Darc

Document Archive Controller

Donald Broady
broady@nada.kth.se

Hasse Haitto
haitto@nada.kth.se

Peter Lidbaum
pli@nada.kth.se

Magnus Tobiasson
tobima@nada.kth.se

IPLab/NADA*
Royal Institute of Technology
S-100 44 STOCKHOLM
SWEDEN

IPLab Report 63, TRITA-NA-P9306

March 19,1993

Keywords: SGML, hypertext, information retrieval, CSCW

Abstract

Darc is a multi-user, cross-platform (Sun SPARC/X11 & PC/Windows 3.1) database and information retrieval application designed for storing, reusing, and querying large quantities of SGML[1]-encoded documents. It consists of three major parts:

- *An SGML document on-line delivery tool (i.e. a viewer) with annotation and interactive hypertext facilities.*
- *A set of document databases with access control of users, and network-based support for Computer Supported Cooperative Work (CSCW) through hierarchical groups.*
- *A virtual filing tool to specify hierarchical relationships between documents.*

*Phone: `Int +46 8 790 6000` (Voice) or `Int +46 8 790 0930` (Fax)

[1] Standard Generalized Markup Language, an ISO-standard for document markup.

Background and Funding

Darc (**D**ocument **Ar**chive **C**ontroller) is the principal outcome of the *Computer Supported Knowledge Work* research project, a three-year joint effort between the Interaction and Presentation Laboratory (IPLab) at the department of Numerical Analysis and Computing Science of the Royal Institute of Technology, and the Department of Educational Research, Stockholm Institute of Education (Sweden's largest Teachers' College). The project has been funded by grants from the Swedish National Board for Industrial and Technical Development (NUTEK)[2], the Swedish National Board of Education (Statens Skolverk), and the Swedish Council for Planning and Coordination of Research (FRN).

This report is an overview which emphasizes features of the Darc system. Design rationale and technical aspects will be covered in subsequent papers.

Introduction

There are currently three commonplace techniques for organizing document collections:

- Information Retrieval (IR) methods
- Hierarchical file systems
- Hypertext solutions

The Darc system applies all of these techniques in one coherent framework for document management.

About The Overview

This overview will explain the Darc system from the viewpoint of its users, who belong to one of five categories. These categories form a hierarchy whose levels are numbered 1 to 5 (in ascending order). These levels are described in the next five sections. Some knowledge of SGML is assumed of the reader, but is not absolutely necessary.

1 The Guest

Darc has password-controlled access of database contents, with users belonging to one of five categories. At the lowest level in the Darc system, a *guest* has the least privileges. This category is for the occasional user, as a guest cannot modify anything in the database and is allowed access only to documents available to everyone.

[2] Formerly known by the acronym STU. The project's Swedish title is "Datorstöd för kunskapsarbete," project ITYP 90-02737P.

1.1 Document Databases

Being a database system, Darc is designed to handle vast quantities of documents (literally tens of thousands). Preferably, the documents are marked-up with the ISO standard SGML [7], since it is an SGML-based system and has support for full-text presentation and navigation of such documents. Naturally, the system can also be used as a repository for documents coded in other formats but without the benefits of the SGML support.

A guest can perform traditional, index-based bibliographical searches and view documents on-line. The guest will probably encounter and use *views*. These are covered in the next section.

2 The Reader

Anyone who uses Darc regularly should be granted at least *reader* privileges, as they include the right to create *views*—a mechanism which allows documents to be ordered and accessed in hierarchical structures.

2.1 A Virtual Filing Tool

Views are based on the metaphor of how one organizes the *hierarchical* file structure of a hard disk, but are more flexible. In effect, they allow users to create a personalized interface to database contents, and reduce the need to hunt for documents using traditional index-based searches.

A view (as shown in figure 1) is a set of labeled boxes, or *nodes*. Each node may contain documents as well as other nodes, commonly called subnodes or children. Of course, this representation has nothing to do with the actual physical storage of the documents. Views can be stored and accessed like any other database object.

