasz, "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems" *Hypertext '87 Papers*, pp. 345-366.

Kay88 Alan Kay and Adele Goldberg, "Personal Dynamic Media" in (Adele Goldberg, editor) *A History of Personal Workstations*, Reading, MA: Addison-Wesley Publishing Company, 1988, pp. 254-263.

Lick60 J. C. R. Licklider, "Man-Compute Symbiosis," *IRE (now IEEE) Transactions on Human Factors In Electronics* March 1960, pp. 4-11.

NIST90 personal communications with Jean Baronas, National Institute of Standards and Technology, Room B263, Building 225, Gaithersburg, MD 20899, January 24, 1990.

Nels88 Theodor H. Nelson, "Managing Immense Storage: Project Xanadu provides a model for the possible future of mass storage," BYTE January 1988, 13(1):225-238.

Samu89 Pamela Samuelson, "Whey The Look And Feel Of Software User Interfaces Should Not Be Protected By Copyright Law," *Communications Of The ACM* May 1989, 32(5)563-572.

Sher89 Chris Sherman, *The CD ROM Handbook*, New York: McGraw-Hill, 1989.

Smit88 John B. Smith and Stephen F. Weiss, "Hypertext," *Communications of the ACM* July 1988, 31(7):816-819.

Webs88 Dallas E. Webster, "Mapping the Design Information Representation Terrain," *IEEE Computer* December 1988, 21(12):8-23.

Yank87 Nicole Yankelovich, Bernard J. Haan and Steven M. Drucker, "Connections in Context: The intermedia system," *IRIS Technical Report 87-3*, 1987.