Automatic creation of hypertext networks from technical documents

Fabrice Papy and Jean-Louis Vuldy
Electricité de France-Direction des Etudes et Recherches
1, avenue du Général de Gaulle
92141 Clamart, FRANCE

Email: fabrice.papy@ici.der.edf.fr

Email: jean-louis.vuldy@der.edf.fr

Tel: (33-1) 47 65 38 85

Fax: (33-1) 47 65 35 23

Abstract: This paper describes a practical method to automatically create hypertext networks from technical structured documents. The physical structure of the documents allows to cut them in nodes when the crossed references determines associative links. The differents steps of the processing line are presented.

Keywords: hypertext networks, structured documents, physical structure, logical structure, Rich Text Format, Backus Naur Form, nodes, links, crossed references.

1 Introduction

Most works and softwares about hypertext try to show the functionalities of such information representation way (navigation, association, reader's orientation...).

Authoring systems (Hypercard, Toolbook,..) supply various tools providing information or allowing to import it from others environments via specific filters (graphic and text converters).

Once information created or imported, authors have to design a subjective empirical model for representing their information in the form of a hypertext.

We just describe a generic process to create automatically hypertext thanks to the physical structure of documents without proposing a new authoring system or a methodology for designing hypertext. [Rada 90] [Rada 91] [Rada 92].

Indeed, althougt hypertext is not clearly defined, researchers often qualify it in words of informations nodes and links, permitting to navigate from one node to another [Conklin 87] [Garg 88] [Nielsen 90a].

Therefore, we have implemented a processing line where these basis components are produced, then sent to an hypertext system which creates hypertext networks. This processing line we describe is made of three elements as follows:

- an automaton which transforms every structured document into information units,
- a crossed reference parser which identifies explicit links between nodes,
- an authoring system (Hypercard, Toolbook) which uses nodes and links to generate the final hypertext network.

2 Structured documents

The documents we automatically process are essentially technical text document about electrical power engineering. These documents have got a logical and physical structure.

By logical structure, we mean a set of abstract entities like titles, chapters, sections, paragraphs,.. defined into author's (and reader's) mind and commonly used into many types of document (newspaper articles, letters, reports, manual..) [Furuta 89] [Furuta, Stotts 89].

The combination and the diversity of such entities can obviously differ from a document to another. For example, a letter may use certain types of entities when a softaware reference manuel would use others.

Logical structure being higly linked with the content (and the semantic so) of the document, it is the cognitive mechanism of reading which permits the type of the logical entities to be identified (see figure 1).

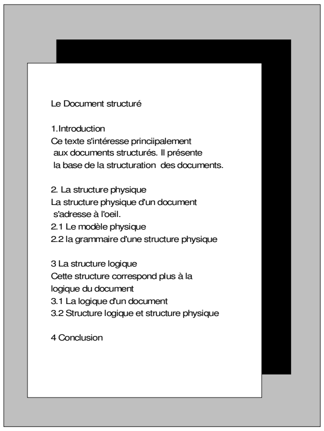
The physical structure of a document associates each logical entity with a graphic representation and gives their location upon the page [Ingold 90] [Derrien *et al*. 89]. Text processing softwares (and printing peripherals) offer many options such character set, size, margins, ligne spacing... to realize text formatting.

Consequently, using a particular text formatting allows to identify or to characterize a type of logical entity [Virbel 87].

But that is right for complex structured documents where numerous types of logical entities are used. Indeed, the formatting of a press article, a letter or a novel, even if it permits to identify the nature of the logical entity, gives little information about its characterization (we can deduce that the paragraphs which are into the upper part of the letter correspond to receiving or dispatching zones, but what to say in words of characterization of differents paragraphs making up the body of the letter or paragraphs making up the article of a newspaper).

On the other hand, for the technical documents (user manual, reference guide, technical and scientific reports) which are often made up of different kinds of logical entities, formatting undisputably brings better readability of a document (see figure 2) and emphasizes the meaning of the various used entities.

Then it becomes possible to browse selectively a document, using the predefined form of logical entities as selection criterion. By memorizing the physical layout of a logical entity, we "glance through" the document, reading only the logical elements corresponding to this layout. [Southall 88].



