SLAC-PUB-7150
May 1996

WEB DOCUMENT ENGINEERING*

Bebo White
Stanford Linear Accelerator Center (SLAC),
Stanford University
Stanford, California

bebo@slac.stanford.edu

http://www.slac.stanford.edu/~bebo/bebo.html

Abstract

This tutorial provides an overview of several document engineering techniques which are applicable to the authoring of World Wide Web documents. It illustrates how pre-WWW hypertext research is applicable to the development of WWW information resources.

Definitions

For the purposes of this tutorial, a Web document (or hypertext document) is a collection of Web pages (or hypertext pages). Each Web page is typically a single computer file. This file may or may not be an ASCII file containing HTML markup.

A Web site can be thought of as a single Web document if it represents a single, logical information space. As a result the techniques described in this paper may be used in the design of a Web site.

Web Document Engineering (WDE) is defined as the application of software engineering techniques to the design of Web documents. WDE also incorporates techniques unique to the support and development of the hypertext or hypermedia medium which have resulted from the research and development in hypertext systems. In addition, WDE draws from experience found in the established disciplines of human-computer interaction and on-line documentation authoring.

Why Web Document Engineering?

At a time when potential Web authors and information providers are told "teach yourself Web publishing in a week," why is Web Document Engineering important? The simplicity of HTML often leads to spontaneous ("markup as you go") page and document design. From a software perspective, many of these pages are analogous to the "spaghetti code" produced by inexperienced programmers.

Software engineering techniques facilitate the design of large and complex software projects. Components of these projects are often independently designed and coded modules produced by different programming staffs. A precise specification of the functionality of these modules and their interfaces is what makes these projects successful. The most obvious goal of software engineering is that the software solution meet the stated requirements. Four properties that are sufficiently general to be accepted as goals for the entire discipline of software engineering are:

1. modifiability
2. efficiency

* Work supported by Department of Energy contract DE-AC03-76SF00515.
Paper based on a tutorial presented at the Fifth International World Wide Web Conference, Paris, France, May 6–10, 1996.

3. reliability
4. understandability

The goals of WDE are similar to those of software engineering - to produce a structured, maintainable, modifiable system (i.e., document) capable of performing the function for which it is intended. However, Web/hypertext projects differ from traditional software development projects in several critical ways.

- they may involve different skill sets in addition to software designers and programmers;
- the design of a Web/hypertext document usually involves capturing and organizing a complex information domain and making that domain accessible to users;
- inclusion of multimedia/hypermedia presents additional challenges.

Web/hypertext document design is therefore a challenging process that is currently more of an art than a science. The need for prototyping and intensive testing with users is even more pronounced in Web/hypertext development than it is in traditional software development because the level of user tolerance to errors in Web documents is very low. Software developers should always attempt to take advantage of their programming medium by using their knowledge of their application environment (e.g., language, file system, etc.). Web document authors should likewise fully utilize the medium that hypertext affords.

The Background of Web Document Engineering

Contrary to the conception of many, the World Wide Web is not the first hypertext system - although it is certainly the most successful. Systems with names such as Memex, Xanadu, Augment, and Zog do not have the household familiarity that WWW has. Research into hypertext/hypermedia systems has been going on for more than thirty years. Much of this research has led to tools and techniques which could (and should) be applicable to the Web. Many of what appear to be new problems are actually old problems which have re-surfaced. Web Document Engineering draws upon some of this earlier work and upon work being conducted in support of contemporary hypertext systems to WWW.

In this tutorial, I will briefly describe three hypertext design methodologies which can contribute to Web Document Engineering. Each of these methodologies emphasize the importance of Web/hypertext document usability. In each, document navigation plays a key role since one of the most important factors in the success of a document is the ease with which a user is able to navigate within it, following links between the pages which compose it. Each methodology also pays attention to a user's perception of how the information within the document is organized. It is my belief that consideration of Web Document Engineering complements the attention page authors should give to content. The net result will be to address the observations made by Jakob Neilsen in his usability studies at Sun Microsystems —'People have very little patience with poorly designed WWW sites. As one user put it: 'the more well-organized a page is, the more faith I will have in the information.'"

