The Data Model is the Heart of Interface Design

Robert Akscyn, Elise Yoder, and Donald McCracken

Knowledge Systems Incorporated
RD2 213A Evans Road
Export, PA 15632

Reprinted from *Proceedings of CHI '88*, ACM SIGCHI, May 1988, Washington, D.C.

Abstract: For the past six years, we have been developing a commercial hypermedia system (KMS) based on our previous research with the ZOG system at Carnegie Mellon University. Our experience with ZOG and KMS has convinced us that the data model underlying an interactive system is more important than the user interface in shaping the overall system. In this paper, we show how the KMS data model has influenced important aspects of the user interface. In particular, we show how the properties of KMS frames--their spatial nature, breadth-first view, homogeneity, small size, etc.--affect the nature of the KMS user interface.

Keywords: Conceptual Data Model, User Interface, Hypertext, Hypermedia

---

Introduction

We have been developing hypermedia systems for over a decade. This experience has taught us many lessons about the design of interactive software. If there is one central lesson from our experience, it is the fundamental importance of the data model underlying an interactive software system. We believe this because we have seen the formative influence of the data model on all other aspects of ZOG and KMS.

By the term "data model" we primarily mean what is sometimes termed the "conceptual data model." (For the purposes of this paper, we will consider the data model to be distinct from the user interface.) We would also like the term "data model" to include the underlying data model used to implement the system, although only to the extent it affects user interaction. Finally, we include those aspects of the data model that have evolved by convention, even though they're not enforced by the system.

In this paper, we will argue that a system's data model profoundly influences the user interface and therefore should receive considerable attention. We will use KMS as our example system to show concrete examples of the data model's influence.

Section 2 provides some historical background about ZOG and KMS. Section 3 describes the basic components of the KMS data model. Section 4 describes some of the ways in which the KMS data model influences the KMS user interface.

Background

In 1975 at Carnegie Mellon University, Allen Newell, George Robertson and Don McCracken started the ZOG project to study the general characteristics of large, rapid-response, menu- selection systems [Robe81, McCr84]. In 1980, we worked with the Navy to build a computer-assisted management system for the Navy's newest nuclear- powered aircraft carrier, the USS CARL VINSON. By 1983, the VINSON had a distributed ZOG system running on a network of 28 PERQ workstations [Aksc84b]. In addition, a commercial version of ZOG, called KMS (Knowledge Management System), was first released in 1983 [Aksc88a].

Hypermedia is a generalization of the hypertext concept, which is over 40 years old. In 1945, Vannevar Bush envisioned a machine that would support browsing in a large on-line text and graphics system [Bush45]. Douglas Engelbart developed one of the first computer-based hypermedia systems (NLS) in the 1960's [Enge63]. The terms "hypertext" and "hypermedia" were coined by Ted Nelson [Nels80], whose Xanadu system aimed to provide easy access to large amounts of literature. In addition to the work at CMU, another long-term study of hypermedia has been the work at Brown University [Yank85]. Currently, there are hypermedia research projects at many sites, including major programs at Xerox [Trig86b], Tektronix [Deli86b], and MCC [Gull86]. Conklin [Conk87d] provides a good review of the hypertext/ hypermedia concept. In addition, a recent conference on hypertext systems (Nov. 1987) highlights relevant issues such as supporting collaborative work [HT87].

Over the years we have been zealous users of hypermedia. We have used ZOG and KMS for almost every aspect of our work, including issue analysis, document production, software engineering, communication, and administration. We estimate that we have logged over 10,000 person-hours as hypermedia users, and have created on the order of 50,000 frames (hypermedia nodes).

The KMS Data Model

KMS is a distributed hypermedia system for networks of workstations. It is designed to support organization-wide collaboration for a broad range of applications, such as electronic publishing, software engineering, project management, computer-aided design and on-line documentation.

