HYPERMEDIA, INFORMATION SYSTEMS AND
ORGANIZATIONS: A RESEARCH AGENDA

Tomás Isakowitz
Leonard N. Stern School of Business
New York University
44 West 4th Street, Suite 9-79
New York, NY 10012-1126
(212) 998-0833
E-Mail: tisakowi@stern.nyu.edu

November 1992

HYPERMEDIA, INFORMATION SYSTEMS AND ORGANIZATIONS: A RESEARCH AGENDA

Tomás Isakowitz

Department of Information Systems
Stern School of Business
New York University
New York, NY 10012-1126

Abstract

Although hypermedia systems are becoming more sophisticated and readily available as off the shelf products, there are few hypermedia applications within business and industrial organizations. This paper argues that this phenomena is rooted in the concept of hypermedia applications as standalone programs. Larger acceptance of hypermedia within organizations will occur once this technology is better integrated with other organizational systems and applied to carefully selected tasks. The paper identifies three areas for research in this context: the tasks perspective, which deals with selecting tasks for which to develop hypermedia applications, the knowledge perspective, which deals with representing and managing the knowledge processed by organizations, and the integration perspective, which deals with technical issues in software integration. This paper suggests that solutions to the problems presented will prompt the acceptance of hypermedia technology within organizations.

1   Introduction

What is required for hypermedia to become an integral part of the technology platform of businesses and industrial organizations? This paper proposes a list of important research issues for promoting the acceptance of hypermedia within organizations.

Hypertext [6, 23] allows information to be organized so that users can access it at will. A typical hypertext system consists of *nodes*, that contain information, and of *links*, that represent relationships between nodes. *Buttons* or *anchors* are regions within nodes where links originate or terminate. To traverse a link, a user activates a *button*, e.g., clicking on it with the mouse. The set of all nodes and links is normally referred to as the *hyperdocument*. A collection of hyperdocuments is a *hyperbase*. *Hypertext systems* manage hyperbases and enable users to navigate within hyperdocuments. *Hypermedia* extends hypertext by incorporating multimedia. *Hypermedia nodes* may contain not only text, but also sound, video, graphics, etc. *Hypermedia buttons* may consist of gestures, soundbites, etc. Hypermedia systems provide the following functionality.

- **Annotation:** Users can add their own personal comments onto documents, as well as establish relationships among these without altering the documents themselves.

- **Navigation:** Users can explore hyperdocuments by following links from node to node. Navigational aids include graphical overviews and backtracking.

- **Structure:** Links can be used to establish various kinds of associations among information units. These include cross-references, hierarchies and the ability to share substructures of nodes and links among hyperdocuments.

1.1   Dimensions of the Research Problem

Organizational hypermedia is hypermedia technology adapted to the processing needs of organizations. A key difference between organizational processing and individual processing is the emphasis of the former on information sharing. The need to share information among individuals (e.g., members of a design

group), among groups (e.g., design engineering, manufacturing, marketing, etc.) and among organizations (e.g., McDonnell Douglas Aircraft Company and its 6000 suppliers) places significant demands on organizational information processing [9]. Information from different sources needs to be integrated and processes need to be coordinated. Thus, to be effective, organizational hypermedia should support integration, sharing and coordination.

Moreover, hypermedia should be ubiquitous. This can be achieved by realizing that hypermedia functionality can be offered as a service to other applications in the same way that database management systems provide data management services. An *organization's hyperbase* where all documents of the organization reside, is the counterpart of the organization's database. A *hypermedia engine* capable of processing requests for hypermedia service from applications, is the counterpart of the DBMS, which handles data management requests. For example, a decision support system (DSS), can use the services of the hypermedia engine enabling users to navigate among the elements of a decision process [2, 4, 15].

1.2 Issues for Research

This paper develops three research perspectives related to organizational hypermedia.

The tasks perspective. To promote the acceptance of hypermedia, hypermedia applications should be carefully selected which benefit the most from hypermedia's unique features. It is important therefore to investigate what kinds of tasks are best suited for hypermedia implementation. The tasks perspective, developed in section §2, proposes a framework within which this kind of research can be conducted and raises a number of questions pertaining hypermedia acceptance.

The knowledge perspective. Modern organizations are knowledge handlers. To support their information processing needs, hypermedia needs the ability to represent and manage knowledge structures. The knowledge perspective, developed in section §3, raises a number of important research issues that need to be resolved in this context.