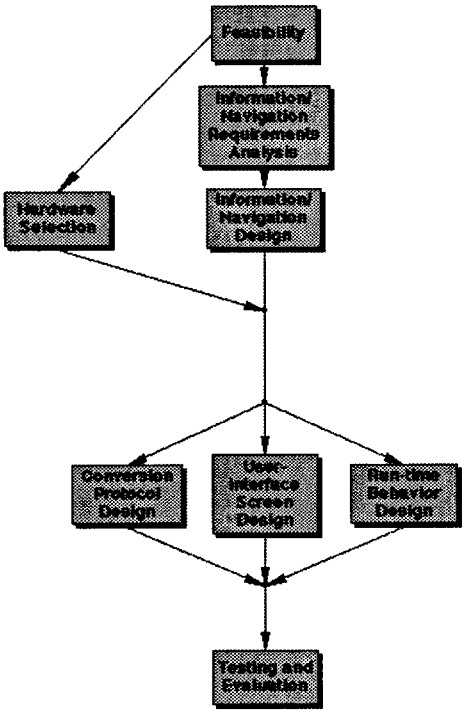
Design Methodologies

Each of the three hypertext design methodologies discussed has a different focus.

- one is structure-based with emphasis on the overall structure of the Web/hypertext document and the implied relationships between pages presented by that structure;
- one is relationship-based; This method analyzes the relationships between the components of the document (pages/hypertext nodes) and suggests the document design accordingly.

- one is information-based; This method concentrates more on content organization. The page relationships are a function of the anticipated use of the Web/hypertext document.

Each of the methodologies discussed assumes a Web document development cycle such as the following (adapted from Isakowitz, Stohr, and Balasubramanian).



The Feasibility step in this development cycle would likely result in the generation of a feasibility document (probably not a hypertext document). In this document, issues such as user needs and objectives would need to be addressed. While this is a critical element in the WDE process, it is beyond the scope of this tutorial. This step would be followed by an Information/Navigation Requirements Analysis leading to the development of a requirements document. It is from the requirements document that the document designer/author works

Each of the methodologies (or perhaps some combination) described in this tutorial are applicable in the Information/Navigation Design phase. It is in this phase that the actual structure of and relationships within the Web/hypertext document begin to take shape. It can generally be assumed that page/document content development occurs independently of this procedure. While one of the methodologies described is information-based, it does not address issues of content quality and quantity.

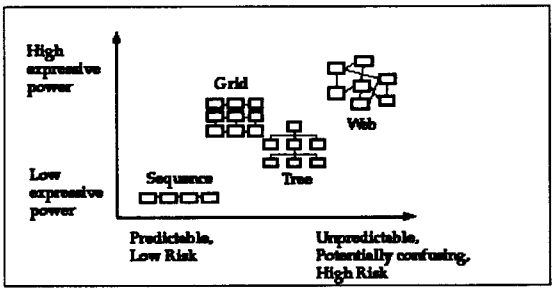
In the Conversion Protocol Design step, each element resulting from the Information/Navigation Design step is coded into the relevant format. For Web documents, this would usually indicate expressing the relationship via an HTML construct. User Interface Design involves the design of the "look and feel" of each element appearing in the Information/Navigation Design model. User interface design would likely include the descriptions of buttons, content layout,indices, and the page location of navigational tools (e.g., menubars, etc.) Decisions about how link traversal and navigational mechanisms are to be implemented are made in the Run-time Behavior Design.

It is also at this time when the page author considers the run-time impact of static versus dynamic pages and the use of server-side (CGI) scripts.

The Information Structure Approach

Even if the decision to author a document in hypertext is a good one, poor design of the document can easily present major problems. Just because document content has been defined into pages and links defined does not ensure that the document will be effective or attractive. "Successful hypertext, just as any successful writing project, depends on good design of the contents. The hypertext author who creates a new work or the hypertext editor who takes existing materials and puts them into hypertext form must take great care to produce excellence. The designer who assumes that it is safe to throw everything into the hypertext network and let the reader sort it out will be surprised by the negative reactions" (Shneiderman and Kearsley 1989).

