Inco-Copernicus Programme

INTELLECT

(INTELligent Learning Environment for Course Telematics)

PL961099

Research Report

INTELLECT-C2-2

Hypertext Hyperlinking Review

Pandelis Ipsilandis
George Blanas
Technological Education Institute Of Larissa

Introduction

The idea of hypertext was originated by Bush (1945) in the description of a device called "memex" that can be used as a supplement to human memory in which an "individual stores his books, records and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility." The essential feature of memex, that was never developed, is its ability to tie two items together.

The first system that had some hypertext features was NLS (oN Line System) a collaborative remote system used to store all research papers, memos, and reports in a shared workspace that could be cross-referenced with each other (Engelbart, 1963).

The term was first made up by Nelson (1965) who defined it as "a body of written or pictorial material interconnected in a complex way that could not be conveniently represented on paper and may contain summaries or maps of its contents and their interrelations and it may also contain annotations, additions and footnotes from scholars who have examined it." Nelson has been working on his vision of a "docuverse" (document universe) where "everything should be available to everyone. Any user should be able to follow origins and links of material across boundaries of documents, servers, networks, and individual implementations. There should be a unified environment available to everyone providing access to this whole space." (Nelson, 1987).

Conklin (1987) considers hypertext as a kind of semantic network which mixes informal textual material with more formal and mechanized processes, and as an interface modality that features link icons or markers that can be arbitrarily embedded with the contents and can be used for navigational purposes

According to Nielsen (1990) a hypertext system is a database system that has no regular structure and provides a totally different and unique method of accessing information.

According to Rao & Turoff (1990) a hypertext system is made of nodes (concepts) and links (relationships). A node usually represents a single concept or idea that can contain text, graphics, animation, audio, video, images or programs. It can be typed (such as detail, proposition, collection, summary, observation, issue) thereby carrying semantic information. Nodes are connected to other nodes by links. The node from which a link originates is called the reference and the node at which a link ends is called the referent. They are also referred to as anchors. The contents of a node are displayed by activating links. Links connect related concepts or nodes. They can be bi-directional thus facilitating backward traversals. Links can also be typed (such as specification link, elaboration link, membership link, opposition link and others) specifying the nature of relationship. Links can be either referential (for cross-referencing purposes) or hierarchical (showing parent-child relationships). Activation of link markers display nodes.

Bieber (1993) defines hypertext systems as complex information systems that provide non-sequential flexible access to information by incorporating the notions of navigation, annotation, and tailored presentation. Data is stored in a network of nodes connected by links.

Advantages of Hypertext Format

Enhancing encyclopedic information into hypertext format has many advantages (Raymond & Tompa, 1988):

1. Hypertext form can support good browsing capability.

2. Electronic media can store large amounts of information.
3. It can provide better visual prominence and more rapid navigation through huge number of entries - a key mechanism to be employed in dynamic formatting of entries according to user specifications.
4. Most users refer to dictionaries, encyclopedias, and training manuals as part of more extended tasks. They would like to save their results and queries for future use, use annotation facilities, the ability to transfer OED text segments to other documents, tools to sort and filter quotations, and tools for statistical analysis of variables.

Basic Features of Hypertext Systems

A hypertext system has the following basic features (Balasubramanian, 1994):

1. A Graphical User Interface, with the help of browsers and overview diagrams, helps the user to navigate through large amounts of information by activating links and reading the contents of nodes.
2. An authoring system with tools to create and manage nodes (of multiple media) and links.
3. Traditional information retrieval (IR) mechanisms such as keyword searches, author searches etc. There are also attempts to incorporate structure queries along with content queries - retrieving a part of the hypertext network based on some user-specified criteria.
4. A hypermedia engine to manage information about nodes and links.
5. A storage system which can be a file system or a knowledge base or a relational database management system or an object-oriented database management system.

Implementation Issues

Converting text to hypertext has been a classic problem while dealing with very large information spaces such as training manuals, encyclopedias, and dictionaries. The following are some of the issues involved in converting text to hypertext (Glushko, 1989):

1. Identifying documents that would benefit readers if converted to hypertext form.
2. Determining procedures to convert them to hypertext format.
3. Preparing documents in an electronic format from paper or other forms.
4. Identifying nodes and links and classifying them into various types (to capture semantics). An important problem related to this issue is called the fragmentation problem. It is still difficult to identify text units that can be separate modules and also serve as cross-references for other entries. Links should follow some model of the user's need for information in some particular context. Deciding on the level of granularity is a difficult problem. Too fine the granularity, greater the problem of fragmentation. Too coarse the granularity, greater the need or the display of large entries.

1. Also fragmentation tends to make an implicit structure (such as a subtle treatment of a theme that may communicate an idea more artistically) explicit, taking away the expressiveness of the statement. Therefore, we have to find means to reduce segmentation of ideas and loss of structural information due to the manipulation of the semantic structure of a linear document.

5. Determining the target of a link as a complete entry, a sub entry, or a derivative form is a challenging task. This involves determining the right part of speech, the etymological root, and applying sense-disambiguation to identify a particular meaning.
6. With present-day video monitors, the display of large entries in their entirety is still a problem. This can be partly solved by having fisheye views and abbreviations. Structural information can be extracted from the tags and employed in the construction of a structural view.
7. Performing the conversion and verifying the results.

The following can be some of the criteria in converting linear documents into hypertext format, both manually and automatically (Glushko, 1989):

1. Utmost care is required while identifying text units as nodes that can be separate modules and still be sufficient enough to be cross-references for other entries.
2. A good design rule is to choose as the basic unit of text the smallest logical structure with a unique name (such as the title for an entry) - this can be used as a selection key in a hierarchical browser, in search lists as candidate keys, as bookmarks, and embedded cross-references.
3. Pages or paragraphs are less suited as hypertext units because they do not form convenient handles for manipulation.
4. It is very important to understand both the explicit and implicit link structures in the printed version of the material. Careful decisions have to be made as to what links to create and what to disregard.
5. It is important to understand the user's task and to support links that follow some model of the user's need for information in some particular context. It is essential NOT to link items that are related in idiosyncratic or superficial ways. Such hypertext links lead to "spaghetti documents". A careful analysis needs to be done as to what implicit and explicit hypertext structures users make use of in the linear document.
6. The organization of the material should be open and flexible. Different kinds of views should be available for different users. For example, a repair manual can contain a training view, a troubleshooting view, a routine maintenance view and a purchaser's view. View descriptions may appear as alternate overview diagrams or webs of information.

Hypermedia templates are defined as sets of pre-linked documents that can be duplicated (Catlin et al., 1991) or as partially-created, properly formatted collection of document skeletons that can be filled in by the user (Rao & Turoff, 1990).

Templates automate the process of creating hypermedia collections by creating the "skeletons" of documents and linking them. They facilitate the design, organization, and presentation of a collection of knowledge in the form of hypertext.

The template can be considered as a composite object comprised of other objects such as nodes and links. The usage of a template can speed up the process of an average user's understanding of the underlying hypertext model or the metaphor. Without a template, a hypertext author will have to start constructing the hypertext collection of ideas from the beginning. The following are some requirements for a hypertext system to provide templates (Balasubramanian, 1994):

1. It should provide some generic operations to create, duplicate, edit or delete a template. Duplication should yield empty documents with nodes and links.
2. There should be facilities to add contents to empty documents, list templates and their constituent documents and links, to display an overview of the template, to access a template by its type ("get a copy of the planning template"), by author, or by creation date.
3. There should be control operations to displaying an overview of the template, to zoom into specific link sets or webs or sub graphs and look at the contents of documents.
4. Strategic choices must exist to find out the master template from which a duplicate was created and to edit the master template. Editing a master template should propagate the changes to all templates created from it.
5. Facilities should exist to specify formats and screen layouts for a template and to add help.
6. Reactive choices must be provided to directly manipulate the contents of documents within a template such as editing, deleting, creating new links etc.

Hypertext Authoring Guidelines

The comprehension and navigation of a hypertext document depends on the reader's ability to construct a coherent mental representation. It is the author's responsibility to ensure the construction of the hypertext document as a coherent entity. The construction of a coherent hypertext document can be considered to be a design problem. There are standard guidelines for writing hypertext documents. Thuring et al. (1991) suggest that a hypertext document should consist of the following three components - the content part, the organizational part, and the presentation part, described as follows:

- The Content Part

Nodes and links can be considered as design objects. Properties (semantics) can be associated with these design objects in order to introduce coherence in a hypertext document.

The content part contains design objects that carry information. They are content nodes that contain information and content links that connect content nodes based on semantic relationships. Content nodes can be either atomic or composite in nature. Content links can be typed specifying the exact nature of the semantic relationship. They can be classified into three types or levels:

Level One: Links with no labels.

Level Two: Links with labels describing global semantic relationships such as "is discussed by", "is illustrated by".

Level Three: Links with more specific labels such as "is criticized by", "is shown graphically".

This classification of links is similar to the more elaborate classification of nodes and links in the hypertext framework developed by Rao and Turoff (1990). According to Thuring et al. (1991) the author creating a hypertext document can initially create a Level Two link to show a general relationship between content nodes. As the author becomes more clear about the relationship between two nodes, the link labels can be changed to Level Three. Thus, "the levels of link label hierarchy support a continuous refinement of the links depending on author's current state of knowledge

While creating the content part, the following design rules can be applied:

a. Composite content nodes should be used to hierarchically structure the content of the document into domain specific sub-units of information.

b. The label of a link should be as specific as possible and should constitute a comprehensible sentence together with the names of the source and destination nodes.

- ## The Organizational Part

Design objects of the organizational part increase coherence by structuring the network under a reader-oriented perspective. Using such an approach, the author can tailor variants of a document for different audiences.

