ELECTRONIC PUBLISHING, VOL. 7(1), 21–34 (MARCH 1994)

A hypertext electronic index based on the Grif structured document editor

HÉLÈNE RICHY

Opera Project, IRISA/CNRS
Campus de Beaulieu
F–35042 Rennes cedex, France

SUMMARY

This paper presents an electronic index service that was developed in the Grif editor by taking advantage of the hypertext facilities available in the system. Grif is a structured document editor based on the generic structure concept that supports both hierarchical structures and non-hierarchical links.

The active cross-reference within the Grif index makes activation and browsing through indexing more powerful than in other systems: the index tables, helpful as a medium for supporting search by keywords in paper documents, support browsing in electronic documents. These indexes are easy to use as they are displayed in the same form as indexes in a paper document.

KEY WORDS Grif Hypertext Index Structured document

1 INTRODUCTION

In ordinary books, an index allows a reader to access essential information easily. With electronic documents, it is also useful and natural to browse a list of terms instead of a whole document, which may be of considerable length. Our goal in developing an electronic index was to combine the advantages of traditional paper indexes and hypertext facilities.

The Grif direct-manipulation document editor transforms a static index into an active electronic one. This index offers new features for navigation into a document that do not exist in static printed material. This feature helps the author to make his document a hypertext in which the reader will be guided by electronic links as well as by the traditional structured form of the document.

The resulting index is more precise than a paper index, because it does not only indicate a page number, but it also points at the related passage within the structured document; this passage may be either a sentence, a few paragraphs, or a whole chapter as well. This passage is displayed in a separate window showing a part of the document.

This paper reports on the experience gained in using hypertext functions in an active electronic index in the context of a system for structured documents. The next section reviews the problem of building and using indexes and presents related work. The third section presents the main principles of the Grif system. Section 4 focuses on the index features and the user interface. The fifth section analyses the first experiments with the electronic index, and the last section proposes, as a conclusion, further investigations.

CCC 0894–3982/94/010021–14

Received 26 February 1993
Revised 7 October 1993

2 WHAT IS AN INDEX?

An index is an aid that facilitates quick location of relevant passages within large documents, such as books, technical reports and manuals. It consists of an alphabetical list of terms[1] in which all the most important subjects appear. Every term, sometimes specified by a gloss,[2] is followed by a list of references to the pages where more details about the subject can be found (see Figure 1).

L
Links **5–6**, 8, *see also* Hypertext
presentation *5–6*
type 5

S
Semi-automatic indexing *see* Marking tool
SGML, *Standard Generalized MarkUp Language* 4

Figure 1. Excerpt from an index table

2.1 The index in paper documents

Before the use of electronic publishing systems, the process to perform when indexing a document was long and complex. *The Chicago Manual of Style* [1] identifies no fewer than five steps in the mechanics of indexing. Traditionally, an index was made from paginated handwritten documents in which the author had underlined the passages that he considered important. Marginal notes indicated terms or headings that may not have appeared in the text.

From the page proofs individual cards were prepared, one item on a card, and alphabetized. When all headings and subheadings had been chosen, and suitable cross-references had been added, the cards were edited into final form. Then the entire index was typed following strict presentation rules to ensure the legibility of the resulting index.

2.2 The index in electronic publishing

Using electronic publishing facilities, it is now possible to quickly compute indexes when pagination has changed. The page numbers are automatically updated by the index processors. It is possible, but not very comfortable, to make corrections after reading the first draft of the index produced. In addition there is still a sequence of tasks involved in preparing an index for a document to be printed. A preliminary task is to decide what to index and which indexes to prepare so that the readers of the index can find what they are looking for. The next task is to prepare the document for indexing: to locate each instance of a word or group of words in the document to be indexed and to associate the proper indexing term. Finally the last task consists in producing and printing a properly sorted and formatted index from this set of terms and locations in the document.

[1] Each term may be a word or a group of words.
[2] A gloss is an explanatory comment added to a text.

The preliminary task requires reflection. At best, the computer can provide only partial assistance to propose indexing terms. Abe and Berry's `findphrases` program [2] may help in determining the list of phrases that will appear in the index. However, there is no way they can substitute for a human indexer.

Most editing systems help to perform the second task. Using FrameMaker [3], Microsoft Word [4], or a formatting program such as LaTeX [5], the preparation for indexing consists of inserting a *mark* and some additional information before each relevant passage within the document. Any editing system offers a command to search for instances of words. A straightforward syntax is used to describe these marks, the typography of the entry, the associated index terms, etc. Much effort has been devoted to help in producing indexes within electronic documents [6–10]. For instance, the `Indx` tool described by Abe and Berry [2] locates the terms within a document to be indexed from a list of indexing terms.