Le Document structuré

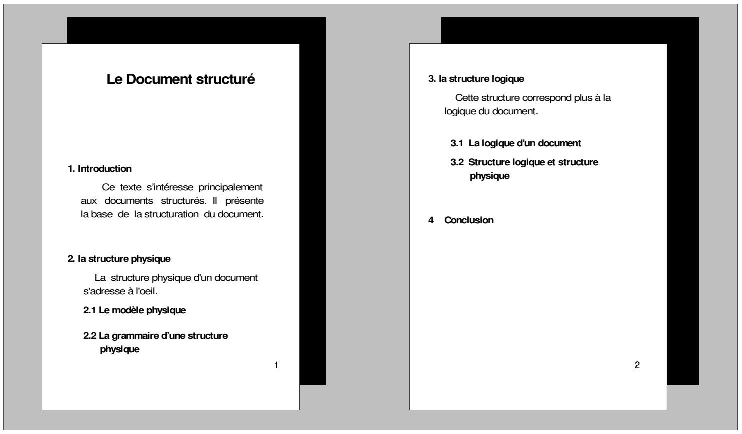
1.Introduction
Ce texte s'intéresse princiipalement
aux documents structurés. Il présente
la base de la structuration des documents.

2. La structure physique
La structure physique d'un document
s'adresse à l'oeil.
2.1 Le modèle physique
2.2 la grammaire d'une structure physique

3 La structure logique
Cette structure correspond plus à la
logique du document
3.1 La logique d'un document
3.2 Structure logique et structure physique

4 Conclusion

Figure 1: The logical structure of a document
Title, sections, chapters, paragraphs... are the logical entities of this document, it is the reading which permits their identification.



Le Document structuré

1. Introduction

Ce texte s'intéresse principalement aux documents structurés. Il présente la base de la structuration du document.

2. la structure physique

La structure physique d'un document s'adresse à l'oeil.

2.1 Le modèle physique

2.2 La grammaire d'une structure physique

1

3. la structure logique

Cette structure correspond plus à la logique du document.

3.1 La logique d'un document

3.2 Structure logique et structure physique

4 Conclusion

2

Figure 2: The physical structure of a document
The formatting of a document permits a graphical identification of logical entities.

Obviously, this remains true only for document whose instances of a same logical entity keeps the same physical layout. This introduces the notion of document physical model that permits to homogenize the physical layout of similar documents made up from the same kinds of logical entities. With the model of the document, each logical entity has then one and only one physical layout at disposal. The process of hypertext automatic creation from structured documents that we describe in this paper supposes that the documents have been structured in accordance with a defined physical model [Tazi 90].

These documents which are wiritten with different word-processors (Word for Windows, Word for DOS, Word for Macintosh...) have been all associated with a particular stylesheet, that stands for a model for the document.

The authors, when they create their documents, associate the adapted formatting with the logical entities. At the end of these operations, the formatted document contains real information along with the characteristics of formatting.

The Rich Text Format (Microsoft 90), is a format of a document proposed by Microsoft that permits to represent in the same document (in ASCII) the text and the characteristics of this text (formatting attributes are described by markers).

As we already said it, and specially with regard to technical documents, the physical structure characterizes the logical entities, so it is possible to handle these logical entities only by the physical layout. The technical structured documents we deal with, correspond schematically to a logical grammar (see table 1) and check also a physical grammar (see table 2) in accordance with their model of document.

TechnicalDocument::= DocTitle Text Sect1

Sect1::= Title Text Sect1 | Title Text Sect2

Sect2::= Title Text Sect1 | Title Text Sect2 | Title Text Sect3

Sect3::= Title Text Sect1 | Title Text Sect2 | Title Text Sect3 | Title Text Sect4

Sect4::= Title Text Sect1 | Title Text Sect2 | Title Text Sect3 | Title Text Sect4

Title::= *character string describing the section title*

Text::= *character string*

DocTitle::= *character string describing the document title*

Table 1

TechnicalDocument::= DocTitle Text Sect1

Sect1::= Sect1Title Text Sect1 | Sect1Title Text Sect2

Sect2::= Sect2Title Text Sect1 | Sect2Title Text Sect2 | Sect2Title Text Sect3

Sect3::= Sect3Title Text Sect1 | Sect3Title Text Sect2 | Sect3Title Text Sect3 | Sect3Title Text Sect4

Sect4::= Sect4Title Text Sect1 | Sect4Title Text Sect2 | Sect4Title Text Sect3 Sect4Title Text Sect4

DocTitle::= <CharacterSize 18 point><bold><left indent 1.4cm> <left aligned><double bottom border>

Sect1Title::= <CharacterSize 14pt> <bold> <left indent 1.4cm> <first indent -1.4cm> <left aligned> <single bottom border>

Sect2Title::= <CharacterSize 12pt> <bold> <left indent 1.4cm><first indent -1.4cm> <left aligned>

Sect3Title::= <CharacterSize 10pt> <bold> <left indent 1.4cm> <first indent -1.4cm> <left aligned>

Sect4Title::= <CharacterSize 10pt> <bold> <left indent 2.1cm> < first indent -2.1cm> <left aligned>

Text::= <first indent 0.7cm> <justified> <CharacterSize 10pt>

Table 2

3 The cutting of the document

"The cutting of structured documents in distinct blocks is not to be done through semantic means but, taking into account the physical structure. Defining, in a concrete way, an information unit it is to decid what is the undivisable piece of information" [Balpe 90].

