CHAPTER 2

IMPLEMENTATION ISSUES

1. Conversion of Text to Hypertext

In recent years, there has been a great amount of enthusiasm in converting every printed document into hypertext form. Though the rationale behind this is not always correct, manual hypertexts such as encyclopedias, dictionaries, training manuals are very well suited for hypertext conversion. These reference materials are not used the same way as other books. They are highly cross-referenced and are used in a non-linear fashion. Readers look for various structural cues such as table of contents, indexing by subject, keywords, authors, page numbers, sections, see-also listings etc.

1.1 Limitations of Printed Text

There are a lot of limitations imposed by the printed versions of reference books. These include [Cook, 1988]:

1. Even though some of the reference books run into many volumes, the amount of information that can be stored is still limited compared to electronic forms of storage. It is also difficult to search through large volumes of printed material.

2. They cannot be updated periodically.

3. Information search is predominantly lexical - the table of contents and the index provide the facility to jump to topics but the amount of cross-referencing is minimal. The printed index is limited by the size and selection criteria of the authors and does not always direct the user to all relevant information.

4. Information cannot be dynamically re-arranged to suit the individual needs of various kinds of users.

5. Information is spread over a number of volumes and after some time information retrieval becomes tedious.

1.2 Advantages of Hypertext Format

Enhancing encyclopedic information into hypertext format has many advantages [Raymond & Tompa, 1988], [Cook, 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.

1.3 Conversion Issues

As much as they are well suited for adaptation to hypertext, converting text to hypertext has been a classic problem while dealing with very large information spaces such as training manuals, encyclopedias, dictionaries. Currently published literature on hypertext contains little work directly related to the scale of transforming large volumes of encyclopedic text into hypertext form (most deal with creating small hypertext documents, not converting large documents to hypertext).

The following are some of the issues involved in converting text to hypertext [Glushko, 1989], [Riner, 1991]:

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.

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.

2. Types of Conversion

There are two ways to convert existing documents into hypertext form - manual conversion and automated conversion [Riner, 1991].

2.1 Manual Conversion

Manual conversion involves using a hypertext authoring tool to create nodes and links manually. Hence, this process depends on the way the author (or the person who builds the hypertext form from a linear form) understands the structure and flow of the presented material. Being a repetitive process, it is prone to human error. There cannot be anything worse than a badly converted hypertext version of a linear document. Also, manual conversion is suitable only for small documents.

2.2 Automated Conversion

Automated conversion facilitates the easy identification of nodes and links based on pre-defined criteria. The output of an automated conversion process can be easily modified/enhanced by authors. Also, large information spaces such as dictionaries, encyclopedias, and training manuals can be converted to hypertext format very efficiently. As mentioned earlier, most linear documents have structural elements such as titles, sub-titles, chapters, sections, paragraphs, sentences, words, figures, tables, and indexes. An automated conversion system must be able to recognize these structural elements, identify nodes and links, and construct the appropriate links to form the hypertext network [Riner, 1991]. Links can capture both the hierarchical and referential nature of the material.

2.3 Guidelines for Conversion

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.

2.4 Automatic Link Construction

In most current systems, a large authoring effort is required to insert links into documents. Very little work has been done in the area of automatic link construction - links based on the semantic analysis of the underlying text. Such a feature requires considerable amount of analysis and the incorporation of an Artificial Intelligence engine.

In an effort towards automatic linking of hypertext nodes, Bernstein proposed a "link apprentice", a program that can examine a draft hypertext and create appropriate links. This can be done by establishing links based on the semantic analysis of the underlying text. Since these "clever" apprentices are intrinsically difficult to construct (they not only need precision but also accuracy and recall), he suggested a "shallow apprentice" - a system which discovers links through superficial textual analysis (of statistical and lexical properties) without analyzing meaning [Bernstein, 1990].

The shallow apprentice uses the Bloom filter method of text searching. Each hypertext page (node) has a Bloom filter hash table where each word is hashed. These hash tables are used to define a similarity between two hypertext pages by taking the normalized dot product of their hash tables. The apprentice will search the entire document checking for similarity to the page upon which the hypertext author is currently working. It will then retrieve the twenty pages that appear to be most similar to the current page.

The hypertext author can also construct hypertext paths or tours by choosing an interesting starting point and requesting the apprentice to construct a path through related material. However, the path may not be in logical order since the apprentice does not check for semantics.

3. Hypertext Templates

Hypermedia templates are defined as sets of pre-linked documents that can be duplicated [Catlin et al., 1991]. Another definition of a hypertext template states that it "is a 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 will definitely 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. Many applications such as collaborative writing, teaching aids etc., have some common basis that can be transformed into a hypertext template.

The following are some requirements for a hypertext system to provide templates:

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.

