

Text-Hypertext Mutual Conversion and Hypertext Interchange through SGML

Min Zheng

*Institute of Computer Science and Technology,
Beijing University, Beijing, 100871, China*

Roy Rada

*Department of Computer Science,
University of Liverpool, Liverpool L69 3BX, England*

Abstract

This paper presents an SGML-MUCH (S-M) system, the I/O subsystem of a collaborative authoring and reuse system called MUCH (Many Using and Creating Hypermedia), for text and hypertext mutual conversion and hypertext interchange. The S-M system can dynamically generate text documents from the MUCH document database by traversing a subgraph of the hypertext database. With various options on traversal, different versions of documents can be generated from the same set of nodes and links in the document database. The documents generated from the MUCH document database include not only document content but also groupware and hypertext information that can be used for hypertext interchange. With this generated document, hypertext for different systems such as Guide and Hyperties can be automatically generated and new features, like alternate outline and automatically derived indices can also be incorporated. The S-M system, which was developed using several public-domain development tools, adheres to standards where possible, and generally focuses on 'openness'.

1. Introduction

Hypertext research involves many contemporary topics including artificial intelligence, human-computer interaction, information system and social psychology. Hypertext is a new means for knowledge representation. As in a semantic net, information in hypertext is represented as a set of nodes connected to each other by a set of links which represent relationships between the nodes.

While hypertext is superior to text for information representation at many aspects, for various reasons, the paper form of documents still dominates the dissemination of information and massive information is available in

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

CIKM '93 - 11/93/D.C., USA

© 1993 ACM 0-89791-626-3/93/0011\$1.50

paper form. If one could create a universal transforming environment under which any text or hypertext information can be automatically transferred from one system or form to another regardless of its original format, the process of knowledge acquisition and dissemination would be dramatically enhanced.

Hypertext interchange and text-hypertext conversion have been studied by many researchers. Some algorithms and systems have been presented and developed for automatically generating hypertext from linear text [Chignell91, Furuta90a, Glushko89, Nunn88, Ritchie89, Sarre90]. Standard Generalized Markup Language (SGML), has been used in several systems and environments for hypertext generation, such as in Guide [Ritchie92], DynaText [EBT90] and World Wide Web (WWW) [Berners-Lee92]. Many hypertext systems such as Intermedia [Yankelovic88] and Hyperties [Congnetics92] support export features, namely converting hypertext into linear documents.

One of the key issues associated with hypertext interchange is the standardization of hypertext models. Several hypertext reference models have been presented [Furuta90b, Halasz90, Lange90, Riley90], which can be elaborated as data structures and used as an intermediate format for hypertext interchange. Experiments using the Dexter model as an intermediate format for hypertext interchange has also been performed [Leggett91].

Usually, bi-directional conversion between text and hypertext and hypertext interchange are considered as two different applications. However, as hypertext interchange generally involves a plain file representation of a hypertext document and the bi-directional conversion between text and hypertext also uses a plain file representation, one might expect a relationship between the applications for interchange and conversion. Can one find a solution which uses one system to support both of these applications?

In our research, a software system — the SGML-MUCH system — has been developed, which enables users to import an SGML document into the MUCH (Many Using and Creating Hypermedia) collaborative hypertext system [Rada91a] and export (a part of) the MUCH hypertext database into an SGML document. An SGML docu-

ment exported from the MUCH system keeps all the information of the hypertext, including links, nodes and all their attributes. At the same time, the exported document is also a structured linear document which can be processed by a normal document preparation system and printed on paper. Once this document is imported into a MUCH database, either the original or another MUCH database, the hypertext will be reconstructed or updated with the SGML document without losing any useful information. With this function, collaborative authoring, especially remote collaboration can be facilitated.

2. Related standards and models

The Standard Generalized Markup Language (SGML) [ISO/IEC88] is a language for logical document structure and is an international standard for publishing. SGML is based on the Generalized Markup Language developed at IBM [Goldfarb90]. It uses the generic markup of the structural elements of a document without regard to their presentation, which is seen as a separate issue. It is based on the principles of the generic encoding of documents and 'marks up' a document's logical structure and not its physical presentation. SGML contrasts to typographic markup, since font and style are not considered during document markup.

HyTime (Hypermedia/Time-based Document Structuring Language) [ISO/IEC92] is a standard neutral markup language for representing hypertext, multimedia, hypermedia, and time- and space-based documents in terms of their logical structure. Its purpose is to make hyperdocuments interoperable and maintainable over the long term. HyTime can be used to represent documents containing any combination of digital notations [Newcomb91].