Furthermore, views are not *static*. Nodes can be cut, copied and pasted. In fact, each user may tailor a view to his or her specific needs, add new nodes, delete obsolete ones, etc. A node can be closed (collapsed) so that it hides its children. The visual aspects of a node are covered in the next two sections.

2.1.1 Labels

A label is the name assigned to a node, and is supplied by the user when the node is created. The label can be any string, but normally reflects the nature of the documents stored within the node, or the methodology used in the hierarchical ordering. Views can be string searched by label.

The user is informed of all accessible views that contain a particular document[3]. This is done when the document is selected, e.g. as the result of a bibliographical search.

2.1.2 Child Indicator

Nodes with subnodes are prefixed with a child indicator. This is a plus sign to the left of the node label when the node is closed, and a minus sign when it is open.

[3] As illustrated in figure 3.

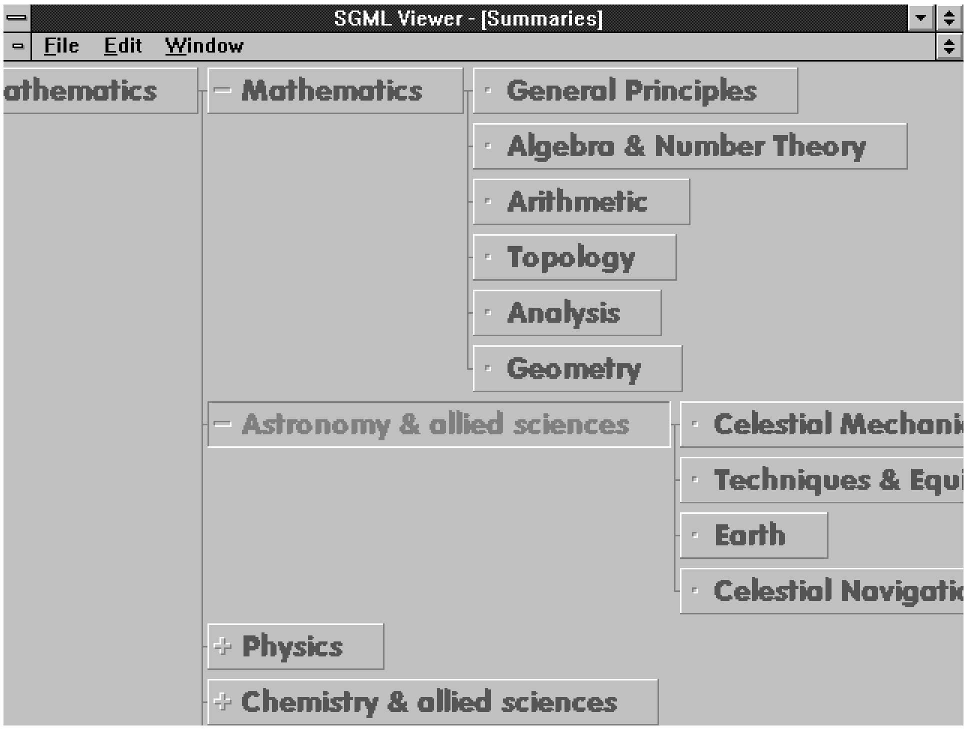

Figure 1: Views allow users to organize documents in hierarchical relations. Unlike the other figures in this paper, this screen snapshot is from the PC/Windows version of Dare. Both versions are functionally equivalent and the document databases are binary compatible across platforms.

Nodes without + or − are leaves, i.e. nodes without children[4]. The same node may exist in several places in a view.

2.1.3 Document Interchange

Views are somewhat similar to symbolic links under UNIX or aliases under Apple System 7, but are more flexible. Finally, the view mechanism is used to *export* a subset of the document database. A selected node, its subnodes and all documents stored therein can be compiled into a new document database. This packaging is a convenient method of database-level document interchange.