Structure nodes organize content nodes and links in a specific manner. Each structure node has a name and a starting node. These can be of two types (Balasubramanian, 1994):

a. Sequencing nodes that allow the author to define the reading sequence through the content net. Readers can read only those content nodes that are determined by the sequencing node.

b. Exploration nodes allow the reader to explore - the reader can simply follow the content links to explore the subnet.

While sequencing nodes constrain the reader's navigation through the document, exploration nodes allow unconstrained access to its content part.

Structure nodes can be connected by structure links which are also classified into two types (Balasubramanian, 1994):

a. Sequencing links associate the content of each sequencing node with a presentation sequence. They can be used to define ordering such as linear sequence, branching sequence etc.

b. Exploration links provide access to exploration nodes. An exploration link is embedded into a sequencing node and points to the beginning of an exploration node.

Sequencing nodes along with sequencing links can present different presentation sequences such as sequential paths, branching paths, and conditional paths.

The following design rules can be applied while creating the organizational part (Balasubramanian, 1994):

1. Choose an appropriate starting point to serve as an introduction to the document.

2. Construct appropriate paths based on reader's interests and knowledge. This can be done by ordering sequencing nodes and links and providing additional information using exploration nodes and links. Thus, the author can create multiple versions of the document some having strictly linear sequences, some having branches, and conditional paths, and some a combination of all three.

- ## The Presentation Part

The presentation part is concerned about the actual display of structure and content and provide the means of navigation. Authors can adopt three styles (Balasubramanian, 1994):

a. Textual Style: There is no graphical display of the structure, the presentation being limited to the display of the content of one or more nodes.

b. Graphical Style: There is a graphical display, such as an overview map, of the structure.

c. Combined Style: Both overviews and the ability to open nodes are provided.

The combination of the content part with the organizational part and the presentation styles would greatly facilitate comprehension and navigation.

- Dynamic Hypertext

Stotts and Furuta (1991) describe virtual structures as dynamic adaptation of hypertext structure. It involves collecting information from user interaction with a hypertext system, making inferences and decisions based on this information and creating appropriate physical changes in the document at appropriate times. Adaptation can occur at two levels - behavior of the document (timing of sequences, providing automated help, presenting collections in parallel or in sequence etc) and structure of the document (the way the nodes are linked).

According to Stotts and Furuta (1991) a hypertext document can be considered to have two layers - a fixed underlying information structure that is created by the hypertext author and a flexible structure that is superimposed on the former and is tuned to each user's requirements. The flexible layer can be generated dynamically. The manner in which information is organized and presented can be altered without actually changing the information relation contained in the original links. This is similar to Bieber's concept of bridge laws which simply map an application's non-hypertext data to a hypertext interface without changing the underlying data. Thus, a document can change to adapt the needs and preferences of individual users, the author's original structure being retained. Such a dynamic adaptation technique has been implemented in the Petri-net based Trellis system developed by Stotts and Furuta (1991).

- Linearization of Hypertext

The connection between the linear form and the hypertext document can be maintained over time. This can be done using document cards which allow users to automatically generate a linear document, in a card, covering some portion of the network. Changes to the document are made in the source cards from which the document was compiled. This allows portions of the paper to be visible in different windows and simultaneously accessible (Trigg & Irish, 1987).

The SmarText Electronic Document Construction Set, a software product that automates the creation and browsing of large hypertext document, presents multiple views of non-linear text in a linear fashion. SmarText readers can choose to traverse one path out of many possible paths. A path is essentially a linear presentation of specific nodes connected by specific links. The text, the index and outlines are constrained by the selected view or path (Rearick, 1991). A path can also be used to collect all interesting documents to form a linear document that can be printed (Utting & Yankelovich, 1989). Another possible method of linearizing hypertext is to take the user's history of interaction and print the contents of the nodes that were traversed during a particular session.

Navigation Issues

The promise of hypertext lies in its ability to produce complex, richly connected and cross-referenced bodies of information. However, it can also become a complex system of tangled webs, confusing both authors and readers. According to Conklin (1987) disorientation and cognitive overhead are the two most challenging problems related to hypertext. He feels that these two problems "may ultimately limit the usefulness of hypertext."

The problem of disorientation or "getting lost in space" arises from the need to know where one is in the network, where one came from, and how to get to another place in the network. In traditional text, it is not easy to get lost. There is the table of contents of topics with page numbers, the index with keywords and page numbers, and also bookmarks. However, in a complex hypertext network, with thousands of nodes and links, it is more than likely that the reader will get lost.

Cognitive overhead is the additional mental overhead on authors to create name, and keep track of nodes and links. For readers, it is the overhead due to making decisions as to which links to follow and which to abandon, given a large number of choices. The process of pausing (either to jot down required information or to decide which way to go) can be very distracting. It can become a serious problem if there are a large number of nodes and links.

Graphical browsers serve as overview displays for large bodies of information, especially in a hypertext system. The user can zoom in to see any portion of the browser in detail but cannot look at the details of the entire network because of screen space limitations (Utting & Yankelovich, 1989).

Graphical browsers help reduce disorientation by providing a two-dimensional spatial display of the hypertext network. They also help minimize cognitive overhead by showing a small part of the network. Though general browsing is a low cognitive load operation, it is inefficient for directed search tasks or fact retrieval. Also, a graphical browser itself can become a tangled web if the hypertext network is large and if the display has to be updated because of dynamic changes in the network.

Disoriented users need a sense of context and spatial and temporal information. Overview diagrams, both at the local and global levels, serve as excellent navigational aids (Utting & Yankelovich, 1989). Global overview diagrams provide an overall picture and can also serve as anchors for local overview diagrams. Local overview diagrams provide a fine-grained picture of the local neighborhood of a node. Overview diagrams for large systems might become complex and might introduce navigational problems of their own (Nielsen, 1990a). Nielsen feels that in order to reduce the propagation of links from the local diagrams to global diagrams, weights can be assigned to the links based on their relevance to the user; this will trim the edges of the graph at the global level. Also, anchors can be differentiated, graphically, based on their estimated relevance to the users. Statistical analysis of the most frequently traversed links by a number of users might provide valuable insights into developing system-generated overview diagrams. In most current hypertext systems, readers have a problem trying to understand the material presented because they view it in the wrong order or they simply cannot comprehend easily. The concept of a path can help solve this problem by allowing authors to determine an appropriate order of presentation for a given audience. It will reduce both disorientation and cognitive overhead since users will follow a pre-defined path which will also narrow down their choices.

The concept of paths was extended to "guided tours" in NoteCards (Trigg,1988). A guided tour is a system-controlled navigational tool that can be entered and exited at the user's will. Progress can be monitored using maps or overviews of the hypertext database. Editing tools are available for creating and managing both tabletops and guided tours.

Nielsen (1990b) feels that even though a guided tour seems to limit the very purpose and potential of hypertext, it can be used to introduce the concept of hypertext to new readers and for small systems. Different levels of guided tours can be designed for different types of users.

Backtracking allows visiting previously visited nodes. Backtracking in a linear fashion and also the ability to arbitrarily jump to previous nodes helps people to bail out of difficult situations. The most preferred method is the path-following principle which allows traversing, in reverse order, those nodes that were previously visited since this approach relies on the user's memory of his or her own navigation behavior. The structure-oriented feedback approach allows users to directly jump to a node without backtracking. However, experiments have shown that combining these two methods might lead to confusion (Nielsen, 1990a).

Nodes that are visited can be timestamped (along with the accumulated time spent at each node) and maintained in a chronological order (Nielsen, 1990a). This feature will help users, at a later date, to recognize whether the node was visited before and if so, the duration allowing users to change their viewing behavior accordingly.

Arbitrary jumps or gotos can be provided enabling users to go to any node in the system. This can be accomplished by zooming in and out using an animated iris that opens for anchored jumps and closes for return jumps. "Landmarks" or prominent nodes can be provided which can always be accessed from anywhere in the system. The concept of a bookmark is similar to a history list except for the fact that a bookmark is placed by the reader only if a particular node will be of interest at a later date. Hence, a bookmark list is smaller more manageable and more relevant to the user. Experiments have shown that random jumps from anchors leading to multiple destinations can be very confusing to users (Nielsen,1990a). This can be avoided by listing the possible destinations when an anchor is activated allowing the user to choose a predictable path.

Embedded menus, as opposed to explicit menus, allow the user to select a word or item embedded within the text of a node and can be selected using a touch screen, cursor keys or mouse pointer. Embedded menus are a better way of indexing for hypertext systems since they emphasize the understanding of concepts. They highlight semantic relationships rather than physical relationships. They provide meaningful task domain terms (as opposed to computer domain terms) and concepts, thereby reducing disorientation (Marchionini & Shneiderman, 1988). Embedded menus reduce the "loss of context" feeling by being part of the information being displayed. They provide information hiding (layers below are shown only when requested). The extent of "embeddedness" can be varied depending on the skill level of the user. They are suitable for learning-by-browsing systems as in museums. Further research is required regarding the negative aspects of using highlighted embedded menus. It is possible that they may cause disruption, reducing speed and comprehension.

Hypertext navigation is also restricted by the physical limitations of the display screen. The inability to view large amounts of information at any one time due to the small size of computer displays falls under the category of context-in-the-small problems (Nielsen,1990a). This includes the issue of readability, when the contents of a node do not fit into one screen and have to be carried over to other screens. Larger displays can only partially solve this problem. Conventional scrolling techniques use arrow keys or paging keys allowing only vertical movement to various places on the screen. While scroll bars do allow two-dimensional navigation, it is not easy to focus on a particular region of interest.