The potential positive and negative impact of creating a hypertext document is realized when defining the relationships between the pages of that document. Figure 2. is an adaptation from Brockmann, Horton, and Brock (Brockmann and others 1989) evaluating the powers and risks associated with four common information organization structures.

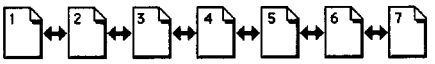


The rectangles in this figure represent elements/pages within the structure of a document.

In general, these structures represent an information space, a generalization of the hypertext document concept. An information space resembles an information database structured according to content and anticipated use.

The Sequence Structure

The simplest of these information structures is the sequence. This structure is the easiest of the structures to design and navigate through because it most closely resembles a conventional paper document. One of the positives ("pros") of this structure is that it is predictable. Users of a document structured in this manner are presented with a familiar and comfortable model. Links between pages are very well defined and navigation is linear (i.e., either forward or backward). Therefore, it is unlikely (if not impossible) for a reader to become "lost" while reading a document of this form.



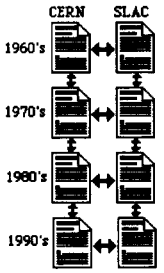
Documents structured in this manner are the least likely candidates for hypertext. There is typically not a "rich" relationship between the content of the pages. Except for the navigational links, each of the pages would actually be a "dead end" (i.e., containing no links) page.

Linear documents "converted" to hypertext (modified by the addition of page navigation links - previous page, next page, etc.) such as page-oriented, on-line documentation are examples of an implementation of the sequence structure.

The Grid Structure

The grid structure is the first of the information structures which can be perceived as multi-dimensional. It can be used to define significantly richer relationships between pages than is possible with the sequence structure. In general, the grid contains pages whose relationships are best described in a tabular fashion.

For example, the grid structure can provide a mechanism for representing the chronological relationship between the pages of a hypertext document. The following example document describes and compares the scientific research conducted at two high energy physics laboratories - CERN (The European Center for Nuclear Research) and SLAC (The Stanford Linear Accelerator Center).



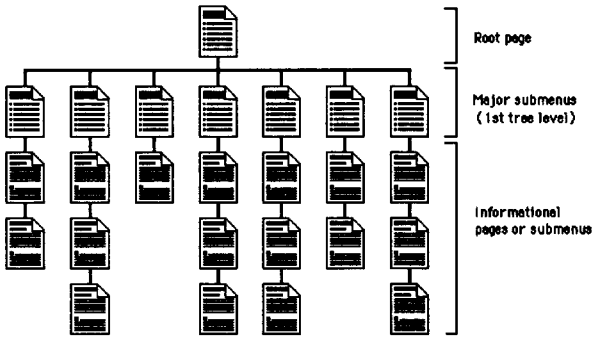
The document can be represented as a four-row, two-column grid composed of eight "pages." The relationships between the pages and how a reader navigates through the document is defined by its grid structure. Column navigation reflects the chronological progress of research at each laboratory. Row navigation permits a comparison of the research for the laboratories during the same time period. Any other navigational paths between pages represent a seemingly illogical sequence (e.g., a discussion of CERN research during the 1960's immediately followed by a discussion of SLAC research during the 1990's).

While the grid structure can be very expressive, it is also very fragile. Small changes in the organization of a document with this structure can result in the need for a major re-structuring of the page relationships. In this example, the addition of an additional laboratory to the model appears to severely jeopardize the direct relationships between pages of the different laboratories in the same time period. 'Such a restriction indicates that a document should be designed according to the grid structure only if that document is expected to remain stable and not undergo any major changes.

The Tree Structure

Use of a tree structure in a hypertext document allows pages to be presented in a hierarchical fashion. Like the grid structure, navigation between the pages is best defined multi-dimensionally.

The tree is perhaps the most common structure for documents which are written modularly. A frequently used example of a document with a tree structure is one which is navigated according to its table of contents. "Home pages" are often designed hierarchically with a tree structure. The following document is designed as a tree with a "home page" as the root node, the first level nodes as "submenus", and all remaining nodes consisting of individual pages or documents.