HyTime may be viewed as an application of SGML and is parsable as SGML. It uses SGML syntax with its own semantics. It has a set of "HyTime architectural forms" as object classes, each of which has a fixed value of a HyTime attribute. Limitless numbers of subclasses can be created by including the HyTime attribute and conforming to the model for a particular HyTime architectural form. HyTime provides a standardized means of expressing information which is difficult to represent in SGML, such as:

- (1) Intra- and extra-document locations, and arbitrary links between them;
- (2) The scheduling of multimedia objects in 'finite coordinate spaces';
- (3) Rendering instructions for arbitrarily projecting such objects onto other finite coordinate spaces and other constructs.

The Dexter model is an attempt to capture, both formally and informally, the important abstractions found in a wide range of existing and future hypertext systems. The goal of the model is to provide a principled basis for comparing systems as well as for developing interchange and interoperability standards [Halasz90].

The Dexter model divides a hypertext system into three layers, the *runtime layer*, the *storage layer* and the *within-component layer* (see Figure 1). The main focus of the model is on the storage layer, which models the basic node/link network structure that is the essence of hypertext.

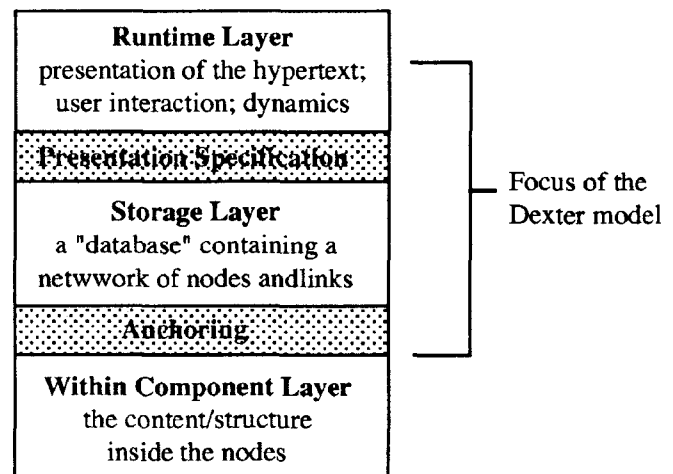


Figure 1. Layers of the Dexter model.

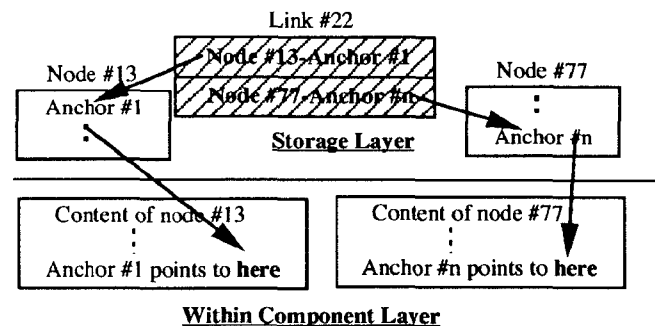


Figure 2. Simple depiction of link mechanism of the Dexter model.

The storage layer only provides the mechanisms to organize components (nodes) and links to form hypertext networks without considering the contents of components (figure 2), which is the task of the within-component layer. The interface between the storage layer and within-component layer is an anchoring mechanism which is used for addressing (referring to) locations or items within the content of an individual component. The runtime layer focuses on how a component/network is presented to users. Between the runtime layer and storage layer, there is an in-

terface called *presentation specifications*, which provides a mechanism to encode the information about presentation into the hypertext network at the storage layer.

3. SGML-MUCH system

The SGML-MUCH (S-M) system was designed as an I/O sub-system of the MUCH collaborative authoring system but it can also be used for general format and text-hypertext conversion. The S-M system can dynamically generate text documents from the MUCH document database by traversing a subgraph of the hypertext database. With various options on traversal, different versions of documents can be generated from the same set of nodes and links in the document database. The generated documents include not only document content but also management and hypertext information which can be used for hypertext interchange. With this generated document, hypertext for different systems such as Guide and Hyperties can be automatically generated and new features, like alternate outline and automatically derived indices can also be incorporated.

3.1. MUCH document database

Conceptually, the document database is a hypermedia database and the information is organized as a database of richly-linked nodes. A node is a unit of information retrievable from the database, as well as a semantic unit of the subject matter.

The document database is organized following a simplified Dexter model with the link mechanism shown in Figure 2. Following the principles of the Dexter model which separate hypertext structure from its content, the MUCH database consists of two parts: the hypertext database (structure) and the document components (contents) which correspond to the storage layer and the within-component layer of the Dexter model.

The MUCH hypertext database includes two kinds of component: link and node-info. The link component connects exactly two nodes and also contains attributes of the link such as link type, author and date of creation. Node-info is the information about individual nodes and includes anchors to the associated document component. The node attributes include name, type, class, author, date, updaters, index terms and others.