Some researchers feel that learning systems should have some amount of disorientation and cognitive overhead in order to facilitate exploration and learning (Mayes et al., 1990). They feel that while most researchers are concentrating on navigation through information space, very little work has been done on navigation through "conceptual space". They contend that simply following links to nodes does not necessarily provide effective learning - "they do not tell us ABOUT anything, but only WHERE it is."

In learning systems, disorientation in conceptual space is required sometimes in order to explore and learn. Thus, users need to be guided not only by system information but also by discovery. The issue of cognitive overhead has not been found significant in learning systems (Mayes et al., 1990). The authors say, the very question "what to do next, will enrich the process of learning rather than detract from it."

Usability and Evaluation of Hypertext

Although the application of metrics to hypermedia has already stimulated considerable interest (Botafogo et al. 92), (Rivlin et al. 94), (Garzotto et al. 94), (Garzotto et al. 95), (Hatzimanikatis et al. 95), (Yamada et al. 95), they have been developed in an ad-hoc fashion, expressing measures in an ambiguous manner and thereby limiting their application. For example, there are many different decisions that have to be made when defining a usability measure or maintainability measure. These decisions have to be made with respect to the goal of the measure and by defining an empirical model based on a hypotheses. Unfortunately, many measures proposed in the literature do not have the motivation behind these decisions documented, making it difficult to understand the underlying assumptions of the measure.

In relation to the contents, which - in the case of hypermedia applications for education - reflects a particular theory of learning, there are two theoretical views that have particular relevance to the design and use of hypermedia applications: The Cognitive Flexibility Theory (CFT) (Jacobson and Spiro 95a), (Jacobson and Spiro 95b), (Spiro et al. 95) and the Situated Cognition Theory (SCT) (Jacobson et al. 96), (Jacobson and Archodidou 97). They are both theories of learning and incorporate not only the planning principles, but they allow deep learning of complex domains and the transfer of knowledge to new situations. They have been tested by empirical evaluations and have had very positive results so far (Rana and Bieber 97), (Carvalho and Dias 97), (Jacobson 97).

The principles of the metrics (Mendes 97) are based on the goal-based framework for software measurement proposed by Fenton and Pfleeger (Fenton and Pfleeger 96), and on the guidelines from the DESMET project (Kitchenham 96), (Kitchenham 93). Both have been extensively used in experiments in the software engineering field (Harrison et al. 95), (Briand et al. 96), (Basili and Rombach 88), (MacDonell 91).

Wright (1991) feels that the following five issues have to be examined while evaluating hypertexts:

1. Adequacy of the content and the interface. Hypertext design involves a mixture of decisions relating to the content, functionality, display, and control. The usefulness of hypertext depends on its purpose, ease of navigation, and the population domain. The authors have to determine tradeoffs among various design principles. For example, for a learning system the authors would have to not only look into the outcomes of using hypertext but also the quality of learning. In short, there is no upper limit on adequacy.

2. Acceptability to readers. There is also the issue of adequacy once the users have learnt to accept a hypertext system. For people seeking more information (say, after an initial exposure to a tourist information kiosk), the restrictive design of the application may pose a limitation (say, the tourist would like to know, immediately, about some good restaurants near a place of historic interest) - in fact, the display may not be large enough to display more details without chunking information. An overview mechanism (for browsers) and a query mechanism (for information seekers) may be more appropriate in such a situation.

3. Adaptability by readers for the task in hand. Hypertext systems have to be appropriate to the tasks that the users are trying to accomplish. There should be facilities for annotation and placing bookmarks. Hypertext interchange (the compatibility of file formats) is an important criterion of adaptability and hence acceptability. With the increasing use of hypertexts for collaboration, their evaluation should include the assessment of benefits to the group as a whole as well as the individuals.

4. Skills of the readers as information users. Hypertext evaluation should include an evaluation of the demand a system may make on the reading skills of the users. This could mean the evaluation of users reading linear text (on print). The system should extend the range of cognitive activities that readers will engage in (that is, provide an insight into strategies which they may not have thought of). The following could be some cognitive extensions :

5. Ability to search using keywords (general IR). Changing the very nature of the work being done by creating some sort of an interdependency between the system and the users.

6. Evolution of new ways of thinking with the continued use of hypertext.

7. Costs of production and dissemination. The cost effectiveness of having information in hypertext form should be considered. Conventional IR systems allow only retrieval while hypertext systems help users integrate units of information retrieved.

Evaluation should also include looking for the availability of automated tools or cognitive prostheses to overcome cognitive overhead and disorientation. Readers adapt different navigation strategies based on the complexity of the task at hand. Such tools might help with a variety of subtasks related to hypertext usage. These subtasks might include specifying search targets and planning the order in which information will be sought. Tools are also required to store and manipulate the information found in order to integrate with other work. However, the effectiveness of such tools might be reduced due to the following :

1. Readers may not realize they need help.
2. Readers may not know how to use such tools.
3. Readers might blindly follow certain procedures without understanding their need.

Hence, the design and evaluation of hypertext systems should include the understanding of how people use their cognitive resources to handle information.

Nielsen also has discussed four categories for the evaluation of hypertext documents (Nielsen, 1991). These are:

a. Utility. This is a measure of whether the hypertext document actually helps a user perform the intended task. This has to be compared with performing the same tasks with linear text. Many empirical tests have shown that readers exhibit poorer performance with hypertext than with paper documents.

b. Integrity. This is a measure of the completeness of the document - whether it is up to date and not misleading and if it is easy to maintain.

c. Usability. Usability can be measured as a combination of these factors:

1. ***Ease of learning*** - It is a measure of how fast a reader can start learning and navigating through a hypertext document. Good presentation structure and graphic design are key determinants.
2. ***Ease of use*** - It is a measure of how fast a reader can locate information. Appropriately defined links, search mechanisms, backtracks, landmarks and other navigational aids can greatly increase efficiency of use.
3. ***Error handling*** - It is a measure of how many errors a reader makes and how easy it is for them to recover from such errors.

d. Aesthetics. This is a measure of how pleasing is the system to the user.

The results of the SHAPE (Southampton Hypermedia Authoring Paradigm for Education) research project suggest that there are four important issues that arise which can influence both the quality of the application and the quality of the development process (Mendes and Hall 97a), (Mendes and Hall 97b):

i. Development effort.
ii. Reusability of information.
iii. Maintainability of the application.
iv. Contents.

The conceptual framework proposed by Fenton and Pfleeger (Fenton and Pfleeger 96) can be applied to the diverse software-measurement activities that contribute to an organization's software practices. The practices can be not only the development and maintenance activities but also any experiments and case studies performed in order to investigate new techniques and tools. It is based on three principles:

i. Classifying the entities to be examined.
ii. Determining relevant measurement goals.
iii. Identifying the level of maturity that an organization has reached.

The entities to be considered for evaluation are (Mendes 98):

i. Application.
ii. Tool.
iii. Maintenance.
iv. Reuse.
v. Authors.

To measure the Maintainability and Reusability of a hypermedia application, the following independent variables can be used (Mendes, 98):

i. Size of the application.
ii. Connectivity.
iii. Structure (topology).
iv. Compactness (Botafogo et al. 92), (Vocht 94).
v. Stratum (Botafogo et al. 92), (Vocht 94).

Compactness indicates "the intrinsic connectedness of the hypertext" (20,54), and stratum reveals "to what degree the hypertext is organized so that some nodes must be read before the others" (Botafogo et al. 92), (Vocht 94).

To measure the entity Tool the following independent variables can be used (Mendes 98):

i. Link representation.

ii. Link type.

iii. Highlighting of anchors.

Link Representation means whether the links are "embedded" within the document or not. The highlighting of anchors refers to whether or not the anchors are explicitly presented to the readers (using a different color, for example). Our hypothesis is that being able to see an anchor can influence both the maintenance and reuse of the corresponding link.

To measure the entity Author the following independent variables have been proposed (Mendes 98):

i. Role.

ii. Experience.

To measure the entities Maintenance of the application and Reuse of information the following dependent variables have been used (Mendes 98):

i. Time.

ii. Difficulty.

The relevant measurement goals can be determined using the Goal-Question-Metric (GQM) approach (Basili et al. 94), which is based upon the assumption that any measurement must be defined in a top-down fashion. The result of applying the GQM approach is a model that has three levels: i) The conceptual level - Goal; ii) The operational level - Question; and iii) The quantitative level - Metric. The goal is refined into several questions and each question is then refined into metrics, either objective or subjective. SHAPE's corresponding GQM is presented below (Mendes 98):

Goal: To evaluate the quality of the hypermedia application, from the authors' viewpoint

Question: What is the quality of the hypermedia application?

Metrics: Maintainability, reusability

Goal: To improve the maintenance of hypermedia applications and the reuse of information

Question: What is the influence of the tool on the maintainability/ reusability?

Metrics: Highlighting of Anchors, representation of links, type of links

Question: What is the influence of the application on the maintainability/ reusability?

Metrics: Size of the application, connectivity, structure of the application, compactness, stratum

Question: What is the influence of the author on the maintainability/reusability?

Metrics: Role, experience

The level of maturity within the hypermedia application development community that is considered for this case is either level 1 or 2. Level 1 typically means that an organization does not provide a stable environment for developing and maintaining software. Level 2 means that there are policies for managing a software project and procedures to implement those policies are established.

Information Retrieval Issues

Navigation or browsing is effective only for small hypertext systems. For large hypertext databases, information retrieval (IR) through queries becomes crucial. Conklin had suggested that search and query mechanisms can present information at a manageable level of complexity and detail (Conklin, 1987).