2.2 On-line Browsing, Annotating, and Linking

Any user may also read documents on-line. Beginning at the reader level and upwards, the presentation tool supports *browsing*, *annotating*, and *linking* of the documents in the database. These facilities will be covered in detail below.

[4] Leaves are prefixed with a cool 3D bullet. :-)

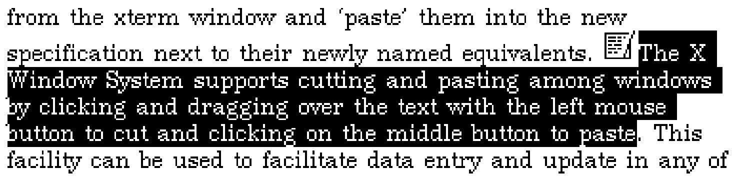

Figure 2: The highlighted section has been annotated. Clicking the icon will open up a separate window to display the contents of the annotation. Like all document objects, annotations can be shared publically with group members or kept private.

3 The Author

Documents may be added to the database (and in some circumstances removed) by users who have *author* privileges or higher.

3.1 Adaptable SGML-based Filing

When dealing with material marked-up with SGML, a system should make use of the existing mark-up to simplify document management. This is done in Darc by adapting database import to various SGML document type definitions. E.g., it is a reasonable assumption that bibliographical data elements will vary for each document class: Therefore, there is an end-user modifiable interface which specifies what elements in a document type definition are to be treated as bibliographical data so that any document will be able to *file itself*[5].

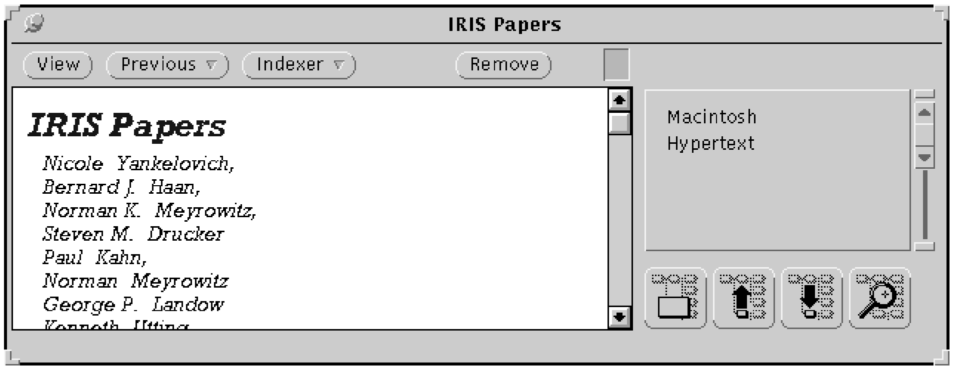

Figure 3: Bibliographical elements are shown in a window of their own. To see the full-text view of a document one selects the `View` *button at the top left. The list at right displays the labels of nodes in which the document is represented in, if any.*

Importing a document marked-up in SGML to the Darc system is extremely simple: First, the document is parsed in a matter of seconds, then the bibliographical data is presented in a separate window (see figure 3). At this point, the user may file the document (i.e. add it to the database), and/or browse through it using the

[5] See section 3.2.3 for how the SGML markup is used for indexing.

on-line viewer. It is thus not necessary to pre-compile or to file a document before reading it on-line.

3.2 An SGML On-line Delivery Tool

The pioneer efforts of Douglas Engelbart's *Augment* at the Stanford Research Institute demonstrated the benefits of organizing files into hierarchical structures, with outline-style access, that can be arbitrarily referenced and linked on-line [4]. The advent of descriptive markup, especially SGML, has since paved the way for on-line viewers to build on this rich heritage of ideas, see e.g. [3, 9]. In the same vein, Darc includes an interactive browser to view SGML-encoded documents and network-based support for CSCW through its notion of groups (see section 4).