A document component is the content of a node. The Dexter model leaves the structure (or format) of the components in the within-component layer open to system developers. The MUCH system was implemented in the Andrew multimedia environment, and the Andrew ToolKit (ATK) format [Borenstein90] was used for the document components. ATK provides a unified data structure called data stream which consists of a set of insets for represent-

ing multimedia information, such as text, graphics, image, animation and sound. In the MUCH database, each blob is an ATK data stream which includes inset data, i.e. multimedia information.

The MUCH hypertext model focuses on link components. Each link has a link type, such as "document", "annotation", "reference" or "thesaurus". A link type can be either hierarchical or non-hierarchical. For instance, the document type is a hierarchical type whilst the reference type is not. The link type determines the character of linked nodes and may affect their future processing. For example, traversing following different types of links may generate different kinds of document.

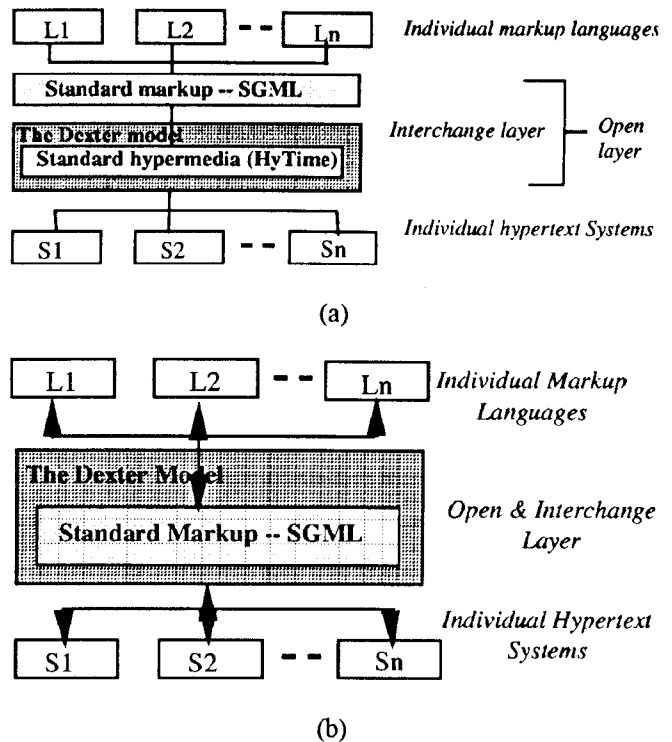


Figure 3. The SHyD model: (a) the original model; (b) the simplified model.

3.2. S-M System Structure

The S-M system is based on a simplified SHyD (SGML-HyTime-Dexter) model [Zheng93]. The SHyD model adopts some international standards (SGML, HyTime) and a widely accepted hypertext reference model (the Dexter model) to facilitate information interchange between text and hypertext. Figure 3 shows the logical structure of the SHyD model and its simplified version. In the simplified model, the HyTime is not used due to some practical reasons. Because of the restriction of SGML on representing multimedia information, this simplified model is suitable for text and hypertext applications rather than hypermedia.

The development of the S-M system generally focuses on 'openness' and uses standards where possible. The structure of the S-M system is shown in Figure 4. In this system, SGML representation plays a key role in information interchange and conversion. The SGML DTD defined following the Dexter model is used as an intermediate format to connect different text and/or hypertext systems.

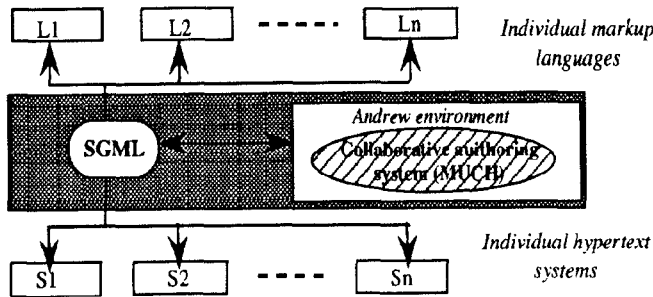


Figure 4. The structure of the SM system.

With the support of the S-M system, the MUCH system can accept information from existing text documents in various formats and reorganize the information in order to generate linear documents for printing, and hypertext for browsing. In addition to importing text into the MUCH system and exporting MUCH hypertext to linear documents, the S-M system also supports automatic hypertext generation for different browsing systems such as Guide and Hyperties with several features, such as alternate outline and automatically derived indices.

The S-M system actually can be used as a general text and hypertext conversion tool. It can convert a text or hypertext document to an SGML document and then convert to either a text or hypertext document in another format. With the MUCH import function, the data transformation processing may take advantage of the capabilities of the MUCH system, such as data reorganization and alternate outline generation. When a document is converted into SGML, it can be imported into the MUCH and processed there, and then exported to an SGML document and translated into other text or hypertext systems.