A KMS database consists of a set of interlinked, screen-sized workspaces called "frames". There may be as many frames as disk space permits, distributed across any number of servers on a network. Frames may contain any mixture and arrangement of text, graphics and image items, each of which may be linked to another frame or used to invoke a program. Over the years, strong conventions have evolved for the format of frames; see the example in Figure 1.

A KMS database may have any structure the users desire, including an interwoven, "bowl of spaghetti" network structure. In practice, however, most databases have a multi-level hierarchy of frames that serves as the "backbone." Users supplement the hierarchy freely with cross-reference links, and links to supporting information such as comments.

Figure 2, on the following page, illustrates a small fragment of a KMS database. In this example we show part of the hierarchy of frames used to represent this paper.

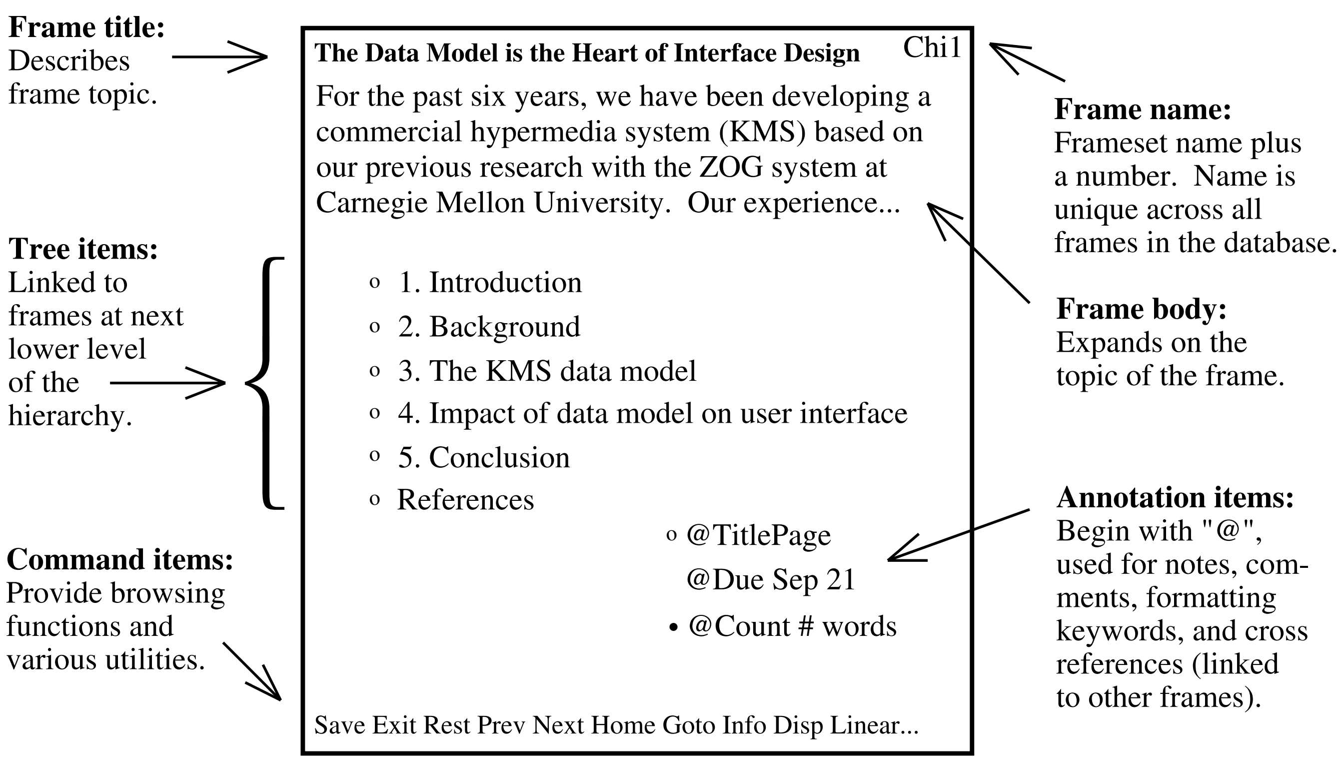

Figure 1. Conventional KMS frame format

A frame may contain any arrangement of text, graphics, and image items. Each individual item can be linked to any other frame (indicated by a small hollow circle) or to an action that invokes a program (small solid circle).