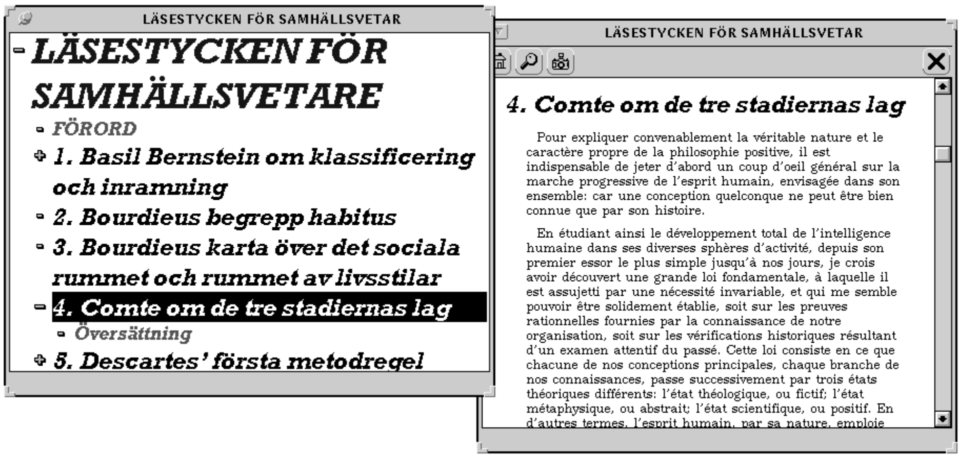

Figure 4: A table of contents and a corresponding full-text view. The user has clicked in the table of contents to rapidly move to the fourth chapter. The items prefixed with a plus-sign in the table of contents can be expanded to show subelements such as sections, subsections, etc.

The Darc on-line delivery tool creates an interactive, outline-style, table of contents from the hierarchical SGML markup that is then used for rapid scrolling of the full-text window contents. Any document element may be defined to appear in the table of contents (or derivatives, e.g., lists of figures). Table of contents entries are made expandable/collapsible (just like nodes in a view) for easy navigation through a document. This is illustrated in figure 4.

3.2.1 Style Sheet Formatting

As SGML separates the format from the content of a document, the screen presentation is governed by (separately stored) style sheets. The style sheets define context-sensitive formatting and are easily redesigned interactively. Differently formatted versions of a document may be open simultaneously.

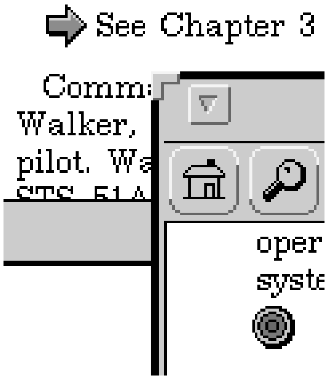

Figure 5: The arrow at the top left is a markup-based cross-reference displayed as a hypertext link. The referred element will have a 'target' icon, such as the one shown at the lower right. Links can be navigated bidirectionally.

3.2.2 Markup-based Linking

The Darc presentation tool resolves, from the SGML mark-up, intra-document cross-references[6] into live hypertext links. The targets of mark-up based links are indicated, so that a referred element is also linked to the elements that refer to it (see figure 5).

The SGML structure is displayed on demand (in a separate window, as shown in figure 6), in what is usually called a *tree*. This provides another way of maneuvering through a text, as well as a way of relating the contents of the full-text window to its hierarchical SGML context.

3.2.3 Markup-based Indexing

Section 3.1 describes how bibliographical data is extracted from the markup. In the same spirit, one can specify what element contents are to be indexed, what the corresponding indexing field should be referred to as, and whether the indexing should be case sensitive. One example would be indexing the contents of, e.g., `<AU>` tags to an index referred to as 'Author', but only when that tag is found in the front matter. Using the full-text index capability of Darc one can look up all documents containing a particular word.

Because the viewer and the database cooperate, one can use the database indices from within the viewer: Any selected word in the text can be looked up *as if it were* an author name, a title word, a keyword, etc. Search parameters are stored so that one can easily redo a search.