Intermedia, developed at Brown University, provides the following features for hypermedia templates [Catlin et al., 1991]:

1. Intermedia system provides the facility to create templates including the documents and links that make up the template. That is, a hypermedia author has the ability to create nodes, links, and link sets or webs within a template. A list of webs can be associated with a template one of which can be chosen as the default when the template is duplicated.

2. Documents within the same template can be linked. Users can also link a document in a template to another document outside of the template.

3. The user can specify the folder or directory under which each document is created and also the folder where the template has to be duplicated. The user can also name and save a template for future use. The system will make copies of all folders and documents and automatically link them just as the original template was linked. All new documents will be displayed for the purpose of editing.

4. When a template is duplicated, all associated documents and links can be easily accessed in new folders. The user will be prompted to choose one of the webs associated with the template. The user can open a template, add document members, delete members, rename them, create or modify links etc. Contents of documents can be edited.

5. The user can easily find out which template was used to make a new hypermedia collection.

6. The original template itself is write-protected so that users do not edit it accidentally.

Researchers at Brown University believe that the ability to duplicate collections of linked material can be extended to other hypertext environments. Research is required in the area of propagating editing changes to documents that were created using a particular template. The concept of class-based templates needs exploration - templates should be able to inherit characteristics from other templates (similar to the concept of inheritance in object-oriented systems). With inheritance, when an author changes a parent template, all of its sub-classed templates would change accordingly.

4. General Guidelines for Authoring Hypertext Documents

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 no established guidelines for writing hypertext documents. Guidelines have been developed, by Thuring et al., for the construction of a coherent hypertext document. Such a document should consist of the following three components - the content part, the organizational part, and the presentation part [Thuring et al,1991].

4.1 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:

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. According to Thuring et al., 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." [Thuring et al., 1991].

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.

4.2 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:

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:

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:

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.

Based on the above, the following hierarchy of design objects is derived.

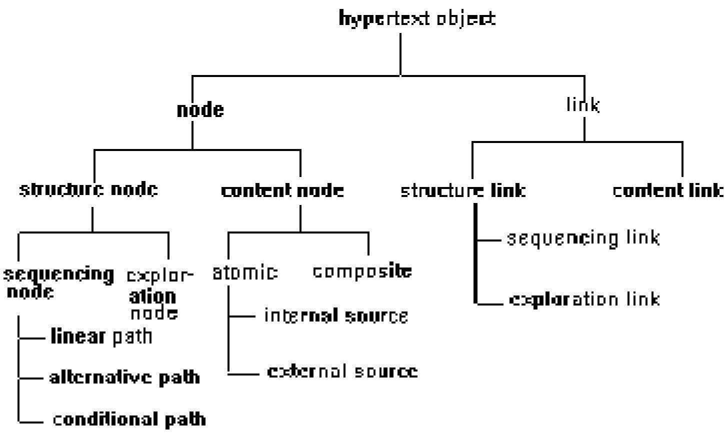

Figure 2.1 Hierarchy of Design Object Classes [Thuring et al., 1991].

4.3 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:

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.

5. Dynamic Hypertext

Halasz had identified dynamic or virtual structures, computation, and extensibility/tailorability as some of the issues to be addressed by next generation hypertext systems [Halasz, 1988]. Most current generation hypertext systems implement a static and explicit model of hypertext - nodes, links, and link markers must be declared explicitly and be fully enumerated during creation time as opposed to being declared dynamically and generated upon demand [Bieber, 1993].

Information systems such as Decision Support Systems (DSS) and Expert Systems require a dynamic implementation of hypertext, one that relies primarily on virtual structures and computation in order to generate a hypertext network in real time [Bieber, 1991]. Bieber developed a DSS shell that supported multiple DSS applications through a hypertext user interface. The user interface containing the hypertext engine provided DSS applications with hypertext functionality such as navigation, virtual structures, computation, and tailored presentation. Since many of the components making up the DSS are generated in real time as a result of user interaction, it was difficult to pre-define all nodes, links, and link markers at creation time.

Bridge laws were developed to determine the appropriate links and link markers automatically and embed them in the interactive application [Bieber, 1991]. These link markers provided access to reports, operations (DSS commands), and other components of DSS applications. Bridge laws are translation routines provided by the application to the hypertext interface. They map the elements defined in the application's original non-hypertext data or knowledge base to entities in the hypertext engine. They do not alter the application's data or knowledge bases.

A user interface control subsystem was developed to maintain global information about user profile, user-defined links, comments, application keywords and application supplied bridge laws. It was responsible for interpreting contents coming from the application (using bridge laws) to provide virtual link markers by highlighting the objects. It tailored different views of the application based on different sets of filters.

Stotts and Furuta describe virtual structures as dynamic adaptation of hypertext structure [Stotts & Furuta, 1991]. 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, 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.