Figure 2. Fragment of a hierarchy of frames

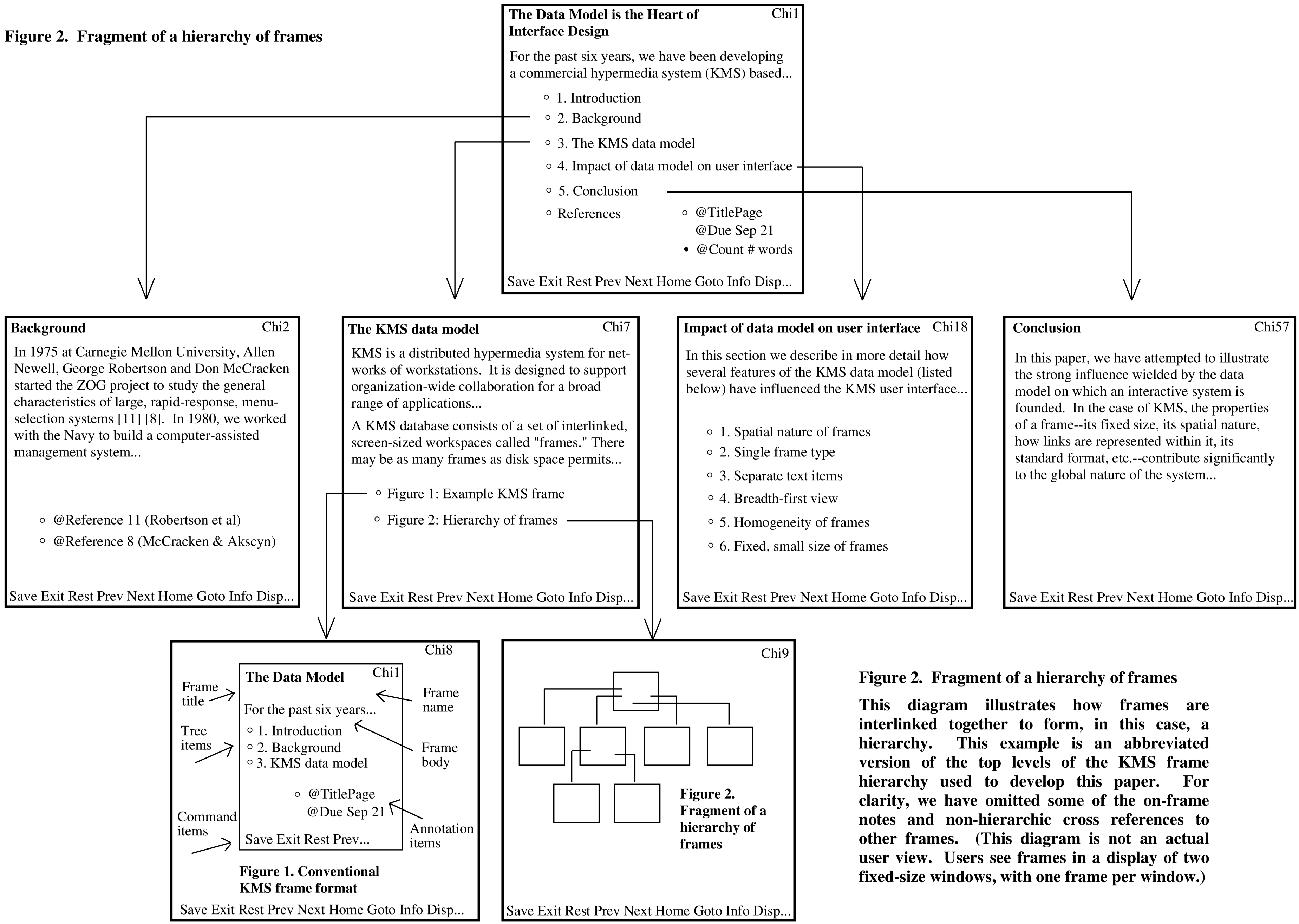

Figure 2. Fragment of a hierarchy of frames

This diagram illustrates how frames are interlinked together to form, in this case, a hierarchy. This example is an abbreviated version of the top levels of the KMS frame hierarchy used to develop this paper. For clarity, we have omitted some of the on-frame notes and non-hierarchic cross references to other frames. (This diagram is not an actual user view. Users see frames in a display of two fixed-size windows, with one frame per window.)

The impact of the KMS Data Model on the KMS User Interface

In this section we describe in more detail how several features of the KMS data model (listed below) have influenced the KMS user interface. Although the data model makes its presence felt in many areas of the system, we have chosen those aspects that have had the most pervasive effect on the interface.

Summary of data model features and their impact

1. Spatial nature of frames
- Reduces hidden state problems
- Reduces tedious repositioning for two-dimensional structures

2. Single frame type
- **Reduces the number of concepts**
- **Reduces the number of modes**

3. Separate text items
- Avoids highlighting for links
- Provides natural default operand scope

4. Breadth-first view provided by a frame
- **Makes it easy to represent hierarchical structures**
- **Reduces the need for a graphical browser**
- **Makes it easy to edit structure**

5. Homogeneity of frames
- Provides structural regularity

6. Fixed, small size of frames
- Eliminates scrolling
- Decreases system response time
- Facilitates simultaneous access (by multiple users)

KMS data model features and their implications

1 Spatial nature of frames

A fundamental aspect of KMS is the spatial nature of frames. The boundaries of a frame define a two-dimensional "canvas" upon which the user can freely place text, graphics and image items. Here are some implications of this fundamental feature:

Reduces hidden state problems. Because space is represented in a natural way, separate from the items placed in it, users do not have to infer how space is represented. By contrast, the typical text editor represents space in a degenerate way. Space to the left of items is usually represented by a combination of spaces, tabs, and carriage returns. Space to the right of items may be explicitly represented in a similar manner, or it may not be represented at all.