At this step, the production of the index can be performed electronically. The index processor collects the multiple instances of index terms into a single list of locators (page numbers) sometimes compressing adjacent page numbers, sorting the entries in alphabetical order, formatting the entries and the subentries, and inserting cross-references at appropriate places.

Generally, the index produced by these electronic publishing systems is not completely satisfying. Indeed it can require much additional work before a good index is produced. It is still necessary for a human to spend time eliminating useless page references, modifying the typographical style, changing some headings and subheadings, and adding or suppressing cross-references and glosses. Even so, many index tables produced by electronic publishing systems are still inconsistent. As an example, most index processors do not check the validity or formatting of cross-references: in the index tool of Microsoft Word, cross-references are considered as comments. Thus, even though cross-references may be included in the index (perhaps mixing "*see*" and "*cf.*"), no check is made to verify that the referenced entry really exists in the index tables.

2.3 The index in electronic reading and hypertext

Any hypertext offers to the user the ability to organize and to retrieve information in a non-sequential fashion, by creating, linking and manipulating information in a directed graph of nodes, interconnected by typed links [11,12]. Additional features are also provided, such as facilities for browsing, navigating and searching in the hypertext, attaching of scripts and attributes to nodes and links, etc. But the basic mode of interaction with hypertext systems is navigational. As an example, the knowledge-based system proposed in [13] adds a browsing facility while creating contextual links in hypertext; indices are modified incrementally, but are not automatically deleted.

Although hypertext is now used for many applications, hypertext and traditional documents are not really integrated. In addition, the problem of efficient search and retrieval over large hypertexts is not easy to solve. When existing in an editing system, the hypertext facility consists in inserting buttons that are used for linking pages in a document or for linking other documents or graphics. Unfortunately, because the document is not deeply structured, these editing systems cannot automatically link elements such as captions, cross-references, index tables, and indexing marks. Even when the text editor includes hypertext functionality, the functionality is insufficient for generating indexes. An interesting approach has been proposed in [14] where the authors have shown how to generate links

from existing documents based on user specifications. A specification language describes the programmed generation of any kind of links, and among them are the index links. This is a very promising approach and should be integrated into a structured text editor.

The approach we propose in Grif considers indexes as structured elements within a structured document. The existing languages are used to describe the structure, the links, and the presentation of the index. The index application has required extensions to the Grif editor for building the index tables. The ordinary Grif editor can be used for updating and for navigating through the document.

3 MAIN PRINCIPLES OF THE GRIF SYSTEM

Grif is a system that combines the main features of hypertext and structured documents [15]. Thus, it provides integrated and interactive support for the development of active structured documents. As has already been noted by several authors [16–18], combining electronic documents and hypertexts is a very promising approach.

3.1 Document features in Grif

Grif is an interactive editor for structured documents. In Grif a document is considered in the same way as in SGML [19,20] or ODA [21]: it is represented as an abstract (logical) structure that combines elements of different types. With the SGML version of Grif, the user writes a DTD (document type definition) in the SGML language; with the native version, he uses the S language, which is functionally equivalent to SGML, with some additional features.

The Grif editor, driven by the DTD, guarantees that the final document is structurally correct; each document is represented in the system by its logical structure, which is an instance of a generic logical structure [22]. All documents with the same generic logical structure belong to a document type, such as the type *Report*, *Letter*, *Chapter*, etc. For instance, the definition of a *Report* says that the document must contain a title, an author name and a variable number of chapters and that each chapter must contain a heading and at least two sections.

The graphical aspect of a document type is defined by *presentation rules* that are grouped into a *presentation model*. The image of a document displayed on the screen or printed on paper is elaborated from its logical structure, following a presentation model. Even with the same DTD, documents may have to look different under different publishing requirements. Thus, various presentation models may define the output style of structured documents of the same type.

The graphic representation of a document is usually made up of several *views* defined in the presentation model. Each view is displayed in a different window. As an example, for a report, the presentation model defines a view for the whole text (*Full_text* view), another for the table of contents, and another for the bibliography. A feature that makes multiple views especially useful is *synchronization*. Clicking on the image of an element in any view causes other opened views to scroll in such a way that the same element is displayed in all views where it can be displayed. As an example, clicking on a section heading in the table of contents makes it possible to move immediately to the same heading in the body of the document.