3.2.4 String Searching

Naturally, one can also do string searches of document contents. The result of such searches are displayed in a pick list, as illustrated in figure 7. When one clicks in

[6] Encoded with the `ID`, `IDREF` and `IDREFS` attribute mechanism; typically references to elements such as footnotes, tables etc.

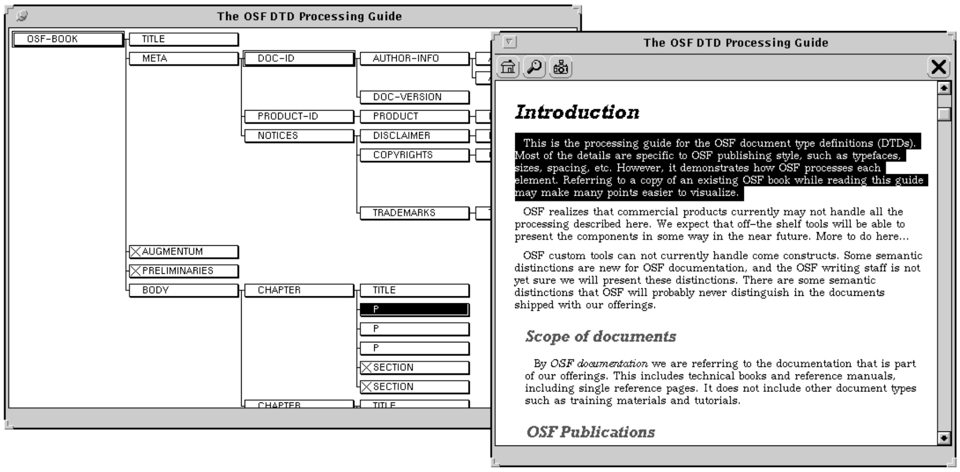

Figure 6: The SGML structure of a document can also be shown as a tree, or hierarchy of nodes, to be used both for navigation and for style sheet access. The marked section in the full text view to the right corresponds to the marked element in the SGML tree. Node elements of the tree can be expanded and collapsed just as the nodes in views (see figure 1).

the pick list, the corresponding line will be scrolled to the top of the full text view. All matching occurrences are shown highlighted. Just as for index-based searches (in the previous section) string search parameters are kept in a list so that one can swiftly redo a search.

3.2.5 Annotation and Layered Hypertext Support

In addition to the links automatically derived from the mark-up, documents can be annotated and linked by hypertext links which are external to the document markup. These user-created links can span documents and are decoupled from restrictions pertaining to SGML element boundaries.

A special form of annotation is *highlighting*, where blocks are highlighted as if by a marker pen.

Webs

Annotations and links are connected to blocks of contiguous selections or *anchors*. This additional data[7] is stored separately in what is usually called *webs*.

Branching Bidirectional Links

A link can fork to a multi-way branch, leading to a choice of several destination anchors. Just as the markup-based links, the dynamically created links can be followed from either direction.

[7] I.e. anchor, link, and annotation information.