Reduces tedious repositioning for two dimensional *structures.* Representing the space surrounding a piece of text as part of the text has the side effect of tightly coupling the positions of items. Changes to the space between two visible text components will have a ripple effect on the positions of other items. As a result, editing two-dimensional structures such as tables can be very tedious. However, there are situations where the user does wish the items within a frame to be coupled, such as a simple list of paragraphs. These are more difficult to edit in KMS, but we believe the tradeoff is in our favor.

2 Single frame type

KMS has only one frame type. Variety is provided at the level of individual items within a frame--the text, graphics and images. This single frame type has the following implications:

Reduces the number of concepts. Frames unify into a single construct what are often distinct levels in traditional computing environments. For example, the functions of a "desktop", directories, files, clipboards, menu bars and pull-down/pop-up menus are all provided by frames. This unification simplifies the user’s conceptual model considerably, and eliminates the need for many commands (e.g., commands for manipulating directories).

Reduces the number of modes. Having a single frame type encouraged us to develop a single editor that could operate on all types of objects in a frame. Furthermore, we merged navigation with editing, providing a single major command context in which all commonly used commands are directly available to the user. Having a single frame type and single direct-manipulation mode greatly simplifies the user’s model of the system. (In fact, we no longer describe KMS as "having an editor.")

3 Separate text items

KMS uses an individual text item as the anchor for a link to another frame. (Many hypermedia systems use a substring embedded within a larger text item.) Users break text into smaller chunks rather than having one large block of text on a frame. This has the following effects on the user interface:

Avoids highlighting for links. The extent of each anchor is readily apparent, since the corresponding text item is surrounded by white space. No text highlighting is necessary to "set off" the anchor. KMS need only denote whether an item has a link (i.e., whether it is serving as an anchor). KMS does this by displaying a small circle icon to the left of each linked item.