Conventional IR systems focus on keyword based automatic searching (in conjunction with Boolean operations), weighting of words based on their statistical properties, ranking of documents according to probability of relevance, automatic relevance feedback for query modification and query languages (Croft et al., 1990). However, very few (or none) of these methods retrieve complete or accurate information. Too general a query may yield a lot of items and too specific a query may retrieve no items. Thus, traditional IR is an inherently uncertain process. Combining inference techniques could eliminate or minimize uncertainty. In hypertext systems, a weighted keyword search combined with hypertext links can improve IR by finding only a subset of nodes or "hits" whose links can then be followed to other semantically related nodes (Carlson, 1989).

According to Halasz (1988) query and search mechanisms can be classified into content search and structure search. Content search is standard IR technique extended to hypertext systems. That is, all nodes and links are treated independently and examined for a match to the given query. On the other hand, structure search will yield the hypertext sub-network that matches a given pattern. Query facilities which combine aspects of both content search and structure search will be capable of acting as filters. Based on the user's query, the interface will display only those nodes and links that match the query, filtering out other parts of the network.

Bruza (1990) proposed a two-level architecture for hypertext documents, the top level called hyperindex (containing index information) and the bottom level hyperbase (containing content nodes and links). The hyperindex consists of a set of indexes linked together. When an index term describing the required information is found, the objects from the underlying hyperbase are retrieved for examination. Navigating through the hyperindex (not the hyperbase) and retrieving information from the hyperbase is called "Query By Navigation" (Bruza, 1990).

Bruza's measures to determine the effectiveness of index expressions in the hyperindex include:

a. ***Precision***: The ratio of relevant objects associated with the descriptor to the total number of objects associated with the descriptor.

b. ***Recall***: The ratio of the number of objects associated with the descriptor to the total number of relevant objects.

c. ***Exhaustivity***: The degree to which the contents of the objects are reflected in the index expressions.

d. ***Power***: The ratio of a descriptor's specificity to its length.

e. ***Eliminability***: The ability to determine the irrelevance of a descriptor and stop the search.

f. ***Clarity***: The ability to grasp the intended meaning of the descriptor.

g. ***Predictability***: The ability to predict where relevant descriptors can be found in the index.

h. ***Collocation***: The extent to which the relevant index terms are near each other in the index.

Experiments and empirical studies are still required to determine these retrieval measures for hypertext-based IR systems.

Beeri and Kornatzky(1990) have suggested a logical query language that would allow structural queries over a hypertext network.

Indexes can also be considered as "precompiled links", providing immediate access to required information without navigating through the "document space" (Frisse & Cousins, 1989). Researchers have suggested using the vector space model to organize a hypertext collection into clustered hierarchies (Crouch et al., 1989). In this model, the content of each node or document is represented by a set of possibly weighted terms. Thus, each document can be represented by a term vector and the complete document collection can be represented by a vector space whose dimension is equal to the number of distinct terms to identify the documents in the collection.

The object-oriented concept of abstraction (generalization/aggregation) will greatly benefit IR in hypertext systems (Botafogo & Shneiderman, 1991). Abstraction is the concealment of all but relevant properties of an object or concept. Aggregation is the clustering of related objects to form a higher level object. Generalization is the property of treating a set of similar objects as a generic object. These two concepts of abstraction can be effectively used to simplify hypertext structures. A set of related nodes and the links between them can be treated as a semantic cluster (as opposed to the hierarchical cluster proposed by Crouch et al. (1989) having the following properties:

a. They form a subgraph of the hypertext graph.

b. The compactness (the degree of interconnectedness of the hypertext) of the subgraph is higher than the compactness of the whole graph.

Graph theoretical algorithms such as biconnected components and strongly connected components can be applied in the formation of such clusters or aggregates. A similar approach called Aggregation Clustering With Exceptions (ACE) has been proposed to identify aggregates or clusters and exceptions (those that do not fall into clusters) (Hara et al., 1991). Such clustering mechanisms facilitate effective IR from hypertext systems.

Lucarella has written that whereas conventional IR techniques focus on "what to where" (we know what we want, but we wish to find out where in the database it is), hypertext browsers focus on "where to what" (we know where we are, but we want to know what is there) (Lucarella, 1990). IR in a hypertext system can combine these two techniques to greatly enhance the process of finding relevant information. Hypertext browsing can supplement conventional IR by allowing users to discover retrieval cues that successively can be used for query formulation. Query facilities can supplement hypertext browsing by providing the user with a set of relevant nodes for browsing.

Integration Issues

Most current hypertext systems are closed systems - material created in one system cannot be transferred or integrated with material created in another system because of proprietary document formats and storage mechanisms. Conversion programs are difficult to write since the formats are not disclosed by organizations (Fountain et al., 1990).

Bieber (1993) has suggested the following front-end and back-end requirements for system-level approaches to hypermedia integration or client/engine cooperation.

In order for the hypermedia engine to provide functionality such as management of link markers, comments, trails, overviews, filters, forward navigation and backtracking, the front-end should provide the following:

1. Tracking the location of objects such as link markers and providing their identifiers to the engine when a link marker is selected.

2. Front-end must request from the engine editing permissions for insertions, deletions, and modifications.

3. User interface must provide hypermedia prompts.
4. When the front-end saves a document with embedded hypermedia objects, the objects should also be saved.

The hypermedia engine would provide functionality such as linking, annotation, backtracking, filtering, and overviews on behalf of the back-end. The back-end should provide the following functionality:

1. Provide specific information about its structure and its applications' documents.
2. Bridge laws must be written by developers. This could be done through a bridge law editor instead of writing predicate logic.
3. Back-ends should provide control information and interpretive mechanisms along with the objects that are sent through messages. For example, objects that have to serve as link markers could be tagged.
4. Back-ends should support hypermedia engine commands same as the front-end (command lists and context sensitive information).
5. Back-end should incorporate a standard document interchange standard such as ODA or SGML.

Some of the issues involved in developing an open hypertext system include the following (Pearl, 1989):

1. The User Interface for the creation and management of links should be consistent with the editors provided by the individual applications.
2. Since the Link Service and the applications are separate processes decisions must be made about sharing/dividing the responsibility for exception handling and user dialogs.
3. The Link Service should detect and remove dangling links, either implicitly or explicitly. Implicit removal can happen when a user tries to follow a link from its valid end to its invalid end by suggesting to the user to remove the link. Explicit removal can happen, through a link garbage collection mechanism, by tracking links, validating nodes and removing invalid links.
4. While versioning of data objects can be left to the individual applications, the Link Service must still handle the versioning of links. However, the consistency of a versioned hypertext cannot be guaranteed if nodes are versioned separately from links.
5. Unstructured documents such as ASCII files cannot be handled elegantly since they are not uniquely indexed nor do they carry semantics.
6. The issue of traversing links across networks and locating objects located at remote sites is very important due to performance and cost factors. Also, decisions have to be made about where on the network should the Link Service process be located. Another related issue is the invocation of an application which may not be currently running although the user is following a link to a node managed by that application.

Whereas most models have focused on design metaphors and implementation abstractions, very little work has been in the area of a general framework for hypertext functionality. Rao and Turoff

(1990) observed that "Hypertext should be treated as a general purpose tool with approaches to handling nodes, links, and retrieval, that fits within the context of any application and conveys common meanings to users. To accomplish this, we need a comprehensive framework for hypertext based on a cognitive model that allows for the representation of the complete range of human intellectual abilities."

Document Standards

Unlike linear documents which are static, generic, and structured, hypertext documents are unstructured and can be dynamic. Hence, current structured document standards are not sufficient to represent hypertext networks. A tree based hierarchy is relevant but NOT sufficient for hypertext. There should be a hierarchical framework with a system of typed links to cover the cross-references of structured documents and the links of hypertext. The current forms of ODA and SGML are not sufficient enough for the representation and exchange) of hypertext. These need extensions to provide a proper typed-link mechanism. SGML does not specify layout or presentation information (which is important for hypertext) or how to handle images and graphics. ODA does address these issues but it is not sufficient.

At the same time, a single standard may not be enough due to the diversity of usage of hypertext applications - large volume hypertext systems are different from highly interactive systems. While the former require highly efficient search capabilities (and standards), the latter require better individualized responses and navigational tools. Hence, these two may require different standards.

Limitations of SGML (Balasubramanian, 1994):

1. SGML allows cross-referencing within the same document. This can be done by assigning unique identifiers to elements that need to be referred to elsewhere. However, the uniqueness of the identifier (and hence, the element) is applicable only within the current local document. Hence, only elements within the same document (and only those having unique identifiers) can be linked. Therefore, this mechanism can only be used in a hypertext document to refer to elements within the same document and not other documents.

2. SGML cannot support time dependent data such as audio and video and also graphics and images. Rendering of events is not possible in SGML, that is, displaying a map of NY and a link that zooms into Manhattan.