It is from the physical structure and the RTF representation of styles that the program determines the information units. A direct correspondance exists between the type of the logical entity, its typographical attributes and its RTF identification. (see table 3).

To each document, we link a file that contains the cutting "rules" which in fact are the declaration of the RTF identifier, that permits to distinguish the different logical entities.

Using these rules, the cutting permits on one hand the adaptation to every kind of structured document from a model (initial work will consist in identifying the RTF markers assuring the physical formatting), and on the other hand a certain flexibility for the determination of the information unit.

This program using RTF documents as input, , delivers at the end of the process a set of files (the nodes) that came with a description which keeps the sequential link of reading of the initial structured document.

Logical item	**Typographic attributes**	**RTF identification**
DocTitle	<CharacterSize 18pt> <bold> <left indent 1.4cm> <left aligned><double bottom border>	\s4
Sect1Title	<CharacterSize 14pt><bold><left indent 1.4cm> <first indent -1.4cm><single bottom border>	\s251
Sect2Title	<CharacterSize 12pt> <bold> <left indent 1.4cm> <first indent -1.4cm> <left aligned>	\s252
Sect3Title	<CharacterSize 10pt> <bold> <left indent 1.4cm> <first indent -1.4cm> <left aligned>	\s253
Sect4Title	<CharacterSize 10pt><bold> <left indent 2.1cm> < first indent -2.1cm><left aligned>	\s254
Text	<CharacterSize 10pt> <first indent 0.7cm><justified>	\s2

Table 3

4 The location of links

Once the nodes are created, one needs to locate the common explicit links. These explicit links are the references in the text that authors integrate into their documents to direct the readers to another part of the current document or to another document that can provide further information about the approached subject .

Regarding technical and scientific reports, thesis and assimilated documents, the crossed references have been standardized by the ISO [ISO 82] [ISO 86].

In fact, this standardization of references in the text is not rigorously applied by the authors (nor by the organisms) who often adopt a hybrid codification, representing a compromise between the writing practices and the necessity to provide an unambiguous reference.

For example, within the particular framework of technical documents, relative to the field of electrical power, a grammar of references has been specially determined to find a paragraph, a picture or a mathematical equation within the documentation.

As this documentation consists of several manuals destined for various uses (administration, reference, use, validation,...) they are themselves made up of several parts composed of documents (themselves composed of chapters, sections, pictures, equation, spreadsheets,...). The following system of identification has been finalized; a letter identifies in a unical way a guidebook (U for use, A for Administration,...), an unical number (related to the manual) of two figures is associated with each part of this manual and all the documents of a part have an unique key (related to the part) of two figures. So referring to the fifth document of the third part of the user manual will be expressed U.03.05. Within a document, the recommendations of the ISO [ISO 78] have been adopted for numbering divisions and subdivisions.

Through this example, it is clear that an algorithm for the search of references based on the recommendations of the ISO could not adapt to the smallest variation in the codification of a reference.

We have finalized a program of analysis capable of dealing with different forms of references.

The basic idea consists in associating with each document the grammar of references expressed in a Backus Naur Form formalism.

The algorithm having to deal with the documents from this BNF grammar is a Top-down recursive algorithm [Tremblay 85].

We give here two examples of grammar to find back references of different kinds.

Example 1: Reference [U.01.10], which points the 10th document of the first part of User Manual.

```
Ref1::= CrochetOuvrant CorpsRef1 CrochetFermant
CorpsRef1::= Manuel Point Partie Point Document
Manuel::= "U" | "A" | "C"
Point::= "."
Partie::= Nombre
Document::= Nombre
Espace::= " "
Nombre::= Digit Nombre
Digit::= "0" | "1" | "2".|...|"9"
CrochetOuvrant::= "["
Crochet Fermant::= "]"
```

Example 2: Reference (see Chapter 1)

```
Ref2::= ParenthOuvrante CorpsRef2 ParenthFermante
ParenthOuvrant::= "("
ParenthFermante::= ")"
CorpsRef2::= Voir Espace Titre Espace Nombre
Voir::= "See" | "see" | "cf." | ...
Titre::= "Chapter" | "chapter" | "chap." | "Chap."
...
```

In the nodes where the algorithm cheks the grammar, corresponding values and location informations are given. These values are strings of characters corresponding to non terminal non recursive identifiers . The location informations gives for each identifier the line number, the beginning position and the length of the checked string.