Provides natural default operand scope. Representing text as separate items provides a natural default operand scope for common operations such as moving, copying and deleting. Over ninety percent of the commands invoked in KMS (by frequency) are invoked on the default scope, i.e., a whole item. This separation of text, combined with context sensitive mouse buttons using a three button mouse, reduces the average time per operation to less than half of what it is with typical menu-selection interfaces.

4 Breadth-first view provided by a frame

Each KMS frame provides the user with a breadth- first overview of the frames in the hierarchy below. A user can simply read the frame’s tree items, which link to lower-level frames in the hierarchy, to get a sense of what lies below. Here are some of the effects we attribute to the breadth-first aspect of frames:

Makes it easy to represent hierarchical structures. Although a KMS frame may be linked to any frame, KMS databases generally have a strong hierarchical flavor, which provides organizational discipline. Frames are easily linked together to form hierarchies, since the breadth-first structure of a canonical frame makes it an ideal building block for this purpose. Higher-level frames serve as indexes, and lower level frames represent the hierarchical structure internal to things such as documents and programs. Using hierarchical frame structures to represent a document eliminates the need for embedding structure-defining commands in the text.

KMS has two types of links: hierarchical and non-hierarchical. Non-hierarchical links are prefaced with the character "@", which users (and programs) can easily recognize and easily change. We believe this distinction is absolutely critical to the usability of large-scale hypermedia databases. Knowing which type of link they are selecting helps users remain oriented in the database. It also helps users build a coherent mental model of its structure.

Reduces the need for a graphical browser. In addition to the breadth-first view, KMS provides only one other view of the contents of the database--a linear view of a hierarchy of frames. This view permits users to create well-formatted documents from material in KMS and see the documents on-line. We do not provide a graphical browser. The breadth-first view, combined with rapid system response, seems to serve well for navigation within the database. Graphical views might be useful, but we’re not convinced they’re worth the additional complexity.

Makes it easy to edit structure. Users make many more structural changes to their KMS-based documents than they do to documents based in text files. We ascribe this behavior to the structural clarity of the breadth-first view and the ease of making structural changes. For instance, users can more readily perceive that the subsections of a given section are incomplete or out of order.

Users can easily restructure documents. For example, moving a section to another part of the document hierarchy or changing the order of chapters is as simple as moving a single item.

5 Homogeneity of frames

Items in frames may be placed in any arrangement desired by the user. However, over the years, a conventional format for frames has evolved that has remained remarkably stable (see Figure 1). We liken this format to the chemical element carbon, since it combines so well to form many useful structures. Use of this conventional format has the following effect:

Provides structural regularity. Having a conventional format gives frames a visual regularity that makes it easier for users to perceive the components of the frame, interpret them, and make a decision about what to do next.

6 Fixed, small size of frames

Frames in KMS have a fixed size. Frames are nearly as large as conventional workstation screens (a small amount is used for a message window). This size can accomodate a complex diagram in landscape orientation or two pages of a document side-by-side. Having a fixed size for frames that is much smaller than the typical size of a document file yields some important benefits:

Eliminates scrolling. Perhaps the most important implication of fixed-size frames is the elimination of scrolling operations in KMS. We consider scrolling to be a very inefficient method for moving around in a knowledge structure; KMS provides fast navigation in a hieararchy as an alternative to scrolling. We also believe that having scrolling available encourages people to build less useful linear structures rather than hierarchical ones. The absence of scrolling also makes the user interface simpler.

Decreases system response time. Since KMS operates by accessing a frame only when it is to be displayed (or processed by a program), the amount of data retrieved from disk at each request is much smaller than the average text file representing an entire document. On average, frames are 1 kilobyte in size. This small size, combined with the incremental access strategy, greatly speeds up the initial access of data--a factor that becomes even more important as the database increases in size and is more widely distributed.