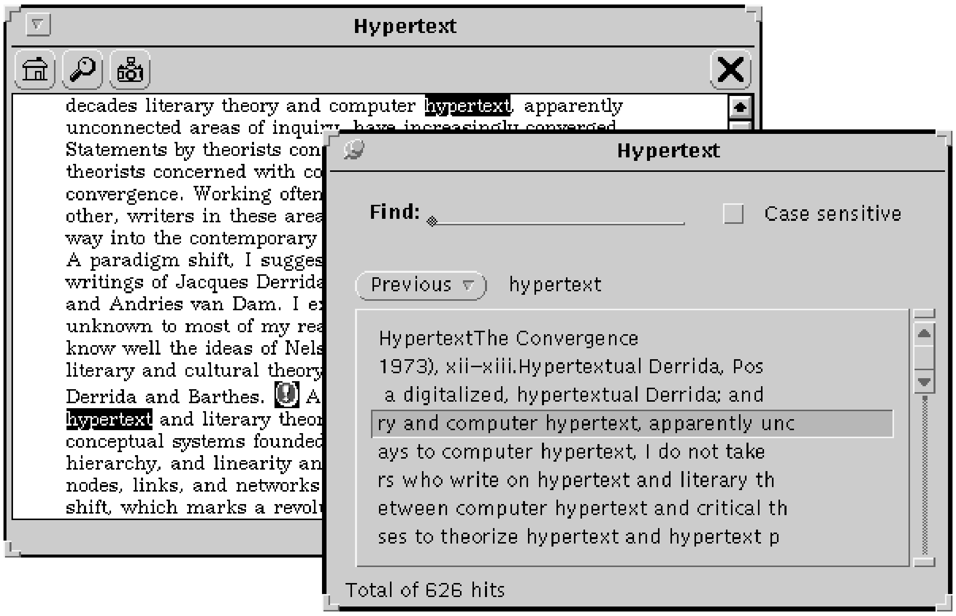

Figure 7: String searches result in a pick list, where the matched string is displayed concordance-style with its surrounding words (here the word 'hypertext' has been sought). The button marked `Previous` *is a list of search strings which have been entered during the session. Notice also the highlighted icon in the full text view; this is a footnote icon, and it has been highlighted because the string being sought occurs in the footnote contents (as can be deduced from the pick list).*

The linking functionality outlined above was pioneered by Brown University's Institute for Research in Information and Scholarship (IRIS) in the *Intermedia* system [6]. We have expanded this functionality in three ways:

- **Unmodal webs**: The externally stored webs can be opened and closed at any time while reading a document. The on-line display will adapt itself accordingly.
- **Concurrent webs**: Several webs, i.e. collections of annotations and/or links, can be open at the same time. A useful metaphor is to think of each open web as a transparency layer upon which the links and annotations are attached; the document is displayed as if it were seen through these layers of transparencies. The anchors to which annotations and links are connected can reside in different webs.
- **Enhanced user feedback**: The user is informed of which documents are contained in a web, as well as the reverse (*"in which webs are there links to this particular document?"*).

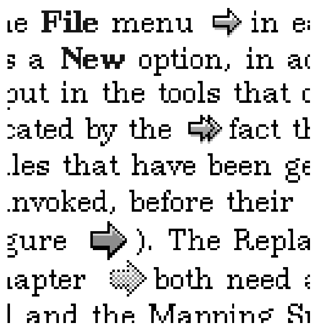

Figure 8: Web-based links are shown with a somewhat smaller icon than their mark-up based counterparts, see the link at the top of the figure. Below, on the fourth line, is a multi-way link (indicated by double arrows).

User Disorientation and The Web Manager

A classical problem in hypertext systems is user disorientation [2]. Solutions to this problem have usually taken the form of maps [5], sometimes in conjunction with a history list or *path* [6]. Brown [1] explores the problem of providing *hierarchical* and *cross-referencing* links (or equivalently, *structural* and *unstructured* links) with respect to the *Guide* system.

In *Guide*, hierarchical links are used to encapsulate document sections that are expanded at will. In contrast, Darc uses the natural document structure inherent in the markup, allowing the user to manipulate this structure by interacting with the table of contents or the graphical SGML tree. The cross-referencing links have three variants:

- **Markup-based cross-references**. The presentation tool resolves intra-document links from the SGML markup as explained in section 3.2.2.
- **Targets of string searches**. Navigation as a result of string searches is also a form of unstructured linking. Darc uses a pick list (see section 3.2.4 and figure 7) as the starting point for such navigation. All links remain available when one follows such a link, and previous search strings are kept during the session.
- **Web-based cross-references**. This is the most powerful (and potentially most disorienting) link mechanism, as it allows unrestricted linking among all available documents. The *Web Manager* is the user interface to these links: All links and annotations[8] are accessible in a list, and any link endpoint anchor contents can be previewed without effectuating the jump.