3.2 Hypertext features in Grif

The tree structure is not sufficient for fully representing the document. Therefore it is possible in Grif to establish links between an element in a document and another element in the same document or in another document. The Grif editor allows any link to be embedded within the structured document. Of course, the generic approach offered by the DTD is also used for links and chunks, which are, like any other element, specified in DTDs. Thus, the document model is consistent and the hypertext aspects of it are homogeneous; the notion of logical structure encompasses both hierarchical structures (as usual in a structured document) and nonhierarchical links (as usual in hypertexts).

Types of links

Constraints imposed by DTDs on links concern only the types of the origin and of the target. For instance, a reference to a section (a *RefSect* element) can point to an element in the same document or in another document provided that element is of type section. They are not related to the notion of document. The type of the target element can be restricted to a unique type, but it can also be a choice among a list of types or even any type. Links can be either internal or external. A link is internal when the origin and the target belongs to the same document. Otherwise, a link is external.

All links available in Grif are directed. They can be represented either by elements or by attributes of the type *reference.* As in SGML, an attribute is information associated with an element which provide additional semantics to that element. The type *reference* represents a link between the element with which the attribute is associated and another element. But when there is no element with which the attribute could be associated, the link is represented by an element: a terminal element in the hierarchical structure of a document.

Like any other elements, reference elements are specified in the DTD, which indicates where they can appear in the hierarchical structure and which type of element can be the target. As an example, a *Paragraph* must contain a sequence of basic elements which may be either a character string (*Text*) or a reference to a section (*RefSect*), or a reference to a figure (*RefFig*).

Presentation of links

The links are divided into two main categories: *cross-references* and *inclusions*. A cross-reference establishes a relation between two elements and does not modify these elements. Only cross-references defined in the DTD can be used in a document. As part of the logical structure, they are presented according to the chosen presentation model. For instance a *RefSect* is displayed as a section number.

On the contrary, inclusions do not need to be specified in the DTD; inclusion is an operation offered by the editor. Inclusions allow information to be shared, the same information being available at both the origin and the target. All changes made in the target element are automatically reflected in the copy. An inclusion is an actual copy of the target element, which takes place of the origin element when the document is displayed or printed. This does not mean that the copy looks exactly like the target: if the environment of the inclusion is different from the environment of the target, its graphical aspect is usually also different, because most presentation rules use inheritance.

To allow the user to take advantage of links, they must be clearly visible. For that purpose, the origin of all links is displayed in a specific colour.[3]

User interface

The explicit declaration of both structure and presentation of structured documents has a number of advantages [16,23,24]. One important issue is that all editing commands take links into account and make sure that they always satisfy their definition in the DTD. When an element that is the origin or the target of a link is moved to a different place, the link itself is automatically updated to follow the element. When deleting an element that is the origin of a link, the editor removes the link. When deleting a target element, it keeps the link and allows the user to assign another target to that link.

The origin of a link can be created with menus (either the Create menu or the Attribute menu) which propose only correct options: the user first creates the origin of the link and then clicks on the target element. The target element is accepted only if it has the type specified in the DTD. By double-clicking an element that is the origin of a link, the user asks for the target of that link: the editor displays and highlights the target element. If it is not visible in any opened view, the editor opens a view displaying it. In the case of an external link, the editor loads the target document and opens a view to show the target element. More details on these hypertext features can be found in [15].

4 INDEXING FEATURES IN GRIF

The electronic index service has been developed by taking advantage of the hypertext facilities provided by the editor. This application handles the kind of indexes presented in Section 2. An index table is a list of terms, sorted in alphabetical order, which may have any number of entries and references.

4.1 Active indexes and structured documents

All index tables are structured documents of the same generic type. An index table contains both references to relevant passages within the document and cross-references to other entries in the same index table. A passage which is delimited for indexing purposes may be a single word, a sentence, a sequence of paragraphs, or a whole section as well.

Moreover active indexes are built using hypertext links. And so, they offer a new means to access the document content. They support activation and browsing. The user can take advantage of all the links (i.e., reference elements or attributes in the DTD of the index table) for moving across the electronic document; by double-clicking an entry in an index table, the document is scrolled directly to the required place. For instance, by double-clicking the entry *Grif* in the Index view shown in Figure 2 (on the left), the Full_text view is displayed (at the top) and the sentence *Grif is a structured ... links* is highlighted.[4]

Like any other structured element, the appearance of the index tables is defined by a *presentation model*. There may be several presentation models, each defining a different layout style: entries on separate lines and subentries in paragraph, variations in punctuation

[3] On monochrome displays, colour can be replaced by a pattern that simulates grey.
[4] This figure refers to a previous draft of this paper.