To establish the connection between SGML and individual text and hypertext systems, the following issues need to be considered:

- * How to define an SGML DTD which can represent hypertext structure and at the same time, keep the structure of linear documents.
- * How to export a hypertext without losing any information and make it processable by a normal text processor.
- * How to import an SGML document into the MUCH system. For those documents exported from MUCH, how to guarantee that the hypertext is not corrupted.

- * When partial hypertext import/export is considered, how to export a part of the hypertext and how to update the information when an exported part with modification is imported back.

4. Defining SGML-MUCH DTD

In text documents, especially those with large volumes such as textbooks, some link information is implied. Usually, there are two kinds of links: one is a hierarchical link which is implied in the document structure like chapter, section, subsection...; and the other is a non-hierarchical link, like bibliographic citations, indices and footnotes. This kind of link is often given explicitly. Similarly, in a hypertext, the nodes are usually explicitly linked by two kinds of links — referential links and organizational links. Referential links are non-hierarchical links which may connect any two points in the hypertext, whereas the organizational links, which connect a parent node with its children, implement hierarchical information [Conklin87].

The similarities discussed above are the foundation of bridging text and hypertext. However, hypertext usually has much richer links than text, and the nodes in a hypertext may have more attributes than a text section has. This is especially true in a collaboratively created hypertext. To export a hypertext to a linear document with all the hypertext information, we need a markup language which has extra structures not included in conventional markup languages.

To define a DTD to represent hypertext, one needs an underlying reference model, such as the Dexter model. On the basis of this model, the SGML-MUCH (S-M) DTD can be defined in such a way that a document is composed of many blocks and links. A block is a subject matter corresponding to a section of paragraphs in a text document. A link is an element with an attribute list which includes link type, source and destination points, and other information. A link element is embedded in the source node and points to the destination node with anchor IDs for both source and destination. The location of the two end points can be resolved by the anchors in the node information.

One requirement for the S-M DTD is that the document marked-up by this DTD should be able to be processed as a normal text document. In order not to interfere with the normal structure of the text documents, a new 'info' element is defined to represent the node attributes instead of using the ATTLIST of the node. Besides, both 'info' and 'link' elements are defined as optional sub-elements in the heading element in every block. By simply ignoring these two elements, a document can be processed as a normal SGML document. Figure 5 is a part of the DTD showing the structure of the SGML documents.

```

<!element document - -
  (titlepag, header?, abstract?,
   toc?, lof?, lot?, block, chapt*,
   (appendix, chapt+)?, biblio?)
  +(footnote)>
  .....
<!entity % inline " (#pcdata | f |
  x | %emph; | sq | %xref)* " >
<!entity % sect "heading, header?,
  block ">
<!element heading o o (%inline)>
<!element chapt - o (%sect, sect*)
  +(footnote)>
<!element sect - o (%sect, sect1*)
  +(footnote)>
  .....
<!element sect10 - o (%sect)>
<!element block o o (info?, link*,
  block*)>
<!element info - o anchor*>
<!attlist info id CDATA #IMPLIED
  type CDATA "text"
  class CDATA "document"
  author NAME #IMPLIED
  date CDATA #IMPLIED
  update NAME #IMPLIED
  terms CDATA #IMPLIED
  others CDATA #IMPLIED >
<!element anchor - o empty>
<!attlist anchor id NUMBER #IMPLIED
  loc CDATA #IMPLIED>
<!element link - o empty>
<!attlist link type CDATA "document"
  sanchor NUMBER #REQUIRED
  -- Source anchor number --
  dnode CDATA #REQUIRED
  -- Detination node --
  danchor NUMBER #REQUIRED
  -- Destination anchor number --
  .....>

```

Figure 5. A part of the S-M DTD which shows the structure of a document.

5. Converting Hypertext to Linear Document

In the MUCH system, there are four classes of links: document, thesaurus, annotation and reference. The document link is the organizational link and the others are referential links. When a MUCH hypertext is converted into a text document, the nodes connected by document links are linearized and the links among them are implied in the hierarchical structure of the document.

In the MUCH system, a node has no type information. Its type is determined dynamically by the type of link pointing to it while traversing. In principle, the MUCH system allows a node to be the target of more than one pointer with different types. In this case, the node type

becomes variable depending on the traversal. This may cause some problems under certain circumstances. For instance, a node may be pointed to by both 'document' and 'annotation' links. When one wants to export that node as part of a document but the annotation link is visited prior to the document link in the traversal, the node will be treated as a comment node instead of a document node. When the exported document is printed, the content of this node may be lost.