ODA, a standard for the storage and interchange of multimedia documents, deals with both logical structure and layout structure or presentation (unlike SGML). ODA currently includes graphics and images and extensions are being considered to handle audio, video, and hypertext (Cole & Brown,1990). Limitations of ODA and possible modifications are described below (Balasubramanian, 1994):

a. Separation of logical structure and layout structure. Though ODA supports both logical structure and layout structure, they are not completely separated. In order to change the style of a document the logical structure must be edited since the layout process uses the logical structure, the generic structures and the content architectures to create the specific layout. This limitation can be eliminated by carrying over the SGML mechanism of applying different set of layout and presentation styles (or style sheets) for different views of the same logical document.

b. Comprehensive attribute inheritance. The ODA mechanism for inheriting layout attributes (such as placement of blocks of contents within pages and rectangular areas called frames) and presentation attributes (such as character sets and the placement of items within blocks) is not sufficient. If an attribute value is not specified for the object or its class, then the value can only be inherited according to the object's position in the tree and not according to its class (chapter, list etc.). Attribute inheritance can be achieved by adding a facility called "style tables" which will enable the style inherited by an object (and hence its format) to depend both on its class and is position in the document. This will be very valuable for hypertext in order to distinguish between objects of the same type that have different status (such as open and close buttons). It can also be extended to specify changes of state (for example, when selecting a hotspot) by changing the style table.

c. Links. ODA does not have the ability to specify the purpose of a link and also how the layout process can express that purpose. This can be accomplished by having classes for links (just as there are classes for logical objects). The class of the link will determine how and where in the document the link can be used. Thus, the representation of the link will depend on both the class and its position in the document.

d. Selective and multiple presentation. ODA does not have the ability to suppress the appearance of a logical object (or contents) during the layout process nor the ability to present the object many times. Such a feature will be of great help in a hypertext document where a reviewer's comments can be suppressed from appearing in a printout or different versions of the same basic document can be produced for various purposes. This can also be accomplished by the usage of style tables suggested earlier.

e. Complete interactivity. The ODA layout process is sequential and page based and hence does not provide complete interactivity. It does not support online editing capabilities such as the ability to scroll through a document, the ability to display selected items (outlining facility), the ability to popup additional information on demand (such as footnotes, glossaries etc.), the ability to "fold" documents revealing hidden sections only on request, the ability to follow links automatically. Complete interactivity would require extensions. Outlining can be done by having style tables that select objects by class and required level. Popup displays can be arranged by changing to a different style table and returning to the original table after the popup information has been displayed. Similarly, folding can be achieved indirectly through popups and popdowns. Link traversal can be done by replacing the current object with the target object or displaying the target object as a temporary popup item. A style table can be used to specify whether or not to display the linked object.

The Hypermedia/Time-based Structuring Language or **HyTime** is an International Standard for representing hypertext links, and synchronization of static and time-based information contained in multiple conventional and multimedia documents and information objects (SIGLINK, 1992). It addresses the limitations of SGML (Newcomb et al., 1991). HyTime supports cross-referencing facilities to uniquely identified elements in external documents. It also extends SGML's reference capability to accommodate elements with no unique identifiers in the same document. It provides pointers or location addressing schemes that contain the necessary information in order to locate cross-referenced data. It is independent of data content notations, link types, processing, presentation, and semantics. HyTime supports addressing by name, by position in the document, and by semantic construct. Links can be established to documents that conform to HyTime as well as those that do not.

HyTime allows all kinds of multimedia and hypertext technologies (whether proprietary or not) to be combined in any information product. It addresses only the issue of interchange of hypermedia information and not the standardization of presentation (same as SGML), user interfaces, query languages etc. Objects in a HyTime hypertext document can include formatted and unformatted documents, audio and video segments, still images, animations, and graphics.

HyTime is an SGML application conforming to ISO 8879. It provides the notion of "Architectural Form" to SGML. An architectural form is a syntax template around which a document author can build semantic constructs for linking and coordinate space addressing. It is highly flexible and extensible. The interchange format can be defined in Abstract Syntax Notation 1 (ISO 8824) and can be encoded according to the basic encoding rules of ISO 8825 for interchange using protocols conforming to the OSI model. The full set of HyTime functionality supports "integrated open hypermedia", the "bibliographic model" of hyperlinking that allows links to anything, anywhere, anytime.

HyTime is intended for use as the infrastructure of platform-independent information exchange for hypermedia and synchronized and non-synchronized multimedia applications. Application developers will use HyTime constructs to design their information structures and objects and the HyTime language to represent them for interchange (SIGLINK, 1992).

CCITT has proposed the future international standard for multimedia and hypermedia information objects, also known as the **MHEG** Standard. "The scope of the MHEG standard is to define the representation and encoding of multimedia and hypermedia information objects that will be interchanged as a whole within or across applications or services, by any means of interchange including storage devices, telecommunications or broadcast networks." (CCITT, 1992). The initial objectives of the MHEG standard include meeting the following requirements:

1. Provide abstractions for real-time presentation including multimedia synchronization and interactivity.
2. Provide abstractions for real-time interchange with minimal buffering using normal speed data communications.
3. Provide abstractions for direct manipulation of information without any additional processing.
4. Provide linking facilities between elements of composite multimedia objects.

The main MHEG classes include: Content Class, Selection Class and Modification Class, Link Class, Script Class, Composite Class, and Description Class. The objects play a federating role, enabling different applications to share the basic information resource. These objects can be encoded using ASN.1 or SGML and will provide a common base for other CCITT recommendations, ISO and other standards, user defined architectures and applications.

In order to make hypertext systems fully portable, existing document standards such as ODA and SGML must be extended to support unstructured documents and linking. International standards such as HyTime and MHEG are emerging to support hypertext functionality and multimedia information in applications. Only when hypertext functionality becomes an integral part of our computing environment will knowledge workers accept and incorporate hypertext into their daily work process.

Hyper-G: The next generation hypertext system

Andrews, K. Kappe, F. Maurer, H. Schmaranz, K., (1994)concentrated on features they find desirable in second generation hypermedia information systems. They have compared the features found in the first generation hypermedia system WWW with those found in what might be the first second generation hypermedia system, Hyper-G.

In Hyper-G the setting is considerably more general: chunks, called "clusters" in Hyper-G terminology consist of a number of documents. A typical cluster may, for example, consist of five documents: a piece of text (potentially with inline images), a second piece of text (for example in another language, or a different version of the same text, or an entirley different text), a third piece of text (the same text in a third language perhaps), an image and a film clip. Anchors can be attached to textual information, to parts of images, and even to regions in a film clip. Links are not part of the document but are stored in a separate database. They are both typed and bidirectional: they can be followed forward (as in WWW) but also backwards. The textual component of a document is usually stored in so-called HTF format, also a derivative of SGML, but can also be stored as a PostScript file.

The support for multiple pieces of text within a cluster allows Hyper-G to handle multiple languages in a very natural way. It also elegantly handles the case where a document comes in two versions: a more technical (or advanced) one and one more suitable for the novice reader. As indicated, pictures can be treated as inline images or as separate documents. Often, inline images are convenient, since the "author" can define where the user will find a picture in relation to the text. On the other hand, with screen resolution varying tremendously, the rescaling of inline images may pose a problem: if a picture is treated as separate document, however, it appears in a separate window, can be manipulated (shifted, put in the background, kept on-screen while continuing with other information, etc.) independent of the textual portion (which in itself can be manipulated by for example narrowing or widening its window). Thus, the potential to deal with textual and pictorial information separately provides more flexibility when required. Text can be stored in Hyper-G not only in HTF, but also in PostScript format. Since most printers are geared towards printing PostScript files, almost all word processing packages are capable of producing PostScript files as output: thus, all word processing packages can be used to prepare data for Hyper-G using this approach. Also, PostScript files allow the incorporation of pictures and formulae; they offer the user the possibility to view documents exactly as if they were printed (given high enough resolution screens), and such Hyper-G documents can be printed with the usual professional PostScript quality. For one of the major applications of Hyper-G, the Journal of Universal Computer Science (see Section 4), a full-text search engine has been implemented for PostScript files as well as full hyperlinking facilities. The use of standard compression techniques allows the PostScript files to be compacted to about the same size as equivalent HTF and HTML files. Thus, the use of PostScript with high-quality PostScript viewers built into the native Hyper-G clients Amadeus and Harmony (for MS-Windows and X Windows respectively), gives Hyper-G the necessary professionalism for high quality electronic publishing of journals, books, and manuals.

In addition to the "usual" types of documents found in any modern hypermedia system, Hyper-G also supports 3D objects and scenes. The native X Windows client for Hyper-G (Harmony) provides four different ways to navigate within such 3D models. Finally, Hyper-G allows the use of documents of a "generic" type. This permits future extensions and the encapsulation of documents otherwise incompatible with Hyper-G.

There is considerable difference in the philosophy of links in WWW versus Hyper-G. The ability to attach links to parts of a picture is clearly desirable, when additional information is to be associated with certain sub-areas of an image. That links are bi-directional and not embedded in the document has a number of very important consequences: first, links relating to a document can be modified without necessarily having access rights to the document itself. Thus, private links and a certain amount of customization are possible; second, when viewing a document it is possible to find all documents referring to the current one. This is not only a desirable feature as such, but is of crucial importance for being able to maintain the database. After all, when a document is deleted or modified, all documents referring to it may have to be modified to avoid the "dangling link syndrome", or to avoid being directed to completely irrelevant documents. Hyper-G offers the possibility of automatically notifying the owner of a document that some of the documents that are being referred to have been changed or deleted, an important step to "automatic link maintenance". Thirdly, the bi-directionality of the links allows the graphic display of a "local map" showing the current document and all documents pointing to it and being pointed at, an arbitrary number of levels deep. Harmony makes full use of this fact and provides local maps as an invaluable navigational aid that cannot be made available for WWW databases ((Andrews et al 1994), (Fenn et al 1994)). Finally, the fact that links can have types can be used to show to the user that a link just leads to a footnote, or to a picture, or to a film clip, or is a counter- or supporting argument of some claim at issue: typed links enhance the perception of how things are related and can be used as tool for discussions and collaborative work.

Navigation in WWW is performed solely using the hypertext paradigm of anchors and links. It has become a well accepted fact that structuring large amounts of data using only hyperlinks such that users don't get "lost in hyperspace" is difficult to say the least. WWW databases are large, flat networks of chunks of data and resemble more an impenetrable maze than well-structured information. Indeed every WWW database acknowledges this fact tacitly, by preparing pages that look like menus in a hierarchically structured database: items are listed in an orderly fashion, each with an anchor leading to a subchapter (subdirectory). If links in WWW had types, such links could be distinguished from others. But as it is, all links look the same: whether they are "continue" links, "hierarchical" links, "referential" links, "footnote links", or whatever else.