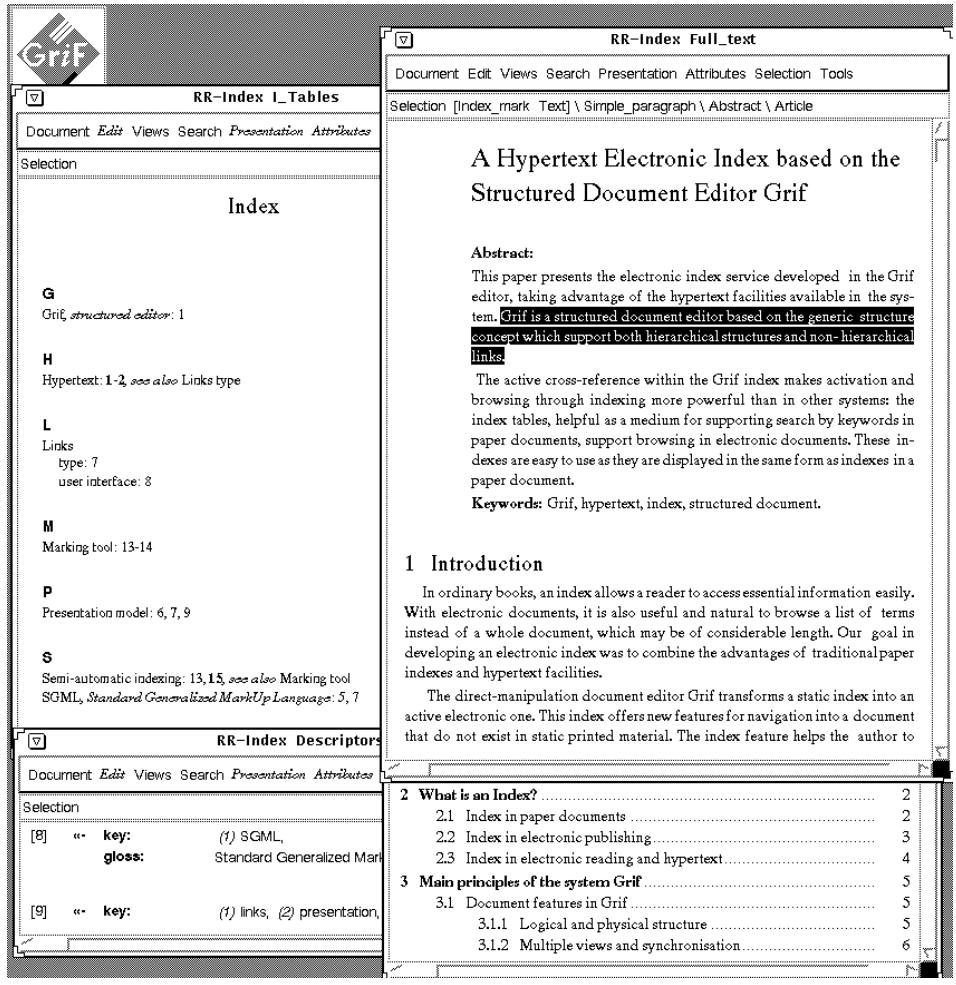

Figure 2. Several views for editing a document

(commas, periods, en-dashes), indention of lines, leading of lines, colour and display of the passage delimiters, etc. Consequently, the end-user can see the index table and the delimiters of the indexed passages, with different layouts depending on the type of work he or she is doing.

For a document with an index, the presentation model defines several views, a view for the whole text and a view for the index tables. Other views may help the author to check the index content (see Section 4.2 below).

4.2 Implementation of indexes with Grif

Generic structure extension

The generic structures defining document types are made extensible in such a way that they can accept the pieces of generic structure needed by an application. The elements and the

attributes required to handle the indexes are defined as a generic structure extension, called the *Index* DTD. Such an extension can occur dynamically. Thus, any kind of document may have index tables. The *Index* DTD (see Figure 4) defines a number of elements and attributes that are listed below.

Indexed passage delimiters

A paired component, called *Indexed_passage_delimiter*, is defined in the *Index* DTD. These paired components are used as passage delimiters. In many cases, several logical elements are concerned by an index entry. The delimiters comply with the logical structure of the document. The pairs are siblings. But several passages may overlap: in a sequence of three paragraphs, the first and the second may constitute a passage, referred by the index, and the second and the third may constitute another passage, referred by another index entry (see the paired components $\ll_2 \gg$ and $\ll_4 \gg$ in Figure 3).