The MUCH system is largely an authoring system, the document link is the central part of the hypertext and the other links are used to help authors to write better documents. To guarantee that documents can be exported properly, the document link is given highest priority and must be traversed prior to the other links. To realize this, a two pass traversal strategy is adopted. In the first pass, links other than document links are suppressed and a linearized document following document links is generated. The other link information and nodes are added to the document by the second pass.

Exporting a hypertext does not mean that one has to export the whole hypertext. Sometimes, exporting partial hypertext is useful, especially for remote collaboration. When many authors write the same book at different locations, a part of the document with comments, reference and other information may need to be sent back and forth among these authors. The S-M system is designed to support partial hypertext importing and exporting. For exporting, it allows users to specify from which node, following what criteria and how deep it will go. With all this information, the system can determine and export a subgraph of the hypertext network.

6. Converting SGML Document to MUCH

The import function of the SGML-MUCH system is to import an SGML document, which is either a document exported from the MUCH database or one prepared as a normal text document. Converting text to hypertext involves decomposition of the document and establishing hyperlinks among outline headings based on its structure. In addition to the links among decomposed nodes, some other links also need to be established, such as bibliographic citations, indices, cross-references and footnotes.

The import program of the S-M system accepts two kinds of linear document: one is a normal text document marked up in various formats such as LaTeX, Troff, ATK (Andrew ToolKit format), GNU texinfo, Practitioner Questionnaire format and plain text with numbered headings; and the other is documents exported from the MUCH system complying to the structure of the SM DTD shown in Figure 5. Based on the simplified SHyD model, text documents in various formats need to be formalized by

going through the SGML representation before being put into the hypertext database. This means that the text to MUCH conversion can be simplified to a conversion from an SGML document with SM DTD.

To generate hypertext from an SM DTD compliant document, the import program operates on the markup document to generate an abstract hypertext as follows:

- * The document is parsed and a tree structure of the outline is reproduced in accordance with the structured document.

- * For each pointer to a bibliographic citation or a footnote, the complete citation or footnote is located and associated with the pointer.

- * Each author-defined index term is copied into an index node. Next the index terms of that node are sorted alphabetically, and a link is created between each index term in a text block and its corresponding entry in the index node.

- * For sections with "info" and/or "links" elements, usually in a MUCH exported document, the information in the "info" element is resolved and extra links not implied in the linear representation are created. In case of normal text documents which have no "info" or "links" elements, default information will be used to fill the attributes of node-info components.

When importing a document which was exported from MUCH, the import program must guarantee that all the information that was exported can be restored without corrupting the document database. While importing the exported document, conflicts between the imported document and the existing document database may occur. For example, some nodes of a particular document may have been changed since the last export. In these cases, when the exported document needs to be imported again, the import program will check and try to solve these conflicts. The import program also allows users to decide how to update a document database with an imported document.

7. Converting SGML Documents to Other Hypertext Systems

The MUCH document database is a macro literary hypertext which is not suitable for hypertext distribution. However, the S-M system can automatically generate hypertext for several browsing systems, including Guide and Hyperities, from a linear document by going through the S-M representation. With this function, hypertext interchange between MUCH and other hypertext systems can also be realized.

Several hypertext systems define a specialized, or hypertext, markup language so that they can read an ap-

propriately marked file and generate their hypertext database. To convert a text file into a hypertext is actually to convert it to the hypertext markup language. To interchange information between MUCH and other hypertext systems, our approach is to export (a part of) the database into an SGML document and convert the exported document into the individual hypertext markup languages. This approach to hypertext interchange has the following features:

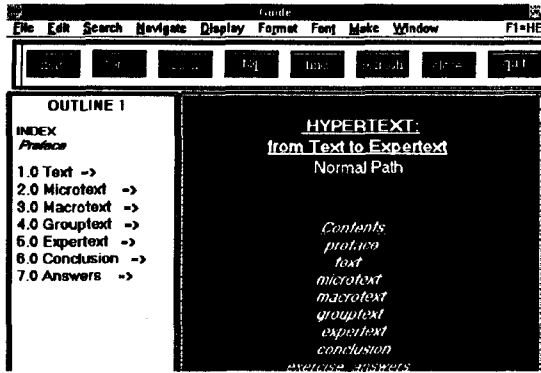
- * It is based on standard representations and models. The hypertext interchange mechanism of the MUCH system employs SGML and the Dexter model. Planned extensions are to HyTime and/or MHEG for hypermedia interchange.

- * It combines text-hypertext bi-directional conversion and interchange into one application. The structure of text and hypertext documents can be represented in a unified, standard representation (S-M DTD) without interference between them in individual applications. Utilities for conversion between SGML and MUCH, SGML and individual hypertext systems, and between SGML and individual text systems enable the information in one system to become interchangeable with several other text and hypertext systems.