[8] Optionally filtered, e.g. so that only annotations or links of one's own are displayed.

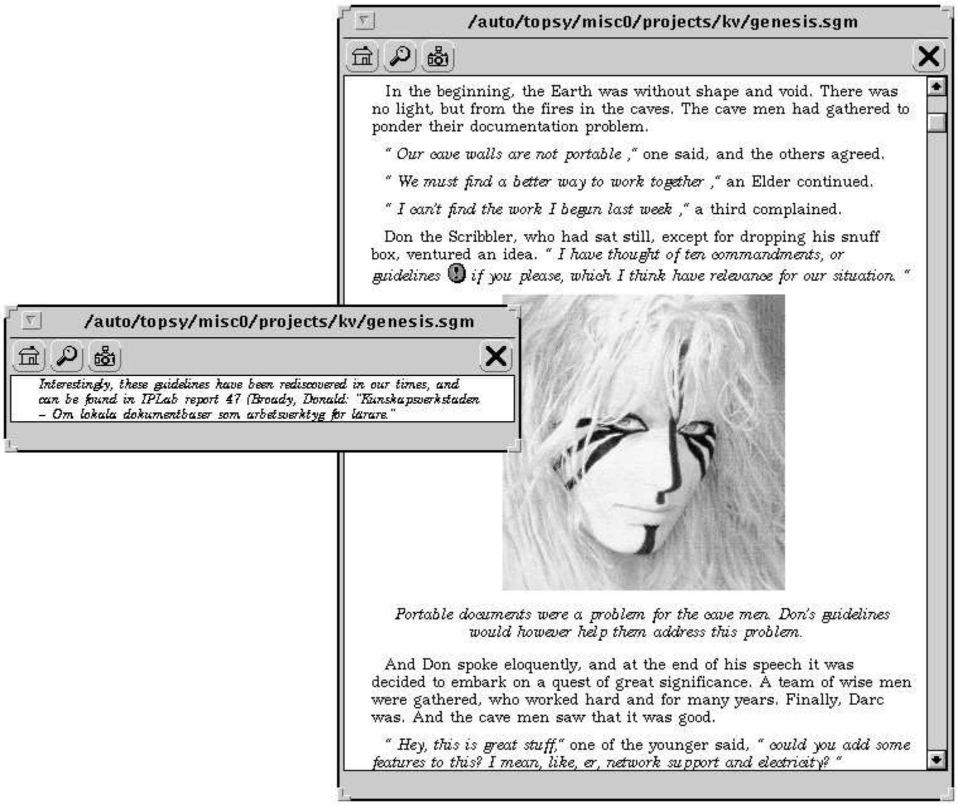

Figure 9: Any document element may be displayed either in-line or in a separate window. The latter display is usually preferred for floating elements such as tables, figures and footnotes. In this figure, a color picture is displayed in-line while a footnote, indicated by an (!) icon (to the left and above in the picture), has its contents displayed in the external top window.

3.3 Inline Display of Elements

Any element content can be shown in-line or displayed—on demand—in separate windows[9]. In the latter case, the elements are indicated with a clickable icon (see figure 9). This behavior is of course tailorable through the mechanism of style sheets.

3.3.1 Graphics Support

The presentation tool has built-in support for common graphic formats, currently Sun raster, `bmp`, `GIF`, `JPEG/JFIF`, `pbm/pgm/ppm`, `pcx`, and `xbm`.

[9]Elements typically presented this way are footnotes and tables.

4 The Group Editor

The Darc database manager is designed for collaborative work groups and provides multi-level access control for all objects across the network. It is the role of the *editor* to create and maintain groups, and to assign accounts and privileges to new users. An editor can remove documents belonging to the group[10] and exclude users that he or she has introduced into the system. There is strict compartmentalization to support document data integrity.

All documents, annotations etc. can either be shared with other groups or kept private. The editor is normally the leader of a work group or project and should provide some *quality control* for documents belonging to the group.