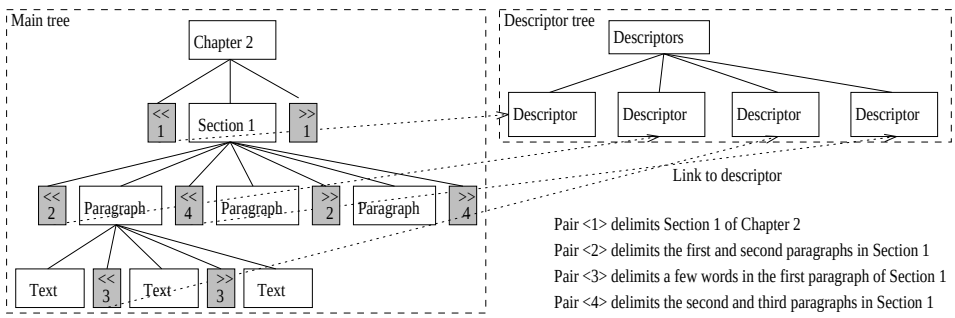

Figure 3. Delimiters in the document structure

An Indexed_passage_delimiter can be inserted around a previously selected passage within the structured document, by calling the Create menu. When an opening or closing paired component is selected, the Grif editor automatically selects the paired component. When the delimited passage runs over a series of consecutive pages, the Grif indexing system will display a range of pages in the index table.

The passages delimited by Indexed_passage_delimiters are linked with *Indexed_passage_descriptors*. This is achieved with reference attributes, called *Link_to_descriptor*, associated with each opening paired component. Figure 4 shows a piece of the *Index* DTD that specifies some of the elements and some of the attributes needed by the electronic index using the S language (some simplifications have been made). This specification extends the root element (ROOT) of any document. It defines the Indexed_passage_delimiter as a paired component (PAIR). A mandatory (!) attribute, called Link_to_descriptor, is a reference to an element of type Indexed_passage_descriptor.

Descriptors

An *Indexed_passage_descriptor* describes an index entry. These descriptors are associated elements (ASSOC). Each descriptor contains a term (KEY) and sometimes a subject, a

```
STRUCTURE EXTENSION Index
EXTENS
    ROOT + (Indexed_passage_delimiter);
STRUCT
    Indexed_passage_delimiter (ATTR !Link_to_descriptor = REFERENCE
      (Indexed_passage_descriptor)) = PAIR;
ASSOC
    Indexed_passage_descriptor (ATTR !Link_to_delimiter = REFERENCE
      (Indexed_passage_delimiter)) =
    BEGIN
     Keys (ATTR Level = low,normal,high) = LIST [1..3] OF (Key = Text);
    ? Subject = Text;
    ? Semantics = Text;
    ? Gloss = Text;
    END
```

Figure 4. Generic structure for indexes (delimiters and descriptors)

semantics,[5] and a gloss (see Figure 4). All the descriptors are displayed in a separate view of the document. By double-clicking an Indexed_passage_delimiter, the descriptor view is automatically scrolled to display the linked Indexed_passage_descriptor.

In the same way, a cross-reference may be defined using a *Cross_reference_descriptor*. Unlike the Indexed_passage_descriptor, these descriptors are not linked to a paired component. They are associated elements. Using a command *Open a view*, an additional view can be opened to display the descriptors of cross-references: new descriptors can be added and edited. The Grif indexing system will add cross-references related to these descriptors into the index table when building it.

Building options and index tables

Some building options, such as the title of each table, the list of the relevant subjects, the presentation of the relevant terms, the presentation of the group headings, the sorting order will have an effect on the resulting presentation and content of the index tables. The structure of these options is described in the *Index* DTD. These options can be edited in a separate view.

In order to generate the index within a structured document, the Grif user interface provides a new command within the *Tools* menu. When clicking on this command, the user asks for the generation of one or several indexes. Grif then executes specific procedures which collect all terms, sort them and finally generate one or several structured and formatted index tables with some internal links (cross-references inside an index table), links to the indexed passages, and links to the descriptors.