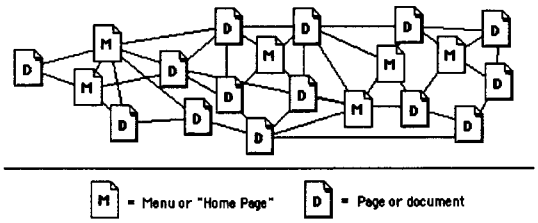
Like the sequence structure, the tree structure is likely to present an organizational model with which many users are familiar and comfortable. However, document authors and maintainers are also likely to encounter problems which are common to a hierarchical data structure.

Relationships between the elements of a tree structure are typically difficult to modify. For example, in the previous figure, the structure indicates that any links between informational pages and the root page must follow a path including a major submenu. Any direct link between these pages would violate the relationships defined by the structure and alter the common navigational tools of each page. Consequently, it may be a rare circumstance when a "pure" tree structure is actually used.

The tree structure is limiting in the manner by which pages can be added. In the example, adding a new page at the first level of the informational pages which is a new "child" of one of the existing major submenu pages, would force the modification of navigational links in multiple pages. Like hierarchical data structures, documents with a tree structure may periodically have to be "pruned" if the structure becomes "too shallow"(i.e., too many pages at the same hierarchical level) or "too deep" (i.e., too many hierarchical levels).

The Web Structure

The web structure is recognizably the most expressively powerful of the structures as well as the most potentially dangerous to a user. Pages and their links can be organized in any topological pattern which best defines their relationships and the navigational paths within the document. The document can be viewed as multdimensional (i.e., hyper-spatial). From a user's perspective, the relationships within the document may appear hopelessly confusing. For example,tools for navigation within such a document become critical in order to prevent a user from becoming hopelessly lost or confused.



For the design of complex web structures, the Relationship Management Methodology Approach may be applicable (Isakowitz, Stohr, and Balasubramanian 1995). In this method, entity-relationship tools are used to define the complex relationships encountered in a hypertext document structured as a web.

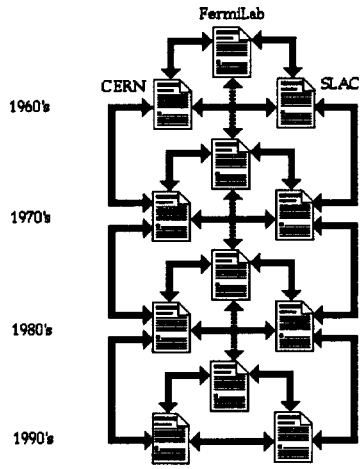
Composite Structures

Quite often the document structure is best expressed as a composite structure - some combination of the four fundamental structures. This structure can best be realized when a "page" of a document is actually another independent document or a break from the fundamental document structure.

Composite structures can also be used to represent more subtle relationships within Web/hypertext documents and to resolve what may initially appear to be design issues or flaws.

In the previous example of the use of the grid structure, the applicability of that structure to the document design would appear to collapse if another laboratory (e.g., FermiLab) were added. Adding another column to represent FermiLab would disrupt the navigational model between pages presented in the structure.

However, this problem is prevented if each row of the grid is considered as a web structure (actually three pages in a ring) rather than as sequence structures. Consequently the freedom of navigation between laboratory pages in each decade is preserved. This "composite" grid structure is described as follows:



The Relationship Management Methodology Approach

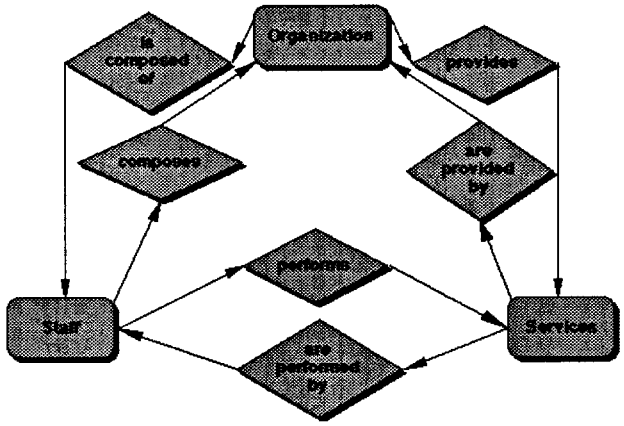
The Relationship Management Methodology (RMM) was first proposed by Tomas Isakowitz, Edward Stohr, and P. Balasubramanian. This is perhaps the most robust and complete of the methodologies described in this tutorial.

