Donald Broady, Hasse Haitto, Peter Lidbaum & Magnus Tobiasson,
*Navigational Issues and Multi-User Layered Hypertext in the Darc System*,
Institutionen för numerisk analys och datalogi, KTH, Stockholm,
29 March 1993, 10 p.

Navigational Issues
and
Multi-User Layered Hypertext
in the (Darc) System

Donald Broady
broady@kada.kth.se

Hasse Haitto
haitto@kada.kth.se

Peter Lidbaum
plidbaum@kth.se

Magnus Tobiasson
tobiasson@kth.se

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

March 29, 1993

Keywords: Darc, hypertext, database, concurrency, web.

Abstract

Darc is a multi-user, multi-platform (PC, Unix, Sparc) database and information retrieval application designed primarily for SIMO research documents. Among the features is a full-text document browser, in which markup-based hypertext linking is complemented by on-line indexing and annotation facilities through concurrent editors. Cooperative work is supported through a new hierarchical user group notation.

*PLab (Platform Laboratory) at KTH

---

FAKSIMIL

Donald Broady, Hasse Haitto, Peter Lidbaum & Magnus Tobiasson,
*Navigational Issues and Multi-User Layered Hypertext in the Darc System*,
Institutionen för numerisk analys och datalogi, KTH, Stockholm,
29 March 1993, 10 p.
Navigational Issues
and
Multi-User Layered Hypertext
in the Darc System

Donald Broady
broady@nada.kth.se

Peter Lidbaum
pli@nada.kth.se

Hasse Haitto
haitto@nada.kth.se

Magnus Tobiasson
tobima@nada.kth.se

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

March 29, 1993

Keywords: SGML, hypertext, databases, concurrency, webs.

Abstract

DARC is a multi-user, cross-platform (PC/Windows 3.1 & Sun SPARC/X11) database and information retrieval application designed primarily for SGML\(^1\)-encoded documents. Among its features is a full-text document browser, in which markup-based hypertext linking is complemented by on-line linking and annotation facilities through concurrent webs. Cooperative work is supported through a new hierarchical user group mechanism.

\(^{*}\)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.
1 A User Scenario

Consider the following scenario: A user is querying a document database at a workstation, and finds a set of documents matching some criteria, e.g. keywords. The user begins perusing the document on-line, follows cross-references displayed as hypertext links, and perhaps makes a few electronic annotations to the documents. Every now and then, the user finds some interesting connection between documents and inserts links between corresponding passages. Satisfied of these findings, he or she then organizes the documents hierarchically in folder-like fashion for fast retrieval next time around. Finally, the user selects the newly created 'folder' for export, perhaps to continue work on the PC at home, or to publish it electronically e.g. as courseware.

This is a typical example of work with IDim. It is a multi-user document database system designed primarily for SGML-encoded documents that runs on the PC/Windows 3.1 and Sun SPARC/X11 platforms.

2 Introduction

IDim uses three techniques for organizing document collections:

- Information Retrieval (IR) methods to access large collections of documents, where e.g. full-text and structure-sensitive indexing is performed by user-tailorable criterias on the SGML markup.
- A virtual, hierarchical file system to access database contents in much the same way as window-based file managers. This interface is personal, so that a user can construct any number of such views of the database.
- Hypertext services. Links are either based on the SGML markup or interactively created by the user.

Also, IDim has access control of database objects and a novel, hierarchical user group concept.²

²Broady 1993] is an overview of the system. A preliminary version was shown at ECHT’92 [Lucarella 1992].

3 Navigational Issues

Douglas Engelbart’s pioneering work with Augment convincingly demonstrated the benefits of organizing files into hierarchical structures, with outline-style access, that can be arbitrarily referenced and linked on-line in a collaborative work group environment [Engelbart 1968]. The use of descriptive markup, especially the international standard SGML [SGML 1986], has since paved the way for on-line viewers to build on this rich heritage of ideas, see e.g. [EBT 1991, Raymond 1992]. In the same vein, IDim has an interactive browser to view SGML-encoded documents and network-based support for cooperative work through its notion of groups (see section 6). Within IDim, SGML-encoded documents can be navigated, annotated, and arbitrarily linked with any other document in its database.

3.1 Structured Documents

SGML documents are by nature structured, and can be represented as trees (reflecting the structure of the hierarchical markup). The IDim on-line delivery tool creates an interactive, outline-style, table of contents from the SGML markup. The table of contents is then used for rapid access to the corresponding full-text window contents. A feature of IDim is that documents can be browsed directly without a pre-compilation step. The SGML tree structure3 is displayed on demand as an alternative way of accessing the contents. Both mechanisms are illustrated in figure 1.