6. Linearization of Hypertext

The reverse problem of converting text to hypertext is to linearize a hypertext document for printing. The need for a linear, printed document will exist for some time to come. Yankelovich et al. suggest that "printing a branching [hypertext] document in a linear fashion poses both technical and conceptual problems." [Yankelovich et al., 1985]. It is easy to linearize a hypertext document having a strict hierarchical structure by performing a depth-first tree traversal, by printing the first chapter and its sections and moving onto the next chapter and so on. However, in the general case where the hypertext document is a highly connected network without any special order, it is very difficult to produce a good linear document [Nielsen, 1990].

Activities involved in the production of a linear document are together referred as "document preparation" [Trigg & Irish, 1987]. According to Trigg and Irish, document preparation does not include writing activities such as notetaking and reorganizing. It can happen throughout the writing process, often commencing well before the final text is composed. Paper structuring or layout is done in an outline, which is massaged and fine-tuned for some time before any text is written. In NoteCards, the outline takes the form of a filebox hierarchy whose filebox titles correspond to section titles of the paper. Some writers just use a single text card to capture the overall structure of the paper to be written. This may involve pulling up other relevant cards and copying text from these note or paraphrasing. This kind of a composition of text (essentially in a linear fashion) is the same as moving away from the concept of hypertext. Smoothing the document and integration is done with the linear document rather than with the source cards.

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]. The concept of paths has also been explored using the Scripted Document System at Xerox [Zellweger, 1989]. 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.

7. Summary

Converting linear text to hypertext has been a classic problem while dealing with large information spaces such as encyclopedias, training manuals and dictionaries. Attempts have been made to convert these printed material both by manual and automatic means. Some researchers have suggested guidelines for conversion. In addition to automatic conversion of text to hypertext based on structural features, researchers have attempted to construct automatic links based on lexical and semantic analysis of text. Hypertext templates facilitate the design, organization, and presentation of a collection of knowledge in the form of hypertext. Researchers have suggested some general guidelines for authoring hypertext documents. These include splitting a hypertext document into three components: the content part, the organizational part, and the presentation part. There have been efforts to make hypertext systems more dynamic by incorporating virtual structures, computation, and filters. Though miniscule in nature, attempts have been made to linearize hypertext documents for the purpose of printing.

References

[Bieber, 1991]. Bieber, Michael. Issues in Modeling a "Dynamic" Hypertext Interface for Non-Hypertext Systems, Proceedings of Hypertext'91, ACM Press,1991.

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

[Bernstein, 1990]. Bernstein, Mark. An Apprentice That Discovers Hypertext Links, Proceedings of

ECHT ’90, 1990.

[Catlin et al., 1991]. Catlin, Karen S., Garrett, N.L., and Launhardt, Julie A. Hypermedia Templates: An Author’s Tool, Proceedings of Hypertext ’91, ACM

Press, 1991.

[Cook, 1988]. Cook, Peter. An Encyclopedia Publisher’s Perspective, Interactive Multimedia, Apple Computer Inc., Microsoft Press, 1988.

[Glushko, 1989]. Glushko, Robert J. Transforming Text Into Hypertext For a Compact Disc Encyclopedia, Proceedings of CHI ’89, ACM Press, 1989.

[Halasz, 1988]. Halasz, Frank. Reflections on NoteCards : Seven Issues for the Next Generation of Hypermedia Systems, Communications of the ACM, July 1988.

[Nielsen, 1990]. Nielsen, Jakob. Converting Existing Text to Hypertext, Chapter 11, Hypertext and Hypermedia, Academic Press, 1990.

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

[Raymond & Tompa, 1988]. Raymond, Darrell R., and Tompa, Frank WM. Hypertext and the Oxford English Dictionary, Communications of the ACM, July 1988.

[Rearick, 1991]. Rearick, Thomas C. 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, 1991.

[Riner, 1991]. Riner, Rob. Automated Conversion, Hypertext/Hypermedia Handbook, Eds. Berk, E. and Devlin, J., Intertext Publications/McGraw Hill Publishing Co., Inc., New York, 1991.

[Stotts & Furuta, 1991]. Stotts, P.David, and Furuta, Richard. Dynamic Adaptation of Hypertext Structure, Proceedings of Hypertext ’91, ACM Press, 1991.

[Thuring et al., 1991]. Thuring, Manfred, Haake, Jorg M., and Hannemann, Jorg. Hypertext ’91 Proceedings, 1991.

[Trigg & Irish, 1987]. Trigg, Randall H., and Irish, Peggy M. Hypertext Habitats: Experiences of Writers in NoteCards, Proceedings of Hypertext ’87, ACM Press, 1987.

[Yankelovich et al., 1985]. Yankelovich, Nicole, Meyrowitz, Norman, and van Dam, Andries. Reading and Writing an Electronic Book, IEEE Computer, 1985.

[Zellweger, 1989]. Zellweger, Polle T. Scripted Documents : A Hypermedia Path Mechanism, Proceedings of Hypertext ’89 Conference, November 1989.