RMM is an entity-relationship based technique. Entity-relationship modelling is an analysis technique usually associated with database design and more recently object-oriented programming. When adopted for WDE the major differences are in the kinds of entities modelled. Entities are objects modeled in terms of the roles they play in a specific system. If the system is that of a Web/hypertext document, then those entities are pages (or

collections of pages) which are present in that document. Relationships are named associations between two or more entities.

Entity relationships can be represented graphically in entity-relationship diagrams (ERD). The most common graphical notation represents entities as rectangles and relationships as diamonds. Arrows and labels indicate the direction of the relationship. ERDs are often created from a narrative description of a system's subject matter by turning nouns into entities and verbs into relationships.

In the following example, entities and relationships describe a simple Web/hypertext document for a service organization.



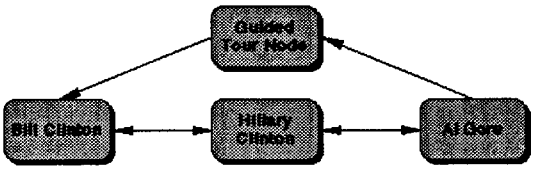
The example can be read as

- the Organization is composed of Staff; Staff composes the Organization;
- the Organization provides Services; Services are provided by the Organization;
- the Staff performs Services; Services are performed by the Staff.

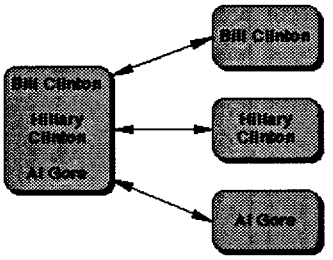
If each of the entities in this ERD (Organization, Staff, Services) are represented by hypertext pages or collections of pages, then the relationships described between the entities can be used to describe the navigational links between the entity/pages. These relationships may be simple hypertext links or more complex structures designed to completely describe use of the entity.These links must also be capable of expressing a one-to-many or a many-to-one relationship when the entities are compound. RMM describes a number of access primitives in order to describe such relationships.

RMM supports navigation across different entities via indices, guided tours, and groupings. An index acts as a table of contents to a list of entity instances, providing direct access to each listed item. A guided tour implements a linear path through a collection of items, allowing the user to move either forward or backward on the path. There are a number of useful variations on guided tours. For example, a circular guided tour links the last element back to the first; a guided tour with return to main has a unique mechanism that contains information about the guided tour itself and is both the starting and ending point of the tour; a guided tour with entrance and exit has different entrance and exit mechanisms.

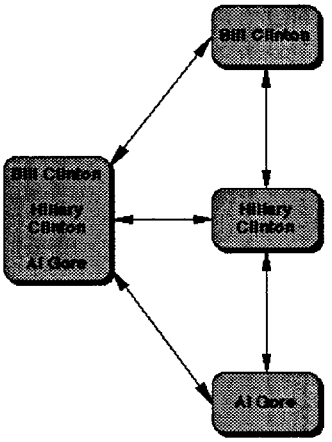
The following example illustrates how a simple guided tour might be used to express the relationships in the Staff entity of the previous example.



The same entity described as an index node is



The indexed guide tour provides the richest description of the Staff entity.



As the final step of the Relationship Management Method, these constructs are incorporated into the entity-relationship diagram to create what is referred to as the Relationship Management Data Model (RMDM). This model provides the implementation criteria used in the conversion protocol design step of the Web development cycle. In the reference, Isakowitz et. al. describe the development of a RMDM to HTML compiler.