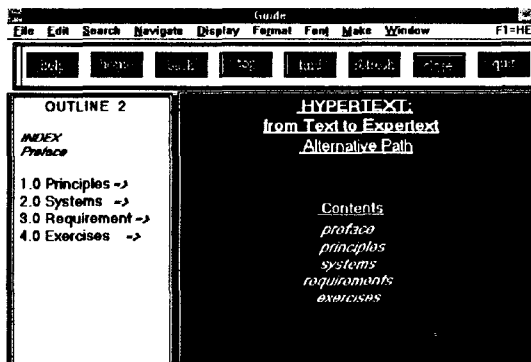
The hypertext generation may involve first-order and second-order hypertext. First-order hypertext directly reflects the original structure of a hypertext. For a hypertext generated from an SGML document, the first-order hypertext is such a hypertext, in which every link is explicitly present in the document. For many documents, the most significant first-order links are those connecting outline headings, but many other first-order links occur, such as those involving bibliographic citations, indices, footnotes and cross-references. Second-order hypertext, on the other hand, is usually derived from first-order hypertext. It adds links to the hypertext which are not explicit in the original form. For instance, the second-order hypertext may present new outlines or links computed from word patterns.

In some books, one can see a recurring pattern. For instance, in our textbook *"Hypertext: from Text to Expertext"* [Rada91b], each chapter (Microtext, Macrotext and Groupertext) has a section on 'History', 'Principles', 'Systems' and 'Exercises'. Thus one can envision another outline; rather than Text, Microtext, Macrotext, Groupertext and Expertext, the main topics of the book can be History, Principles, Systems and Exercises (Figure 6).

The alternate outline described above includes links which do not exist in the original outline. The different outlines or views of the document may not be equally cohesive when read sequentially from beginning to end, but may help some users in some tasks.



(a)



(b)

Figure 6. The normal (a) and alternate (b) outlines of the textbook on Guide.

In the second-order hypertext, new indices can be automatically computed based on the content of the text blocks. The derived index which connects link objects that have similar distributions is especially helpful when creating a large hypertext. With the derived indices, on selecting a derived index link, a user can get a list of the related terms and the numbers of the text blocks in which they occur. By selecting one of these text block numbers the user will be taken to the selected text block.

The automatic indexing can be done with a knowledge of natural language or based on word frequency, we chose word frequency one [Mhashi90]. Word patterns have been used by several research groups to create links between terms in different text blocks [Kaplan90]. These links may be used in searching for a particular fact or in browsing for associations that were not made explicit by the author [Bernstein90]. A simple semantic net can be generated with the word pattern approach and further supports searching or browsing [Mili85].

8. Implementation

The S-M system, including a set of conversion utilities, has been implemented in the Andrew multimedia environ-

ment running under Unix. To develop these data transformation tools, a public domain toolset for generating data translators called ICA (Integrated Chameleon Architecture) [Mamrak92] was used. The ICA development toolset automatically generates translation code based on high-level specification for data translation. ICA assumes the intermediate-form model (Figure 11), the structure-oriented approach and the embedded markup data model, which is coincident to our SHyD model which uses a structured, embedded markup language (SGML) as the intermediate format. It is therefore helpful in the development of our conversion and interchange facilities.

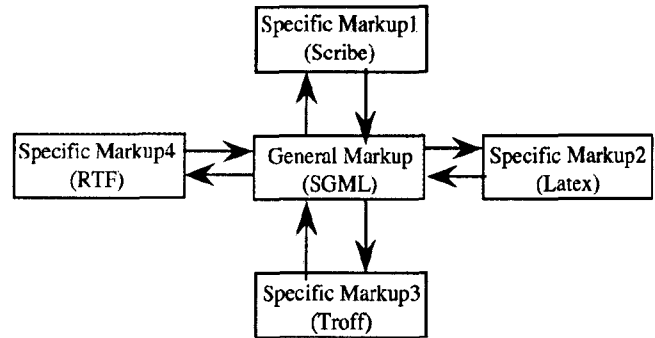


Figure 7. The intermediate-form model of data translation of ICA [Mamrak92].