The resulting index tables are displayed in a separate view, called the *Index* view. Index entries[6] are sorted alphabetically,[7] homographs are shared,[8] terms are sorted, references to

[5] The semantics is used to share homographs.
[6] Grif supports hierarchical indexes: up to three levels of term (item, subitem, subsubitem).
[7] Upper and lower cases can be distinguished. Strings, numbers and special symbols can be differentiated.
[8] A homograph is a word spelt like another, but of different meaning or origin, e.g., BAT.

long passages are represented by a range. The graphical representation of the index tables is generated according to the presentation model. Figure 5 shows a piece of this presentation model using the P language.

```
PRESENTATION Index
VIEWS Main_view
RULES
    Index_title :
        BEGIN
        Width: Enclosing . Width * 80%;
        Size: Enclosing + 6;
        Style: Roman;
        VertPos: Top = NumeroIndex . Bottom + 0.5;
        HorizPos: VMiddle = Enclosing . VMiddle;
        PageBreak: No;
        Line;
        Adjust: VMiddle;
        Justify: No;
        Indent: 0;
        END
    ...
ATTRIBUTES
    Level = high:
        Style: Bold;
    ...
END
```

Figure 5. The presentation model of an index table

The user interacts with the system through this representation. New formatted tables are rebuilt upon user request, for instance, when pagination has changed or when descriptors have been added or deleted. More details about this implementation of active indexes with Grif can be found in [25].

5 FIRST EXPERIMENTS AND ADDITIONAL FEATURES

A first case study has examined the feasibility of our index mechanism on a list of medieval acts. These legal acts are composed of contracts about landed properties or ground rents. Many person names are used to locate the concerned places. Some of them use a nickname, a corporation name, or a birthplace. The spelling often differs. Thus, the volume of indexes is as large as the genuine text [26]. Many cross-references are used. Using Grif, the content of the large index tables can be checked interactively by direct manipulation of links. The structured approach allows moving directly from a selected name in the index table to the relevant act. In this experiment, the index feature of Grif is not used as a tool to browse a finished document[9], but as a means of understanding the document content better and

[9] These medieval acts are included in a book entitled *Un bourgeois parisien du XIII*e *siècle, Geoffroy de Saint-Laurent, 1245?-1290*, Anne Terroine, edited by Lucie Fossier, CNRS éditions.

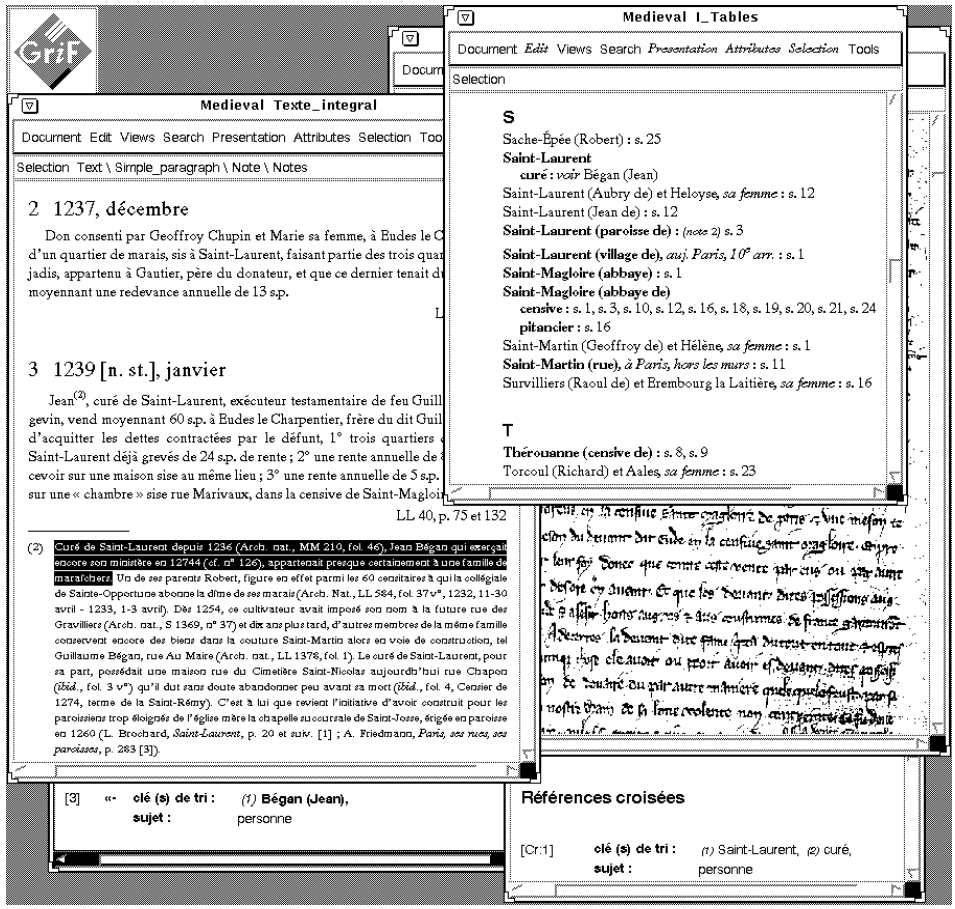

Figure 6. Index table of medieval acts

specially the relations between the various persons and places mentioned in these acts since the original acts are particularly illegible (see Figure 6).