Since entities themselves may have structure, RMM also includes the slice design phase. Slicing involves dividing an entity into meaningful units of information (slices) which are presented to users. Consequently, entity slices are analogous to Web/hypertext pages. If an entity contains more than one slice, the relationships between slices are also expressed using the RMM access primitives. The size of slices can also be determined by other methods such as those considered in the Information Mapping Approach.

The Information Mapping Approach

In general, the decision to write a Web/hypertext document should be based upon the structure of the content of the document. The use of hypertext is most appropriate when,

- the content of the document is a large body of information logically organized or structured into multiple units or fragments;
- these units or fragments are loosely associated with one another, though not necessarily in a sequential manner;
- a user or reader of the document only needs one unit or fragment of the content at any time.

These "Golden Rules of Hypertext" emphasize the units of information (e.g., the pages) which are delivered to the document user. The sizes of these units (or pages) reflects what the document author perceives a user needs or can best use at any one point in time.

The Information Mapping Method, described by Robert Horn, is a collection of tools and techniques designed for the analysis of complex collections of information.These methods attempt to break away from a paper page-based or paragraph-based paradigm. This method was initially developed for the design of on-line documentation (e.g., on-line manuals), but was found to address many of the critical design issues encountered in hypertext authoring.

Information Blocks

Horn describes an information block as "the basic subdivision of a subject matter, replacing the paragraph as the fundamental unit of analysis and presentation in functional and task-oriented text."The art of using the Information Mapping Method is learning how to define these information blocks.

In general, information blocks are composed of between one and seven sentences and/or graphical structures and tables identified clearly with a label. Sentences, graphical entities, and tables are defined as chunks. The magic number of chunks in an information block (approximately seven) is based upon human factors research into the limitations of the capacity of human short term memory.

There are four principles which can be applied in the construction of information blocks:

1. the chunking principle - group all information into small manageable units called information blocks and information maps;
2. the relevance principle - include in a chunk only information which relates to one major subject point; relevance is determined according to that information's purpose or function to the information user;
3. the consistency principle - for similar subject material, use similar words, labels, formats, organizations, and sequences;
4. the labelling principle - label every chunk and group of chunks according to specific criteria.

The similarity between the chunking and relevance principles and the "Golden Rules of Hypertext" strongly suggests the applicability of the Information Blocking Method to the design of Web/hypertext documents. Horn defines intuitive chunking as the dividing of information into chunks without knowing exactly why. Intuitive chunking is what drives authors to think about information in terms of paragraphs, pages, or screens. Precision modularity or modular authoring means chunking information following specific principles and guidelines.

Information Maps

The information block is the basic and smallest unit of information. An information map is defined as a collection of two or more, but usually no more than nine, information blocks about a specific topic. Like the information block, the number of units of information in an information map is based upon estimates of the size/capacity of human short term memory. In a Web/hypertext document, pages would ordinarily be implementations of information maps. This presumes a page as being the basic chunk of information provided to a user at one time. The page size consideration based on human short term memory attempts to insure that the information contained on that page can be readily assimilated and processed by the user within standard perceptual norms

Hypertrails.

Once the information blocks/pages have been identified, the Web/hypertext document can be constructed. The Information Mapping Method defines these documents as hypertrails. More specifically, a hypertrail is a set of links between chunks of information (information blocks and information maps) that organize and sequence information about a particular function or characteristic of subject matters. By identifying a document with a type of hypertrail, the functionality of that document and its intended user audience can be identified. Likewise, authors could establish templates for documents corresponding to specific hypertrail types.

A prerequisite hypertrail defines a set of links between information maps, information blocks, etc. in those circumstances where the connections specify which maps/pages users must understand or which maps/pages the document author wants to insure are read. Examples of the use of a prerequisite hypertrail are on-line tutorials, legacy documents organized as chapters, sections, subsections, etc., and commercial applications where the document author would like to insure that a user visits the requisite pages.

A classification hypertrail defines a set of links in a hypertext document which enables a user to:

- find links higher or lower on a classification tree for a particular subject;
- display a classification structure of a given region or area within the hypertext document.

The classification hypertrail is a specific application of the tree or hierarchical structure discussed earlier. As with that structure, it is important that the document/hypertrail user be aware of the relationships and groupings since they often represent implicit information about the document content.