With the ICA toolset, the development of converters in the S-M system becomes a systematic procedure and the only thing one needs to do is to define a context-free grammar (SM DTD) and mapping tables for the various markup languages, including the hypertext markup languages. However, because the ICA toolkit was designed largely for translating data between text markup languages which are static and context-free, extra work will be required to develop converters involving hypertext markup languages which are dynamic and have context-related features. For example, in the Guide markup language GML (Generalized Markup Language), there are several markups for establishing different links including reference link, command link, note-definition link and pictures. All of these markups are in the form of:

```
:source_command_name Identifier, parameters, and
:destination_command_name Identifier.
```

The identifier of each markup is the only way to connect the source and destination of a link, therefore it must be unique throughout the whole document. ICA can automatically map static strings from one format to another but it cannot handle a string like Identifier whose value depends on the values of other identifiers. In addition to the context related components, ICA cannot handle materials which are dynamically generated whilst converting, such as "Table of Contents (TOC)" and "Index list" which include links to different locations. To generate browsing

hypertext, by using the ICA toolset to increase productivity, we adopted a hybrid approach, namely using the ICA for format conversion and adding several perl scripts to deal with the dynamic and context-related materials.

The S-M system, along with the MUCH system, has been used in our daily work for document processing. Each member of the group can put his documents in various formats into the MUCH database and share these documents with others. Several textbooks have been written using the MUCH system and then translated into hypertext versions for Guide and Hyperties using the S-M system.

The programs which automatically generate derived indices and semi-automatically generate alternate outline have been implemented and applied to the Guide, Hyperties and Emacs-Info version of the textbook *Hypertext: from Text to Expertext*.

9. Conclusion

This paper presented an SGML-MUCH (S-M) system for establishing a connection between text and hypertext. In general, bi-directional conversion and hypertext interchange are considered as two different applications. A text document converted from hypertext may have different structures for different purposes. For processing and printing hypertext with a general document preparation system, the text document can be obtained from linearization of the hypertext in a hierarchical structure with components like chapter, section and subsection. This structure may lose some information included in the hypertext. To convert a hypertext into text as a hypertext interchange format, the text file needs to carry special information which is not normally conveyed in a traditional document, such as the author and date for a particular section and the referential links missed in the hierarchical structure.

The S-M system uses an SGML DTD which can represent hypertext structure and, at the same time, maintain the structure of linear text documents. Using this DTD as a standard representation, hypertext can be exported to an SGML document without losing any information. The generated document can be processed by a normal document processing system. Once this document is imported back into the hypertext system, all information can be recovered. Because the generated document bears hypertext information, hypertext for other hypertext systems such as Guide and Hyperties can be automatically generated. In a newly generated hypertext, in addition to those links explicitly provided in the generated document, more links can be generated through computing. We have experimented with the (semi-) automatic generation of alternate outlines and derived index, and got some promising results. By making use of the hypertext features,

one could explore more functions for dynamic document generation.

The current version of the S-M DTD is more or less linked to the MUCH model. When some other systems with different underlying models are considered, this DTD may have to be modified. To support system independent hypertext interchange, there are several unsolved problems which deserve further study.

Acknowledgement: This work would not have been possible without the support of the Commission of the European Communities under DELTA Project D-2006 OSCAR (an Open System for Collaborative Authoring and Reuse) nor without the help of our colleagues on the OSCAR team in Liverpool.

References

- [Berners-Lee92] T. J. Berners-Lee, R. Cailliau and J.-F. Groff, "The World-Wide Web", *Computer Networks and ISDN Systems*, 25, 454-459, Noth-Holland, 1992.
- [Bernstein90] Mark Bernstein, "An Apprentice that Discovers Hypertext Links", *Hypertext: Concepts, Systems, and Applications*, ed. A. Rizk, N. Streitz and J. Andre, pp. 213-223, Cambridge University Press, Cambridge, England, 1990.
- [Borenstein90] N.S. Borenstein, "*Multimedia Applications Development with the Andrew Toolkit*", Prentice-Hall Inc., 1990.
- [Chignell91] Mark H. Chignell, Bernd Nordhausen, J. Felix Valdez and John A. Waterworth, "The HEFTI Model of Text to Hypertext Conversion", *Hypermedia*, 3(3), 187-205, 1991.
- [Congnetics92] Congnetics Corp., "*Hyperties 3.1, User's Guide*", Congnetics Corporation, Princeton Junction, New Jersey, 1992.
- [Conklin87] J. Conklin, "Hypertext: An Introduction and Survey", *Computer*, 20(9), 17-41, Sept, 1987.
- [EBT90] EBT, "DynaText: Electronic Book Engine from EBT (Electronic Book Technologies): First to Handle any SGML Application", *Seybold Report on Publishing Systems*, 20(2), 18-22, September, 1990.
- [Furuta90a] Richard Furuta, Catherine Plaisant, and Ben Shneiderman, "Automatically Transforming Regularly-Structured Linear Documents into Hypertext", *Electronic Publishing*, 2(4), 211-229, December 1990.
- [Furuta90b] R. Furuta and D. Stotts, "The Trellis hypertext reference model", *Proceedings of the Hypertext Standardization Workshop*, pp. 83-93, Washington DC., 1990.