4.1 Group Hierarchies

All users belong to a common group called *all*. Other groups are descendants of this group, which in turn can have subgroups, so that a hierarchy of groups is established: A group has access to all documents belonging to the group or to its ancestors[11].

Groups can also be combined, so that a new group can be created from existing groups. This scheme allows for a gradual refinement of users into work groups, with each group leader assuming the role of editor.

4.2 Object Ownership and Access Attributes

Every Darc object has precisely one owner and belongs to one group. An object's access is controlled by setting two *attributes* to one of three states. Only the owner can modify the attributes, which are *Read Access* and *Modify Access*. The attributes can be set to:

Public A public object is available to all users. A *public read access* allows everyone access to the object. If *modify access* is set to public, anyone can edit the object, e.g. a view or a web[12].

Protected A protected object is restricted to members of the same (or a descendant) group.

Private A private object is restricted to its owner.

An owner can also give away ownership.

5 The System Administrator

The *system administrator* is allowed unrestricted access to all database objects, regardless of their access status, and is the initial user who assigns accounts to editors (and assists them if they forget their passwords).

[10] A privilege shared with the owner of the document.

[11] One could say that the groups inherit previous access rights but deny access from their parent groups.

[12] *Modifiable* with respect to SGML documents implies only that these can be removed from the database. There is not yet an SGML-validating editor built into the system.

Summing Up Key Concepts

The Darc system combines methodologies from the fields of information retrieval, hypertext, and CSCW, to create a coherent framework for SGML-based document management.

- SGML is used to maximum advantage to simplify document processing in e.g. formatting, filing and indexing. Documents in SGML can be examined directly without preprocessing.

- The information structure conveyed by the somewhat static hierarchical ordering is complemented by the powerful *layered web* mechanism of the presentation tool. Advanced hypertext facilities allow any continuous selection to be made into an anchor, to be highlighted, annotated, or linked to.

- Network-based *group support* is built into the system, distributing the responsibility of user maintenance without sacrificing security considerations or data integrity. Any database object such as a document, view, style sheet, or web, can be shared.

- *Customization* is encouraged in every aspect such as style sheet design, creation of personal views, live hypertext linking, annotation, and highlighting.

- The system emphasizes the use of *hierarchical organization*, as it is an ordering concept which is powerful yet familiar. It is employed in creating the hierarchical categories of users (the five levels), and again, in the user group construct. It is also used in views to organize documents. Looking at the document element level (corresponding to the SGML markup), hierarchical structure is inherent in the table of contents, and (of course), in the graphical tree representation of SGML-encoded documents.

Acknowledgements

The authors thank Rand Waltzman for perceptive criticism of this report.

Copyright Information

References

[1] Brown, Peter: *Linking and Searching Within Hypertext*, Electronic Publishing, Vol. 1(1), April 1988.

[2] Conklin, Jeff: *Hypertext: An Introduction and Survey*, IEEE Computer, Vol. 20, No. 9, September 1987.

[3] Electronic Book Technologies: *DynaText System, Reader Guide*, Providence, RI, 1991.

[4] Engelbart, Douglas C. & English, William K.: *A research center for augmenting human intellect*, Proceedings of the 1968 Fall Joint Computer Conference, Montvale, N.J., 1968.

[5] Envos Corporation, *NoteCards User's Guide*, Release 1.1M, Mountain View, CA, 1989.

[6] Institute for Research in Information and Scholarship, Brown University, *User's Guide: IRIS Intermedia*, Release 3.0, Providence, RI, 1990.

[7] International Organization for Standardization: *Information Processing—Text and Office Systems—Standard Generalized Markup Language (SGML)*, ISO 8879-1986 (E), 1986.

[8] Pirinen, Joakim: *Socker-Conny*, Tago Förlag, 1985.

[9] Raymond, Darrell R.: *Flexible Text Display with Lector*, IEEE Computer, August 1992.

Contents