In the case of example 1, we get the following results:

```
<ref1>
<CrochetOuvrant>::=5,10,1,[
<Manuel>::=5,11,1,U
<Point>::=5,12,1,.
<Partie>::=5,13,2,01
<Point>::.5,15,1,.
<Document>::=5,16,2,10
<CrochetFermant::=5,18,1,]
</ref1>
```

5 The final generation of network

The real layout of nodes and links in hypertext is entrusted to authoring systems such as Hypercard or Toolbook.

For this purpose, some basic scripts have been written on the one hand, to automate the nodes integration within Hypercard stacks and Toolbook books, and on the other hand to assure a minimal associative navigation (next, previous, back, search, history..) within these networks.

6 Conclusion

So far, the cutting of several documents in information nodes from the physical structure has given satisfactory results. A modification of the cutting algorithm (that now works in a assertional way; a node is created as soon as certain markers are found) is yet to be envisaged to solve the stylization problems (end-of-paragraph character, bad use of styles...) and to make cutting more contextual.

Nowadays, we are finishing to integrate at the level of the definitive hypertext network, the detected crossed references.

Inevitably, the automatic creation of hypertext raises the problem of the replacement of modified documents and the links updating. To avoid the overall replacement of all generated hypertext, we have envisaged a solution inspired from the relational databases theory.

Indeed, the cutting of the document in nodes and the detection of reference links display functional dependances among the nodes. Therefore, it is then possible to have a relational representation of nodes and links in conformity with normal forms defined by E.F. Codd.

So, borrowing from relational databases management system the data integrity checking techniques, the crossed reference coherence could be maintained whatever the updating operations are done.

Bibliography

[Balpe 90] Balpe J-P., "Hyperdocuments, hypertextes, Hypermédias", Eyrolles, 1990.

[Conklin 87] Conklin J. "Hypertext: an introduction and survey", Computer, vol. 20 n°9, Sept. 87, 17-41.

[Derrien *et al.* 89] Derrien D., Bouchitté V., Habib M., "Approche objet pour l'analyse de la structure logique des documents", Workshop on Object-Oriented Document Manipulation, Rennes France 29-31 mai 1989, Bigre n°63-64, mai 1989, 226-235.

[Furuta 89] Furuta R., "Concepts and models for structured documents", in Structured Documents, edited J.André, R. Furuta and V.Quint, Cambridge University Press, 1989, 7-38.

[Furuta, Stotts 89] Furuta R., Stotts P.D., "Objects structure in paper document and hypertexts", Bigre 63-64, mai 89, 145-151.

[Garg 88] Garg P.K. "Composition of hypertext nodes", Learned Information, vol. 11, 1988, 63-73.

[Ingold 90] Ingold R., "Reconnaissance de la structure logique d'un document par une méthode d'analyse descendante", Bigre 68, mai 1990, 26-34.

[ISO 78] "Documentation - Numbering of divisions and subdivisions in written documents", 2145, 1978-12-15.

[ISO 82] "Documentation - Presentation of scientific and technical reports", 5966, 1982-03-15.

[ISO 86] "Documentation - Presentation of theses and similar documents", 7144, 1986-12-01.

[Microsoft 90] Microsoft Corporation, "Microsoft: Rich Text Format Specification", 1990

[Nielsen 90a] Nielsen J., "Hypertext and Hypermedia", Academic Press, 1990.

[Nielsen 90b] Nielsen J. "The art of navigating through hypertext", Comm. ACM vol. 33, n°3, March 1990, 297-310.

[Rada 90] Rada R., "Hypertext writing and document reuse; the role of a semantic net", Electronic Publishing, vol.3 n°3, Aug. 1990, 125-140.

[Rada 91] Rada R., "Hypertext and paper: A special synergy", International Journal of Information Management, 11, 1991, 14-22.

[Rada 92] Rada R., "Software reuse: from text to hypertext", Software Engineering Journal", Sept. 92, 311-321.

[Smith, Weiss 88] Smith J.B., Weiss S.F. "Hypertext", Comm. ACM vol. 31 n°7, July 88, 816-819.

[Southall 88] Southall R., "Visual structure and the transmission of meaning", Cambridge University Press, 1988, 35-45.

[Tazi 90] Tazi S., "Aide à la structuration des documents pour les systèmes hypertextes", Bigre n°63-64, mai 1990, 18-25.

[Tremblay, Sorenson 85] Tremblay J-P., Sorenson P.G., "The theory and practice of compiler writing", McGraw Hill Computer Sciences Press, 1985

[Virbel 87] Virbel J., "L'apport de connaissances linguistiques à l'interprétation des structures textuelles", Bigre n°53, Mai 1987, 77-97.