Brown explores the problem of providing hierarchical and cross-referencing links (or equivalently, structural and unstructured links) with respect to the Guide system [Brown 1988].

In Guide, hierarchical links are used to encapsulate document sections that are expanded at will. In contrast, IDim uses the 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 are covered in the next three sections.

³Much as the Tree Mode of WE [Smith 1987].
3.3 Navigation by Search

Navigation as a result of string searches is also a form of unstructured linking. \( \text{Dans} \) uses a pick list as the starting point for such navigation. The result of string searches are displayed in a pick list, as illustrated in figure 3. When one clicks in the pick list, the full-text view will scroll the corresponding line to the top of the browser window. All matching occurrences are shown highlighted. The pick list remains available when one follows such a link, and previous search strings are kept during the session.

3.4 Web-based Linking

In addition to the links automatically derived from the mark-up, or created while searching, documents can be linked by hypertext links, and annotated, in \( \text{webs} \) stored externally. This is the most powerful (and potentially most disorienting) of \( \text{Dans} \)'s link mechanisms, as it allows un-
decades literary theory and computer *hypertext*, apparently
unconnected areas of inquiry—have increasingly converged.

Statements by theorists concerned with convergence. Working often
with each other, writers in these areas have moved into the contemporary
A paradigm shift, I suggest, is taking place in the writing of Jacques
Derrida and Andries van Dam. In the case of Derrida, I am
unsurprised to find the term "hyper," which has been
sought. The button marked Previous is a list of
search strings that have been entered during the session. Notice
also the highlighted icon in the full text
browser window; 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).

restricted linking among all available documents.
Just as links derived from markup, the dynamically
created links can be followed from either
direction.

Such linking functionality was pioneered by
Brown University’s Institute for Research in Infor-
mation and Scholarship (IRIS) in the *Interme-
dia* system [IRIS 1990]. We have expanded this
functionality in several ways, the most important
addition being the capability of having multiple
webs open simultaneously, achieving *concurrent, layered hypertext* support:

- **Concurrent webs:** Several webs, i.e. col-
llections 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 trans-
parencies. The anchors to which annotations
and links are connected can reside in differ-
ent webs.

The mechanism of concurrent webs can be
used e.g. for gradual disclosure of course
materials. An example would be different
educational uses of one and same document
base: The freshmen are offered a web cover-
ing a small number of documents and a
restricted set of links, more advanced stu-
dents use a more extensive document collec-
tion and a wider variety of links, while the
teacher develops even more complex webs for
personal use.

- **Unmodal webs:** The externally stored
webs can be opened or closed at any time
while reading a document. The on-line dis-
play will adapt itself accordingly.

- **Enhanced user feedback:** The user is in-
formed of which documents are contained in
3.6 Document-level Navigation

Figure 2: 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.

Figure 4: At the top of the figure is an annotation connected to an anchor. Below is a highlighted passage. When one clicks on an annotation icon, the corresponding anchor will be displayed as shown here. Database objects such as webs—used for annotations and links—can be shared with other users, group members or kept private.

3.5 Anchors

As in Intermedia, Datta anchors are blocks of contiguous selections. Anchors are central to web-related data and are used in three contexts:

- **Highlighting.** An anchor can be highlighted so that the corresponding text is shown as if it were highlighted by a marker pen. (Figure 4).
- **Annotations.** Annotations are connected to an anchor, so that it is always clear what is annotated. (Figure 4).
- **Linking.** Finally, anchors are the endpoints of web-based links. Several links can point to the same anchor. (Figure 5).

3.5.1 Link Anchor Previewing

The Web Manager is the user interface to the web-based links: All links are accessible in a list, and any link endpoint anchor contents can be previewed without effectuating the jump. As well as annotations, optionally filtered e.g., so that only annotations or links of one’s own are displayed.

3.5.2 Taxonomic Links

Finally, a link can fork to a multi-way branch, leading to a choice of several destination anchors, i.e. Datta supports what DeRose calls taxonomic links [DeRose 1989].

3.6 Document-level Navigation

Datta files documents according to user-tailorable criterias on the SGML markup, with indexing contextually dependent for any element. A nice feature is that the document browser is aware of the database indices of any given document, so that a selection can be looked up in any of the indices. E.g., one can select a name from a bibliographic reference in the full-text browser, and immediately look it up in an index of author names. In this manner, one can reach other database documents by following implicit or associative links.
3.7 Multi-Document Hypertexts

Gluschko characterizes four classes of links to consider in the design of multi-document hypertexts [Glushko 1989]. We notice that Baro has support for all four types.