In addition, a small frame size makes caching schemes very practical. In KMS, accessed frames are cached in their internal record structure, including bit-map images of the text items. As a result, redisplay of a previously accessed frame is very fast, averaging less than 1/4 second.

Facilitates simultaneous access. Because knowledge in KMS is divided into frame-sized units, multiple users can work simultaneously on the same structure, such as a proposal. Since each frame generally contains only one or two paragraphs, the number of frames for even a small document soon outnumbers the authors, giving them plenty of elbow room for simultaneous authoring. We have found that an optimistic concurrency control mechanism is adequate, when combined with fallback methods for handling collisions, and user conventions for asking others to "stay clear" [Aksc88a].

Conclusion

In this paper, we have attempted to illustrate how the data model on which an interactive system is founded can have a strong influence on the design of the user interface. In the case of KMS, the properties of a frame--its fixed size, its spatial nature, how links are represented within it, its standard format, etc.--contribute significantly to the global nature of the system.

We recommend that interactive systems be developed from the inside out--from the data model to the user interface--rather than the other way around. This view contrasts sharply with the philosophy that the user interface should be the dominant system component and thus standardized across programs. We would prefer to see greater standardization at the data model level, with more freedom for variation of the user interface.

References

- [Aksc84b] Akscyn, R. M., McCracken, D.L.: ZOG and the USS CARL VINSON: Lessons in System Development, Carnegie-Mellon Technical Report CMU-CS-84-127. In Proceedings of the First IFIP Conference on Human-Computer Interaction; Interact '84; London, England. 1984.
- [Aksc88a] Akscyn, R. M., McCracken, D. L., Yoder, E. A.: KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations. Communications of the ACM, 31:7, 820-835. 1988.
- [Bush45] Bush, V.: As We May Think. The Atlantic Monthly, July 1945: 101-108. Reprinted in Adele Goldberg (editor), A History of Personal Workstations, ACM Press, New York, 237-247. 1988.
- [Conk87d] Conklin, J.: A Survey of Hypertext. IEEE Computer 20, 9, 17-41. September 1987.
- [Deli86b] Delisle, N., Schwartz, M.: Contexts--A Partitioning Concept for Hypertext. Proceedings of the Conference on Computer-Supported Cooperative Work, 147-152. 1986.
- [Enge63] Engelbart, D.C.: A Conceptual Framework for the Augmentation of Man's Intellect. In: P.W. Howerton & D.C. Weeks (eds.) Vistas in Information Handling, Volume 1. London: Spartan Books. 1963.
- [Gull86] Gullichsen, E., D'Souza, D., Lincoln, P., Casey, T.: The PlaneText Book. MCC Tech. Rep. STP-333-86(P). 1986.
- [HT87] Proceedings of Hypertext '87, TR88-013, University of North Carolina, Department of Computer Science. ACM. March 1988.
- [McCr84] McCracken, D., Akscyn, R.: Experience with the ZOG Human-Computer Interface System. International Journal of Man-Machine Studies 21, 293-310. 1984.
- [Nels80] Nelson, T.: Replacing the Printed Word: A Complete Literary System. Proceedings of IFIP Congress 1980, North-Holland, 1013-1023. 1980.
- [Robe81] Robertson, G., McCracken, D., Newell, A.: The ZOG Approach to Man-Machine Communication. International Journal of Man-Machine Studies, 14, 461-488. 1981.

- [Trig86b] Trigg, R.H., Suchman, L., Halasz, F.G.: Supporting Collaboration in NoteCards. Computer-Supported Cooperative Work; CSCW 86 Proceedings, Austin, Texas. December 3-5, 1986.

- [Yank85] Yankelovich, N., Meyrowitz, N., van Dam, A.: Reading and Writing the Electronic Book. Computer, 18(10). 15-30. 1985.