- [Glushko89] Robert K. Glushko, "Transforming Text into Hypertext for a Compact Disk Encyclopedia", *Proceedings of CHI'89*, pp. 293-298, ACM press, New York, May, 1989.
- [Goldfard90] C.F. Goldfarb, "The SGML handbook", Oxford University Press, Oxford, 1990.
- [Halasz90] Frank Halasz, and Mayer Schwartz, "The Dexter Hypertext Reference Model", *Proceedings of the Hypertext Standardization Workshop*, pp. 95-134, Washington DC., 1990.
- [ISO/IEC88] ISO/IEC, ISO/IEC 8879:1986 - A1:1988 (E). "Information Processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML)", Amendment 1. Published 01-07-1988, International Organization for Standardization, Geneva, 1988.
- [ISO/IEC92] ISO/IEC, ISO/IEC 10744:1992. "Information Technology -- Hypermedia/Time-based Structuring Language (HyTime)", International Organization for Standardization, 1992.
- [Kaplan90] Simon Kaplan, and Yoelle Maarek, "Incremental Maintenance of Semantic Links in Dynamically Changing Hypertext", *Interacting with Computers*, 2(3), 337-366, 1990.
- [Lange90] D. Lange, "A formal model for hypertext", *Proceedings of the Hypertext Standardization Workshop*, pp. 145-166, Washington DC., 1990.
- [Leggett91] John J. Leggett, and Ronnie L. Killough, "Issues in Hypertext Interchange", *Hypermedia*, 3(3), 159-186, 1991.
- [Mamrak92] S.A. Mamrak, C.S. O'Connell and J. Barnes, "Technical Documentation for The Integrated Chameleon Architecture", Department of Computer and Information Science, The Ohio State University, 1992.
- [Mhashi90] Mahmoud Mhashi, Roy Rada, Geeng-Neng You, Akmal Zeb, Antonios Michailidis, and Hafedh Mili, "Word Frequency Based Indexing and Authoring", *Technical Report*, Department of Computer Science, University of Liverpool, 1990.
- [Mili85] Hafedh Mili, and Roy Rada, "A Statistically Built Knowledge Base", *Proceedings of Expert Systems in Government Conference*, pp. 457-463, IEEE Computer Society Press, October 1985.
- [Newcomb91] Steven R. Newcomb, Neill A. Kipp, and Victoria T. Newcomb, "The 'HyTime' -- Hypermedia/Time-based Document Structuring Language", *Communication of the ACM*, 34(11), 67-83, November 1991.
- [Nunn88] D. Nunn, J. Leggett, C. Boyle, and D. Hicks, "The REXX project: A case study of automatic hypertext construction", *Technical report TAMU 88-021*, Hypertext Research Lab., Texas A&M University, College Station, Texas, 1988.
- [Rada91a] R. Rada, A. Zeb, G. N. You, A. Michailidis, and M. Mhashi, "Collaborative Hypertext and the MUCH System", *Journal of Information Science*, 17, 191-196, 1991.
- [Rada91b] Roy Rada, *Hypertext: From Text to Expertext*, McGraw-Hill, London, 1991.
- [Riley90] V. Riley, "An interchange format for hypertext systems: the Intermedia model", *Proceedings of the Hypertext Standardization Workshop*, pp. 213-222, Washington DC., 1990.
- [Ritchie89] Ian Ritchie, "HYPERTEXT -- Moving Towards Large Volume", *The Computer Journal*, 23(6), 516-532, December 1989.
- [Ritchie92] Ian Ritchie, "New Horizons for Hypermedia", *invited lecture at Hypermedia '92*, ed. Theo Bothma, Pretoria, South Africa, March 18-20, 1992.
- [Salton83] G. Salton, and M. McGill, "Introduction to Modern Information Retrieval", McGraw-Hill, New York, 1983.
- [Sarre90] F. Sarre, M. Seidt, and U. Guntzer, "HyperTex - A System for the Automatic Generation of Hypertext Textbooks from Linear Texts", *Proceedings of Database and Expert Systems Applications Database and Expert Systems Applications*, pp. 62-68, ed. K. Wagner, Springer-Verlag, 1990.
- [Yankelovich88] N. Yankelovich, B. J. Haan, N. K. Meyrowitz, and S. M. Drucker, "Intermedia: The Concept and the Construction of a Seamless Information Environment", *Computer*, 21(1), 81-96, Jan. 1988.
- [Zheng93] M. Zheng and R. Rada, "SHyD -- A Model for Bridging Text and Hypermedia", *Proceedings of 1993 ACM Computer Science Conference*, pp. 418-424, Indianapolis, Indiana, February, 1993.