In short, the experiments showed that both authoring and browsing are concerned with active indexes.

- The author can take advantage of the links for navigating from the index to the descriptors and can edit the descriptors immediately, if required. While following the link from the descriptor to the delimiters he can check that the associated passage is pertinent, as well. When the pair of delimiters is suppressed, the corresponding entry in the index table is automatically updated.
- The reader of an electronic index can take advantage of the links for navigating across the document very efficiently and naturally, as the index is displayed on the screen in the same form as in a paper document. So, he or she is not lost when browsing an electronic document from an index table by clicking on the coloured page numbers for locating the relevant passage.

However, the author of the index still has some difficulties in manually identifying good terms for indexing. Thus, among the features considered especially useful is *semi-automatic indexing*: terms determination and recognition of text units such as phrases and paragraphs that deal with the subjects.

As previously mentioned, programs have not yet achieved the level of intelligence required to perform the intelligent process of selecting terms for the index [27,28]. However, the method by which `findphrases` [2] assists in the creation of an index could help the term determination process. Thus all that can be hoped for, in the very near future, is an interactive editor aiding the term determination process and the location of occurrences within a structured document, using information about the context and about the structure of the document [29].

A basic *marking tool* is now being experimented within Grif. This is a very limited treatment of a very complex issue. This command helps the user to find and to select elements depending on their content, their type, or an attribute value. It provides the user with the ability to rely on the structure of the document for locating terms. As an example, the title, the preface, or the bibliography should not be indexed. A list of preselected terms (words or sequences of words) can also be used to search for. Finally the user decides on the insertion of the delimiters around the selected passages. This tool cannot substitute for a real human indexer. It helps to produce the first draft of an index table. The same set of index terms can be used for several related documents. And, when using a full list of preselected terms, only a few entries have to be manually added or suppressed using the Grif editor.

6 CONCLUSIONS

Over the past few years, the approaches of electronic documents and hypertext have often been opposed to each other. However, some hypertext systems use now principles from electronic documents, especially from structured documents. Conversely, many document preparation systems include functionality inspired by the field of hypertext. However, as far as we know, Grif is the only document preparation system that combines these approaches for building indexes within structured documents and provides the same environment for authoring and browsing. The experimental use of the electronic index has shown the effectiveness of the structured approach in the production of an attractive index:

- *Active indexes.* The index table is a context of highly interrelated texts for which it is very useful to follow links and build active indexes. Indexes considered as electronic documents can be easily navigated and browsed by a reader who wants to get information without having to express a query; the page numbers are considered origins of links to structured pieces of text, such as sections or paragraphs, and the cross-references are linked within the index table.
- *Generation of hypertext links.* The structured model of documents helps in designing hypertext links based on the elements and the attributes. It allows a programmed generation of typed links.
- *Structure advantage for managing references.* The references are usually page numbers, but they can also be section, footnote or paragraph numbers, depending on the structure of the document. The structured model of documents helps in calculating, updating, and checking these references.

- *Master indexes.* Building a master index for several documents requires no extension in Grif while combining both external and internal links. Moreover, when a master index has been generated for a set of documents, the quick localization mechanism is very efficient; by double-clicking the origin of a link (an entry in the index table), the editor displays and highlights the target element (the relevant passage within the related document).

Clearly there is an increasing interest in both structured documents and hypertext. The example of index documents above illustrates that the structured approach of the editor Grif provides significant advantages. Areas of further investigation include development of an applicative language in Grif that will allow a user to design more powerful applications dealing with both documents and hypertext.

ACKNOWLEDGEMENTS

I thank colleagues in the OPERA project for their assistance: Jacques André who contributed so much to the early discussions on the design of electronic indexes and to early user feedback, Irène Vatton who helped with testing and integrating, Philippe Louarn for his knowledge of index generation with LaTeX, Éric Picheral and Vincent Quint who provided valuable feedback on early versions of this article, and Anne-Marie Vercoustre and the anonymous referees for their comments on an earlier draft.