In Hyper-G not only can have links a type, links are by no means the only way to access information. Clusters of documents can be grouped into collections, and collections again into collections in a pseudo-hierarchical fashion, since the collection structure is not a tree, but a DAG (Directed Acyclic Graph). The collection "hierarchy" is a powerful way of introducing structure into the database. Indeed many links can be avoided this way (Maurer et al 1994), making the system much more transparent for the user and allowing a more modular approach to systems creation and maintenance. Collections, clusters and documents have titles and attributes. These may be used in Boolean queries to find documents of current interest. Finally, Hyper-G provides sophisticated full-text search facilities. Most importantly, the scope of any of such searches can be defined to be the union of arbitrary collections, even if the collections reside on different servers.

Some WWW applications also permit full-text searches. However, no full-text search engine is built into WWW. Thus, the functionality of full text search is bolted "on top" of WWW: adding functionality on top of WWW leads to the fragmentation of WWW, since different sites will implement missing functionality in different ways. Thus, to stick to the example of the full text search engine, the fuzzy search employed by organization X may yield entirely different results from the fuzzy search employed by organization Y, much to the bewilderment of users. Actually, the situation concerning searches in WWW is even more serious: since documents in WWW don't have attributes, no search is possible on such attributes; even if such a search or a full text search is artificially implemented, it is not possible to allow users to define the scope for the search, due to the lack of structure in the WWW database. Hence full-text searches in WWW always work in a fixed, designated part of the WWW database residing on one particular server.

Thousands of WWW servers are currently installed, and are accessible via the Internet. However, there is no coherence between the servers: if a user wants to search for an item in a number of WWW servers the user has to initiate a new search for each server. This problem is compounded by the fact that WWW knows only two types of access rights: everything allowed (webmaster) or read-only access. Neither are there shades in between, nor is it possible to allow certain users to edit some parts of a WWW server, other users to edit other parts.

In contrast Hyper-G provides various types of access rights and the definition of arbitrarily overlapping user groups. Hyper-G is also a genuine distributed database: servers (independent of geographical location) can be grouped into collections, with the hyper-root at the very "top". Thus, users can define the scope of searches by defining arbitrary sets of collections on arbitrary servers. Different groups can work with the same server without fear of interfering with someone else's data. To be more concrete, suppose 10 departments within a university intend to operate a WWW database. If they operate one server together and if all want to input their own data, the data of department X is not protected from any kind of access or change by department Y! Hence the tendency would be to operate 10 different servers. (Indeed, there are many more WWW servers than there are server sites, clearly demonstrating this phenomenon.) However, if the 10 departments operate 10 different servers and a user from outside wants to look up a person without knowing the department, the user is forced to query all of the servers, one after the other.

Hyper-G, being a distributed database with well-defined access rights of fine granularity, offers a much more satisfactory solution: the 10 departments operate a single server, different users have different access rights: not only can department X not influence the information of department Y, certain parts of the database may have even their read access restricted to certain groups or even to single individuals ("private collections"). Hyper-G may be used anonymously, but if users identify themselves they will automatically be shown their "home collection", where they have collected the information most important for them, and from where they can enter all those parts of the database to which access is permitted for them. Continuing the earlier example, suppose an outside user looks for a certain person. Accessing the single Hyper-G server operated by the 10 departments with a full-text search will find the information, assuming it is present. However, suppose the departments insist on each operating their own Hyper-G server: by simply defining a collection "servers of this university", a single Hyper-G search will still examine all of the 10 databases (assuming they are in a LAN or on the Internet). Even now, the servers offer possibilities not available without proper access control: members of the departments may keep some information just for themselves, or for a group they collaborate with, etc.

If one has access to a local Hyper-G server, all accesses to other Hyper-G servers, but also WWW, Gopher and WAIS sites are routed through the local Hyper-G server. Documents retrieved once are automatically cached (for all users of that server), so they will no longer be retrieved from the remote database next time around. Using the separate link database, it can be assured that new versions of a cached document are automatically retrieved when a request for accessing the document is issued. Although recent WWW servers also support caching, the consistency of cached documents cannot be guaranteed. As we will discuss in (Section 4), caching in Hyper-G applies equally to documents from non-Hyper-G servers. Hence, using a local Hyper-G server may be quite valuable, even if that server is used for nothing much beyond caching!

The acceptance of a hypermedia system is certainly not only dependent on deep technical features, but above all on the information content and the ease of use. Due to the fact that large hypermedia systems tend to lead to disorientation, second generation hypermedia systems have to try very hard, both at the server and at the client end, to help users with navigational tools. Some navigational tools, like the structuring and search facilities have already been described; others, such as maps, history lists, specific and personal collections can also be of great help and are available in Hyper-G; a particular specialty of the Harmony client (assuming an OpenGL environment) is a 3D browser: the "information landscape" depicts collections and documents (according to their size) as blocks of varying size spread out across a three-dimensional landscape, over which the user is able to fly.

First generation hypermedia systems like WWW have traditionally been seen mainly as (simple) information systems. Most applications currently visible support this view: very often WWW servers offer some pleasantly designed general information on the server-institution, but only rarely does the information go much deeper. If it does, usually a "hybrid" system is used, WWW with some add-ons using the scripting interface of WWW.

Andrews, K. Kappe, F. Maurer, H. Schmaranz, K., (1994) believe that hypermedia systems acting as simple information systems, where someone inputs information to be read by other users, do not offer much potential: they will disappear into obscurity sooner rather than later. To ensure the success of a hypermedia system, it must allow users also to act as authors, allow them to change the database, create new entries for themselves or other users, create a personal view of the database as they need it, and, above all, allow the system to be used also for communication and cooperation.

First generation hypermedia systems like WWW almost entirely lack support for such features. Emerging second generation hypermedia systems are bound to incorporate more and more features of the kind mentioned; Hyper-G provides a start.

The native Hyper-G clients Amadeus and Harmony are designed to allow the easy import of data into the server. They are also designed to allow point-and-click link generation: select the source anchor location with a mouse-click, select the destination anchor with a mouse-click and confirm that a link should be created.

Hyper-G supports annotations (with user-definable access rights): in contrast to some WWW clients which also allow annotations which are then kept locally, Hyper-G annotations become part of the database, i.e. are also available when working with other clients, or from another user account or machine. Annotations can themselves be annotated; the network of annotations can be graphically displayed using the local map function of Harmony. Thus, the annotation mechanism can be used as the basis of (asynchronous) computer-conferencing, and has been successfully employed in this fashion. The client-server protocol in WWW is "static" in the sense that the server can only react to requests by the client, but cannot become active itself. In Hyper-G the client-server protocol is "active" in the sense that the server can contact the client: this can be used for example to send update notification to a client, and provides the first (rudimentary) possibilities for client-client communication for synchronous communication, conferencing and cooperation.

Many of the features discussed in the area of computer supported cooperative work (Devan 1993) is planned to be incorporated into Hyper-G. Some of the most widely used functions of the Internet are file transfer (FTP) and electronic mail. Hence, second generation hypermedia systems have to support both FTP and particularly email. Without leaving their hypermedia environment, users must be able to edit, send, and receive email. Email should automatically be presorted by criteria such as subject, author, date, etc. Related pieces of email can be linked together, the local map feature presenting a good graphical overview of the flow of the email discussion pertaining to a certain subject. The hypermedia system should also have the possibility to send mail with delays or on certain dates to act as reminder and as an active personal scheduler. A number of relevant ideas are collected in (Kappe et al 1993b) and are currently under implementation for Hyper-G.

Hyper-G is a first attempt to offer much more basic functionality, yet to continue the path started by WWW and remain fully interoperable with WWW: every WWW client can be used to access every Hyper-G server, albeit occasionally with some loss in functionality; a Hyper-G client may, through a Hyper-G server, access WWW, Gopher, and WAIS servers without any loss of functionality, indeed providing "free" caching as a by-product.

The compatibility of Hyper-G with WWW and Gopher actually goes much further: tools to import complete WWW and Gopher databases into Hyper-G without manual intervention are in preparation. Thus, users of WWW can migrate up to an environment allowing all kinds of searches, access control, etc., without being forced to abandon their current database or their favorite WWW client.

Hyper-G is the most powerful networked multimedia system currently available, and is free of charge for all educational institutions. The source code of the major clients (with the exception of some proprietary segments) is or will be available for developers.

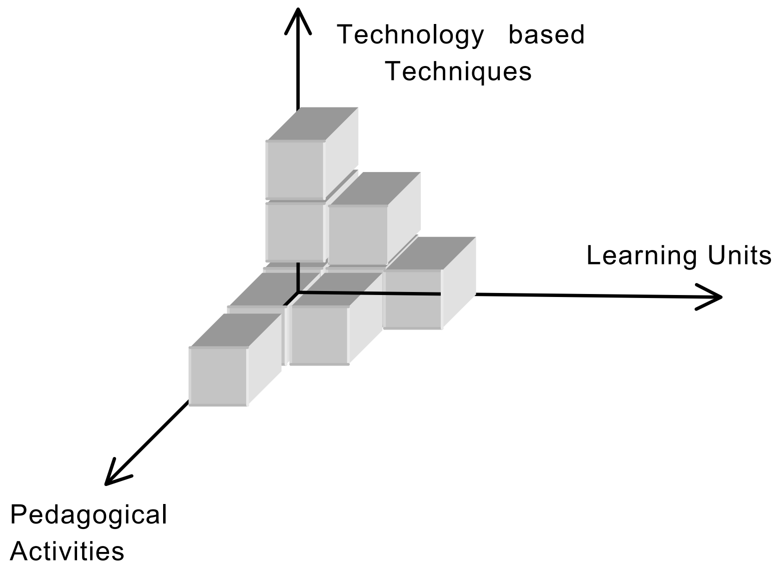
Due to its functionality, Hyper-G is used for a wide variety of applications:

- As a basis for university information systems (with substantial information content at Graz University of Technology and The University of Auckland, and in experimental use at a number of other universities)
- As an organization-wide information and communication system (the European Space Agency, ESA, being the largest user so far)
- As a multimedia infrastructure for museums and exhibitions (MONZ, the new Museum Of New Zealand, the Interactive Information Center in Styria, and the AEIOU project as three major examples)
- As an infrastructure for tele-teaching experiments (the University of Auckland in cooperation with NZ Telecom as a first test-site)
- As an infrastructure for electronic publishing (with a German publishing consortium including Meyer/Brockhaus/BI/Langenscheidt for reference books, and Springer in connection with the new Journal of Universal Computer Science with a backbone of 65 Hyper-G servers as examples)
- As an infrastructure for the cooperation amongst mathematicians in Germany (as recommended by the "Deutsche Mathematiker Vereinigung")

A proposed structure for the Intellect Course

The explored hypermedia - hypelinking issues will be taken into consideration in the development of the Base Level 1 of the Intellect Course. A proposed development action plan, must be defined on a three dimensional space, defined by the following axes:

1. Learning Space
2. Pedagogical Activities
3. Technology / Techniques



Learning Units

Learning Units can be identified on the basis of the existing educational material and should correspond to sections and subsection which serve a specific learning objective. To convey the knowledge of each learning unit, a set of appropriate peadagogical activities must be defined (from the list that follows), and appropriate techniques must be utilised. Not all combination of pedagogical activities, hypertext techniques are technically feasible or educationally fitted for each learning unit. The selection process which will identify, which blocks of the grid in Figure 1 will be filled in order to arrive at a value-added version of the original material.

Pedacogical activities

Hypertext structures can enhance the value of educational material by providing in addition to knowledge, an interactive mechanism between the student and the educational material, which can support different pedagogical activities, which we believe that lead to enhancing knowledge.

Hypertext educational material can thus be organised along various pedagogical activities, which may include:

1. Presentation of Knowledge This can be done by providing: a. Definitions of Concepts. b. Textual and Graphical / Pictorial Descriptions on specific educational elements.	2. Examples Examples illustrate in various forms (text, graphics, animation, video etc.) the educational information which was initially provided in conceptual form and could be: a. Concept examples, usually of short length

c. References to other sources of educational information, especially in the form of direct links.	b. Case studies, which are longer and assist in a better comprehension and integration of knowledge.
3. Exercises Exercises help the student to apply the knowledge acquired through the previous elements (1 and 2). Exercises could provide links to specific learning items contained in (1) and (2), thus reinforcing learning and enhancing the learning process. Exercises can be provided in different forms such as: a. Multiple choice b. Free form question / answer c. Practice exercises aiming in applying certain methodologies and procedures d. Filling the Gap e. Group Exercises	4. Self - Tests Interactive hypertext provides an appropriate media for developing self - tests as elements of the learning process. Self - tests have the same structure as exercises but the aim is obvious different. No interaction with learning items (1) and (2) is provided. Instead, the features of the interaction mode between the learner and the system must include time limits on completing a test, grading mechanisms and appropriate feedback to the learner.
5. Communication Today's hypertext environments implemented on the WEB provide the means for easy communication between the main actors in the learning process: teachers and students. Communication elements could include: a. Bulletin boards containing announcements regarding the module b. E-mail facilities between tutors and students c. Establishment of chat rooms and discussion groups for reviewing various issues and providing communications between students.	

Techniques

Examples of hypertext techniques include animation, exploring hypertext links, video etc. It is obvious that not all techniques are efficient for each pedagogical activity, some could even be totally inappropriate. For example an animation technique could be used to demonstrate the searching mechanism in a database and to provide the means of simulation of data retrieval.

For the purpose of INTELLECT the following techniques could be explored in order to examine the appropriateness of each technique for specific pedagogical activities of each learning unit:

- Hypertext Links (internal and external): For long term knowledge transfer a relational structure can create an interactive environment that permits "repurposing" of existing materials, and offer directions to activities appropriate for the learner's task and level of expertise. A hierarchical structure can be used on the other hand for searching browsing and information retrieval
- Use of Metaphors to create a more accessible interface
- Navigation techniques include access to facilities, such as maps, an index of keywords, a guided tour etc.
- Mapping techniques can include site maps to visualise the document structure and reduce disorientation, and reorientation tools for nor "being lost in hyperspace"
- Easy communication w/ tutor or fellow students
- Simulation and Animation
- Video

The identification of the relationships between these trhee variables (learning units, Pedagogical Activities and Techniques) could lead to the following matrix:

Learning Units	**Pedagogical Activities**	**Techniques:** Hypertext Links	•• •	Animation	•• •	Video	• •
•							
Retrieval	•	• • • •	• •	• • •	• •	• •	• •
	•						
Normalisation	Concepts / Methodologies	*Links to basic definitions of database concepts in previous learning units* *Links to other sources i.e. research papers on internet*	• •		• •	*Experts opinion on normalisation procedure*	• •
	Examples	*Links to concepts / definitions in this unit*	• •	*Illustrate the 3 stages of the normalisation process*	• •		• •
	Exercises	*Links to examples* *Links to concepts / definitions*	• •	*Simulation of results*	• •		• •
	Self - Tests	*Link to knowledge base for providing evaluation of results*	• •		• •		• •
	Communication	*Links to tutor e-mail*	• •		• •		• •
•	• • • •	• • • •	• •	• • •	• •	• •	• •
•							

References

Andrews, K. Kappe, F. Maurer, H. Schmaranz, K. (1994) On Second Generation Hypermedia Systems, Journal of Universal Computer Science, 0, 0, 127-135

Andrews,K., Kappe,F. (1994) Soaring Through Hyperspace: A Snapshot of Hyper-G and its Harmony Client; Proc. of Eurographics Symposium on Multimedia/Hypermedia in Open Distributed Environments, Graz.

Balasubramanian V. (1994) "State of the Art Review on Hypermedia Issues And Applications", http://www.isg.sfu.ca/~duchier/misc/hypertext_review/index.html

Basili, V. R. and Rombach, H. D. (1988) "Towards a Comprehensive Framework for Reuse: A Reuse-Enabling Software Evolution Environment"; Technical Report CS-TR-2158, Dept. of Computer Science, University of Maryland, College Park, MD 20742.

Basili, V., Caldiera G., and Rombach, D. (1994) "The Goal Question Metric Approach", Encyclopedia of Software Engineering, Wiley.

Beeri, Catriel and Kornatzky, Yoram (1990) A Logical Query Language For Hypertext Systems, Proceedings of European Conference on Hypertext, ECHT '90.

Bieber, Michael (1993) Providing Information Systems with Full Hypermedia Functionality, Proceedings of the Twenty-Sixth Hawaii International Conference on System Sciences, 1993.

Botafogo, Rodrigo A., Rivlin, Ehud, and Shneiderman, Ben (1992) "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics", ACM TOIS, 10, 2, 143-179.

Botafogo, Rodrigo and Shneiderman, Ben (1991) Identifying Aggregates in Hypertext Structures, Proceedings of Hypertext '91, ACM Press.

Briand, L., Bunse, C, Daly, J, Differding, C. (1996) "An experimental comparison of the maintainability of OO and structured design documents"; Proceedings of EASE.

Bruza, Peter D. (1990) Hyperindices: A Novel Aid For Searching in Hypermedia, Proceedings of European Conference on Hypertext, ECHT '90.

Bush, Vannevar(1945) As We May Think, The Atlantic Monthly, July.

Carlson, Patricia Ann (1989) Hypertext and Intelligent Interfaces for Text Retrieval, The Society of Text, MIT Press.

Carvalho, Ana Amılia Amorim and Dias, Paulo (1997) "Hypermedia Environment using a Case-Based Approach to Foster the Acquisition of Complex Knowledge"; Proceedings of ED-MEDIA 97, Calgary.

Catlin, Karen S., Garrett, N.L., and Launhardt, Julie A. (1991) Hypermedia Templates: An Author's Tool, Proceedings of Hypertext '91, ACM

CCITT (1992). CCITT Press Release on MHEG, December.

Cole, Fred and Brown, Heather (1990) Standards : What can Hypertext learn from paper documents ?, Proceedings of the Hypertext Standardization Workshop by NIST, Jan.

Conklin, Jeff (1987) Hypertext: An Introduction and Survey, IEEE Computer, September 1987.

Croft, W. Bruce, Belkin, Nicholas, Bruandet, Marie-France, Kuhlen, Rainer, Oren, Tim (1990) Hypertext and Information Retrieval: What are the Fundamental Concepts, Panel Discussion, Proceedings of European Conference on Hypertext, ECHT '90.

Crouch, Donald B., Crouch, Carolyn J., and Andreas, Glenn (1989) The Use of Cluster Hierarchies in Hypertext Information Retrieval, Proceedings of Hypertext '89, ACM Press.

Devan,P. (1993) A Survey of Applications of CSCW Including Some in Educational Settings; Proc. ED-MEDIA'93, Orlando, 147-152.

Engelbart, Douglas C. (1963) A Conceptual Framework for the Augmentation of Man's Intellect, Vistas In Information Handling, Volume 1, Spartan Books, Washnigton D.C..

Fenn,B., Maurer,H. (1994) Harmony on an Expanding Net; ACM Interactions 1,3, to appear.

Fenton, Norman E., and Pfleeger, Shari Lawrence (1996) "Software Metrics, A Rigorous & Practical Approach", Second Edition, PWS Publishing Company and International Thomson Computer Press.

Fountain, Andrew M., Hall, Wendy, Heath, Ian and Davis, Hugh C. (1990) MICROCOSM: An Open Model For Hypermedia With Dynamic Linking, Proceedings of ECHT '90.