3.7.1 Explicit Intra-Document Links

This kind of link connects two (or more, in our case) parts of a document together. Such linking can be performed by either inserting appropriate SGML markup, or by creating web-based links online.

3.7.2 Implicit Intra-Document Links

This class of link is implied by the foregoing, e.g. every "See also"-type of cross-reference should have a matching "Cited by" link. Baro inserts a 'target' icon for such links, as mentioned in section 3.2.

3.7.3 Explicit Inter-Document Links

Glushko points out that explicit inter-document links "pose more challenges for the hypertext designer...because it is harder to predict the usefulness of the information at the end of the link" (a work may be cited for many different reasons).

3.7.4 Implicit Inter-Document Links

These kind of links become apparent from "careful and creative analysis of...the texts". The possibility of accessing indices from within the browser (section 3.6) gives some support for such activities, but it is perhaps, as Glushko concludes, better "to provide functions that make it easy for readers to create private links and notes."

4 Views

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 6) is a set of labeled boxes, or nodes. Each node may contain documents as well as other nodes, commonly called subnodes or children. One can construct any number of views, that can be shared with other users or kept private.

5 User Disorientation

A reader of a hyperdocument encounters two classical problems leading to user disorientation [Conklin 1987]. The first of these is the problem of navigation ("lost in hyperspace"), usually addressed by solutions in the form of maps [Envos 1989], sometimes in conjunction with a history list or path [IRIS 1990, Utting 1989].

In Baro, user disorientation is countered by consistently bidirectional links which permit easy backtracking, the use of unmodal lists (see e.g. section 3.3) in connection with anchor previewing and, finally, conveying a sense of the underlying
Figure 6: 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 \texttt{\textcopyright}. Both versions are functionally equivalent and the document databases are binary compatible across platforms.

document structure through the interactive table of contents and SGML tree representation.

The second problem is of cognitive nature: For comprehension, the reader must create a coherent mental representation of a hypertext. It is thus essential to grasp the boundaries of a hypertext corpus [Utting 1989].

This problem is tackled in several ways: By informative feedback which allows the user to determine what documents are linked (or annotated) in any particular web and, also, to see if a document is connected to some web. The selective filtering options available from within the Web Manager—as mentioned in section 3.5.1)—refine this process: a user can choose to work only with material he or she has created, or material created by a particular group.

Finally, \textit{views} make it possible to encapsulate sets of documents in hierarchical relations. One can e.g. collect all documents pertaining to a web and place them in one or more views to create a corpus. Thus, \texttt{\textcopyright} has a document-level knowledge structuring mechanism.

6 The User Environment

The \texttt{\textcopyright} database manager is designed for collaborative work groups and provides access control for all database objects across the network. Every object belongs to one owner and is attached to one group. What is novel is the way groups are formed. To begin with, all users belong to a common group. Other groups are descendants of this group, which in turn can have subgroups, so that a hierarchy of groups is established. Group administration is performed from within the system and is not related to e.g. UNIX groups. \texttt{\textcopyright} supports multiple inheritance, meaning that a group can be formed through the
A group has access to all documents belonging to the group or to its ancestors\textsuperscript{7}.

### 6.1 User Privileges

Hierarchical ordering is used in many contexts of the system. It appears once again in the privileges users are accorded within their group. We have, in ascending order of privileges, the following user roles:

- **Guest.** A guest can only access objects (views, webs, documents, etc.) accessible to every other user.
- **Reader.** A reader is allowed to create views and webs.
- **Author.** An author can also add new documents to the database.
- **Editor.** An editor is granted all of the above, and can also create new groups and admit new users\textsuperscript{8}.

The privileges map well to the ways work groups tend to function, while the group construct models organizational structures.

### 7 Conclusions

The Dim system is an attempt to manage several current topics in the field of hypertext:

- **Hypertext and structured documents** [Quint 1992].
- **Hypertext and document databases** [Stotts 1991].
- **Collaborative knowledge-structuring tools** [Streitz 1992, Marshall 1991].
- **Integration of hypertext systems with SGML** [Quint 1992, Streitz 1992].

\textsuperscript{7}The groups inherit previous access rights but deny access from their parent groups.

\textsuperscript{8}There is also a system administrator, who is the initial user who assigns accounts to the editors.
References