REFERENCES

1. University of Chicago Press, *The Chicago Manual of Style*, The University of Chicago Press, Chicago, IL, USA, 14th edition, 1993.
2. K. K. Abe and D. M. Berry, '`indx` and `findphrases`, a system for generating indexes for ditroff documents', *Software—Practice and Experience*, **19**(1), 1–34 (January 1989).
3. Frame Technology Corporation, San Jose, CA, USA, *FrameMaker Reference Manual*, 1991.
4. Microsoft Corporation, Seattle, WA, USA, *Microsoft Word 4 Reference Manual*, 1989.
5. L. Lamport, LaTeX*: A Document Preparation System*, Addison-Wesley, Reading, MA, USA, 1986.
6. R. L. Aurbach, 'Automated index generation for LaTeX', *TUGboat*, **8**(2), 201–209 (1987).
7. J. L. Bentley and B. W. Kernighan, 'Tools for printing indexes', *Electronic Publishing: Origination, Dissemination, and Design*, **1**(1), 3–17 (April 1988).
8. P. Chen and M. H. Harrison, 'Index preparation and processing', *Software—Practice and Experience*, **18**(9), 897–915 (September 1988).
9. L. Lamport, *MakeIndex: an index processor for* LaTeX, February 1987. Document included in the MakeIndex distribution.
10. P. Louarn, 'Traitement d'index avec LaTeX', *Cahiers GUTenberg*, (7), 23–28 (November 1990).
11. R. Furuta, 'Hypertext and electronic publishing', In Rizk *et al.* [12], 347–353.
12. *Hypertext: concepts, systems and applications, ECHT'90*, eds., A. Rizk, N. Streitz, and J. André, Cambridge University Press, November 1990.
13. G. A. Boy, 'Indexing hypertext documents in context', in *Hypertext'91 Proceedings*, pp. 51–61, Association for Computing Machinery (December 1991).
14. E. Fahmy and D. T. Barnard, 'Adding hypertext links to a document archive', *Canadian Journal of Information Science*, **15**(3), 25–41, (1990). Also appeared as Technical Report 89–265, Department of Computing and Information Science, Queen's University, Kingston (Ontario, Canada), 1989.
15. V. Quint and I. Vatton, 'Combining hypertext and structured documents in Grif', in *ECHT '92: Proceedings of the ACM Conference on Hypertext*, eds., D. Lucarella, J. Nanard, M. Nanard, and P. Paolini, pp. 23–32, ACM Press, (1992).

16. *Structured Documents*, eds., J. André, R. Furuta, and V. Quint, Cambridge University Press, 1989.
17. P. D. Stotts and R. Furuta, 'Hypertext 2000: Databases or documents?', *Electronic Publishing: Origination, Dissemination, and Design*, **4**(2), 119–121 (June 1991).
18. A. M. Vercoustre, 'Structured editing—hypertext approach: Cooperation and complementarity', In Furuta [30], pp. 65–78. Proceedings of the International Conference on Electronic Publishing, Document Manipulation, and Typography, Gaithersburg, Maryland, September 1990.
19. ISO, *Text and Office Systems—Standard Generalized Markup Language*, October 1986. Document Number: ISO 8879–1986(E).
20. C. F. Goldfarb, *The SGML Handbook*, Clarendon Press, Oxford, 1990.
21. International Standards Organisation, *Text and Office Systems—Office Document Architecture (ODA) and Interchange Format*, 1989. International Standard 8613.
22. V. Quint and I. Vatton, 'Grif: an interactive system for structured document manipulation', in *Text Processing and Document Manipulation*, ed., J. C. van Vliet, pp. 200–213, Cambridge University Press 1986.
23. R. Furuta, V. Quint, and J. André, 'Interactively editing structured documents', *Electronic Publishing: Origination, Dissemination, and Design*, **1**(1), 19–44 (April 1988).
24. V. Quint, M. Nanard, and J. André, 'Towards document engineering', in Furuta [30], pp. 17–29. Proceedings of the International Conference on Electronic Publishing, Document Manipulation, and Typography, Gaithersburg, Maryland, September 1990.
25. H. Richy, 'Les index électroniques de Grif', Research Report 1756, INRIA (September 1992). In French.
26. J. André and H. Richy, 'Utilisation des index de Grif dans le cadre d'actes médiévaux', Internal report, IRISA, (1994). To appear in French.
27. G. Salton, *Automatic Text Processing*, Addison-Wesley, Reading, MA, USA, 1989.
28. G. Salton, 'A comparison of book indexing methods', Technical Report TR 89–1033, Cornell University, Ithaca, NY, USA (August 1989).
29. D. Kerkouba, 'Indexation automatique et aspects structurels du texte', in *RIAO 85*, pp. 227–249, Grenoble, France (March 1985). In French.
30. *EP90*, ed., R. Furuta, Cambridge University Press, September 1990. Proceedings of the International Conference on Electronic Publishing, Document Manipulation, and Typography, Gaithersburg, Maryland, September 1990.