Frisse, Mark E. and Cousins, Steve B. (1989) Information Retrieval From Hypertext: Update on the Dynamic Medical Handbook Project, Proceedings of Hypertext '89, ACM Press.

Garzotto, Franca, Mainetti, Luca, and Paolini, Paolo (1994) "Analysing the Quality of Hypermedia Applications: A Design-Oriented Framework", Workshop on hypermedia design and development, Edinburgh.

Garzotto, Franca, Mainetti, Luca, and Paolini, Paolo (1995) "Hypermedia Design, Analysis, and Evaluation Issues", Communications of the ACM, Special Issue on Hypermedia Design, August.

Glushko, Robert J. (1989) Transforming Text Into Hypertext For a Compact Disc Encyclopedia, Proceedings of CHI '89, ACM Press.

Halasz, Frank G. (1988) NoteCards: A Multimedia Idea Processing Environment, Interactive Multimedia edited by Ambron, Sueann and Hooper, Kristina, Microsoft Press.

Hara, Yoshinori, Keller, Arthur M., and Wiederhold, Gio (1991) Implementing Hypertext Database Relationships through Aggregations and Exceptions, Proceedings of Hypertext '91, ACM Press.

Harrison, R., Samaraweera, L. G., Dobie, M. R., and Lewis, P. H. (1995) "Estimating the quality of functional programs: an empirical investigation", Inf. Softw. Technol., 37, 12, 701-707.

Hatzimanikatis, A. E., Tsalidis, C. T., and Christodoulakis, D. (1995) "Measuring the Readability and Maintainability of Hyperdocuments"; J. of Software Maintenance, Research and Practice, 7, 77-90.

Jacobson, M. J., and Archodidou, A. (1997) "Case-based hypermedia and learning neo-Darwinian evoluationary biology: Promoting conceptual change of complex scientific knowledge", Manuscript submitted for publication.

Jacobson, M. J., and Rand J. Spiro (1995b) "Hypertext learning environments and epistemic beliefs: A preliminary investigation"; Technology-based learning environments: Psychological and educational foundations, S. Vosniadou, E. DeCorte, & Mandl (Eds.), Springer-Verlag, , 290-295.

Jacobson, M. J., Mishra, Maouri, C. P., and Kolar, C. (1996) "Learning with hypertext learning environments:Theory, design, and research"; Journal of Educational Multimedia and Hypermedia, 5, 3/4, , 239-281.

Jacobson, Michael J. (1997) "The Evolution Thematic Investigator: Research and the Design of Hypermedia Learning Environments", Proceedings of ED-MEDIA 97, Calgary.

Jacobson, Michael J., and Rand J. Spiro (1995a) "Hypertext Learning Environments, Cognitive Flexibility, and the Transfer of Complex Knowledge: An Empirical Investigation", J. Educational Computing Research, 12,4, 301-333.

Kappe,F., Maurer,H. (1993b) From Hypertext to Active Communication/Information Systems; IIG Report 363 Graz, Austria; To appear in the Journal of Microcomputer Applications.

Kappe,F., Maurer,H., Scherbakov,N. (1993a) Hyper-G -- A Universal Hypermedia System; Journal of Educational Multimedia and Hypermedia 2,1, 39-66.

Kitchenham, Barbara (1993) "DESMET METHODOLOGY: Guidelines for Evaluation Method Selection", DESMET Project Deliverable D2.3.1, The National Computing Centre Ltd..

Kitchenham, Barbara Ann (1996) "Evaluating Software Engineering Methods and Tool, Part 1: The Evaluation Context and Evaluation Methods"; Software Engineering Notes, 21, 1, 11-15.

Lennon,J., Maurer,H. (1994a) You Believe What Multimedia Is? And What Internet Will Do For You? Well...Think Again!; To appear in Datamation.

Lennon,J., Maurer,H. (1994b) Lecturing Technology: A Future With Hypermedia; Educational Technology 34,4, 5-14.

Lucarella, Dario (1990) A Model For Hypertext-Based Information Retrieval, Proceedings of the European Conference on Hypertext (ECHT) '90.

MacDonell, S. G. (1991) "Rigor in Sofware Complexity Measurement Experimentation"; J. Systems Software, 16, 141-149.

Marchionini, Gary & Shneiderman, Ben (1988) Finding Facts vs. Browsing Knowledge in Hypertext Systems, IEEE Computer, January.

Marcus, Sharon (1997) "Making the Link between Hypertext and Human Cognition", COMM 640 - Final Research Paper, Cornell University

Maurer,H. (1994) Advancing the ideas of World-Wide Web; Proc. Conf. on Open Hypermedia Systems, Honolulu, 201-203.

Maurer,H., Philpott,A., Scherbakov,N. (1994) Hypermedia Systems Without Links; Journal of Microcomputer Applications 17,4.

Mayes, Terry, Kibby, Mike & Anderson, Tony (1990) Learning about Learning from Hypertext, Designing Hypermedia for Learning, NATO ASI Series, Volume F67, Springer Verlag.

Mendes, E. Hall, W. & Harrison R.98 (1998) “Applying Metrics to the Evaluation of Educational Hypermedia Applications”; Journal of Universal Computer Science, 4,4, 382-403

Mendes, M. E. X. (1997) "SHAPE - Southampton Hypermedia Authoring Paradigm for Education", transfer Thesis from MPhil to Ph.D., Department of Electronics and Computer Science, University of Southampton.

Mendes, M. Emilia X. and Hall, Wendy (1997a) "An empirical study of hypermedia authoring for education"; Proceedings of the CAL97 Conference, Exeter.

Mendes, M. Emilia X. and Hall, Wendy (1997b) "The SHAPE of Hypermedia Authoring for Education"; Proceedings of ED-MEDIA & ED-TELECOM 97, Calgary.

Nelson, Ted (1965) A File Structure for the Complex, The Changing and The Indeterminate, ACM 20th National Conference.

Nelson, Ted (1987) All for One and One for All, Hypertext '87 Proceedings, November.

Newcomb, Steven and Newcomb, Victoria (1991) The HyTime Hypermedia/Time-based Document Structuring Language, Communications of the ACM, November.

Nielsen, Jakob (1990) Hypertext/Hypermedia. Academic Press.

Nielsen, Jakob (1990a) The Art of Navigating through Hypertext, Communications of the ACM, March.

Nielsen, Jakob (1990b) Navigating through Large Information Spaces, Hypertext and Hypermedia, Academic Press.

Nielsen, Jakob (1991) Panel Discussion - The Nielsen Ratings: Hypertext Reviews, Proceedings of Hypertext '91.

Pearl, Amy (1989) Sun's Link Service: A Protocol for Open Linking, Proceedings of Hypertext '89, ACM Press.

Rana, Ajaz R., Bieber, Michael (1997), "Towards a Collaborative Hypermedia Educational Framework"; Proceedings of Thirtieth Annual Hawaii International Conference on System Science, Maui 610-619.

Rao, Usha & Turoff, Murray (1990) Hypertext Functionality: A Theoretical Framework, International Journal of Human-Computer Interaction.

Raymond, Darrell, R., and Tompa, Frank Wm (1988) Hypertext and The Oxford English Dictionary, Communications of the ACM, July.

Rearick, Thomas C. (1991) Automating the Conversion of Text Into Hypertext, Hypertext/Hypermedia Handbook, Eds. Berk, E. and Devlin, J., Intertext Publications/McGraw Hill Publishing Co., Inc., New York.

Rivlin, Ehud, Botafogo, Rodrigo, and Schneiderman, Ben (1994) "Navigating in Hyperspace: designing a structure-based toolbox"; Communications of the ACM, 37, 2, 87-96.

SIGLINK (1992). SIGLINK Newsletter. Excerpts on the HyTime International Standard, Marc.

Spiro, Rand J., Feltovich, Paul J., Jacobson, Michael J., and Coulson, Richard L. (1995) "Cognitive Flexibility, Constructivism, and Hypertext: Random Access Instruction for Advanced Knowledge Acquisition in Ill-Structured Domains"; Constructivism, L. Steffe & J. Gale (Eds.), Hillsdale, N.J.:Erlbaum.

Stotts, P.David, and Furuta, Richard (1991) Dynamic Adaptation of Hypertext Structure, Proceedings of Hypertext '91, ACM Press.

Thuring, Manfred, Haake, Jorg M., and Hannemann, Jorg (1991) Hypertext '91 Proceedings.

Trigg, Randall (1988) Guided Tours and Tabletops : Tools for Communicating in a Hypertext Environment, ACM Transactions on Office Information Systems, October.

Trigg, Randall H., and Irish, Peggy M. (1987) Hypertext Habitats: Experiences of Writers in NoteCards, Proceedings of Hypertext '87, ACM Press.

Utting, Kenneth & Yankelovich, Nicole (1989) Context and Orientation in Hypermedia Networks, ACM Transactions on Information Systems, January.

Vocht, J. W. (1994) "Experiments for the Characterization of Hypertext-structures"; Master's Thesis, Eindhoven University Technology, Department of Mathematics and Computing Science.

Wright, Patricia (1991) Cognitive Overheads and Prosthesis: Some Issues in Evaluating Hypertexts, Proceedings of Hypertext '91.

Yamada, Shoji, Hong, Jung-Kook, and Sugita, Shigeharu (1995) "Development and Evaluation of Hypermedia for Museum Education: Validation of Metrics"; ACM Transactions on Computer-Human Interaction, 2, 4, 284-307.

Yankelovich, Nicole, Haan, Bernard J., Meyrowitz, Norman K., Drucker, Steven M., (1988) Intermedia: The Concept and the Construction of a Seamless Information Environment, IEEE Computer, January.