Chronological hypertrails are links between information maps/pages that organize information with respect to time. Examples of chronological hypertrails are those which present:

- a sequence of events;
- a storyline;
- the natural development of a system.

Geographical hypertrails link together descriptions and maps of geographical information. One of the major ways of organizing information is spatially. Geographical hypertrails permit a user to move through space (e. g., zoom in and out, transfer from one physical location to another). This type of hypertrail is especially applicable for graphical information (i.e., information maps with graphical blocks). Examples of WWW use of geographical hypertrails are interactive graphics (in-line images with the ismap attribute).

Project hypertrails are specific kinds of chronological hypertrails that link planned and past events all of which are focussed on a personal or group project. Projects can be described as work organized around a specific goal

that will take longer than a simple task. Workflows and project timelines are useful methods for expressing relationships between items of information.

Structure hypertrails link specific substructures described in information blocks to the larger structure. A user can begin searching a structure hypertrail from any part of the structure or substructure in the hypertrail. There is a logical similarity between geographical hypertrails and structure hypertrails. Geographical hypertrails link spatial relationships between different structures while structure hypertrails link subparts to a larger structure.

Decision hypertrails link all of the information surrounding a decision-making process.

Definition hypertrails provide links between different meanings of a single term in a hypertext document or between related terms in one or more documents.

Example hypertrails link examples provided within Web/hypertext documents. In a typical implementation, users are provided access to an example if and only if they request it by following a link.

Conclusion

Like any large software project, the development of large and complex World Wide Web documents requires application of an engineering discipline. Research and development efforts in hypertext and hypermedia systems have identified important techniques and methodologies which are applicable to the design and authoring of Web documents. The use of such disciplines in the design of these documents addresses critical design issues such as usability,maintainability, and modifiability. The continuing rapid growth of WWW and related technologies will necessitate ongoing development in Web Document Engineering (WDE).

References

Barrett, Edward (editor), (1989), *The Society of Text - Hypertext, Hypermedia, and the Social Construction of Information*, Cambridge MA:The MIT Press.

Brockmann, R. John (1990), *Writing Better Computer User Documentation - From Paper to Hypertext*, New York:John Wiley.

Engelbart, Douglas C. (1963), *A Conceptual Framework for the Augmentation of Man's Intellect, Vistas In Information Handling. Vol. 1.*, Washington, DC: Spartan Books.

Horn, Robert E. (1989),*Mapping Hypertext - Analysis, Linkage, and Display of Knowledge for the Next Generation of On-Line Text and Graphics*, Lexington MA:The Lexington Institute.

Hoyt, Jennifer (editor) (1996), *Related Readings in Hypermedia*, Institute for Advanced Technology in the Humanities at the University of Virginia, <URL:http://jefferson.village.virginia.edu/readings/hypermedia.html>

Isakowitz, T., Stohr, E., and Balasubramanian, P., *RMM: A Methodology for Structured Hypermedia Design*, Communications of the ACM, August 1995 - volume 38, number 8.

Landow, George P. (1994), *Hypertext: The Convergence of Critical Theory and Technology*, Baltimore, MD: The Johns Hopkins University Press.

Lynch, Patrick.(1996), *Web Style Manual*, Yale Center For Advanced Instructional Media, <URL:http://info.med.yale.edu/caim/StyleManual_Top.HTML>.

Miller, George A.,*The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information*, Psychological Review 63 - number 2, 1956.

Neilsen, Jakob. (1995), *Interface Design for Sun's WWW Site*, <URL: http://www.sun.com/sun-on-net/www.sun.com/uidesign/>.

Shneiderman, Ben, and Kearsley, Greg.(1989), *Hypertext Hands-On: An Introduction to a New Way of Organizing and Accessing Information*, Reading, MA: Addison-Wesley.

Thuring, M., Hanneman, J., and Haake, J., *Hypermedia and Cognition: Designing for Comprehension,* Communications of the ACM, August 1995 - volume 38, number 8.

White, Bebo, (1996), *HTML and the Art of Authoring for the World Wide Web,* Norwell MA: Kluwer Academic Publishers.