The integration perspective. The diffusion of hypermedia functionality requires a seamless software integration between information system applications and hypermedia engines. However, most hypermedia systems are monolithic, self-enclosed programs unable

to interact with other applications [20]. This research perspective focuses on the technical issues that need to be resolved to support integration.

The three areas are interrelated. The kinds of tasks to which hypermedia should be applied qualifies the kinds of knowledge it needs to handle, this in turn qualifies the type of kind of integration required; and vice versa, integration places demands on the knowledge capabilities proposed for hypermedia, which in turn affect the kinds of tasks to which it can be applied.

This paper is organized as follows. Sections §2, §3 and §4 develop the three perspectives for research. Section §5 offers some concluding remarks.

2 The Tasks Perspective

Hypermedia is not a panacea for all tasks. It is important to identify the tasks for which hypermedia is best suited. The information processing framework developed by Galbraith [11] and later refined by Tushman and Nadler [32] can be used to research this question. This framework conceives of organizations as information-processing networks with information processing requirements and information processing capacities. As long as there is a match between the requirements and the capacities, the organization will function effectively. The framework is shown in figure 1.

2.1 Task Dimensions

Daft and Lengel [7] identified two key qualifiers of work tasks that determine information processing needs: uncertainty and equivocality. Organizations process information for two reasons: to reduce uncertainty and to reduce equivocality.

Uncertainty is the "relative difference in the amount of information required and the amount possessed by the organization" [11]. As information increases, uncertainty decreases.

Equivocality is the degree of ambiguity inherent in a task. Low equivocality suggests a single interpretation [31]. "High equivocality means confusion and lack of understanding. Equivocality means that asking a yes-no question is not feasible [7]."

A key difference between equivocality and uncertainty is that while uncertainty can be reduced by increasing the amount of data, this does not help to reduce equivocality. Faced with an equivocal problem, managers might not know what questions to ask,
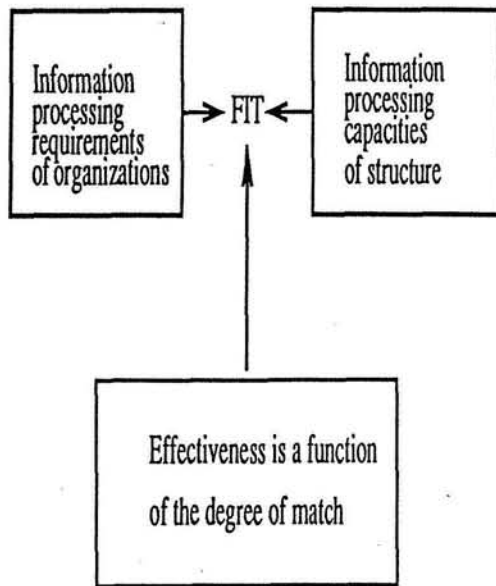
Figure 1: The Information Processing model of organizations. (Adapted from Tushman and Nadler)

whereas when faced with uncertainty, managers know what questions to ask, but might have difficulties finding the data needed to answer these questions.

Uncertainty can be reduced by either decreasing the information processing needs of organizations, for example, by changing the structure of the organization, or by increasing its information processing capacities [11], for example, by introducing a new information system. High equivocality stimulates frequent and informal face-to-face information cycles among managers. The key factor in equivocality reduction is the extent to which structural mechanisms facilitate the processing of rich information. Equivocality leads to rich exchange of views among managers to define problems and resolve conflicts [7].

The information processing requirements for work tasks can be determined by classifying the tasks along the dimensions of uncertainty and equivocality. A task with high uncertainty will require high volume of information processing; while high equivocality requires systems that help disambiguate problems.

2.2 Targeting Hypermedia Applications

The adoption of hypermedia depends on its ability to bridge the gap between information processing needs and capacities. Where will hypermedia have the strongest impact? The information-processing framework can be used to answer this question. First, the extent by which hypermedia functionality reduces uncertainty and equivocality should be determined. Next, tasks with corresponding levels of uncertainty and equivocality should be chosen as candidate applications for hypermedia implementation.

The problem is now to determine hypermedia's contribution to uncertainty and equivocality reduction. This is a difficult research problem because of the lack of techniques to determine these levels. However, the following analysis might help to clarify the problem.

Since uncertainty is reduced by increasing the amount of information, the multimedia component of hypermedia can help reduce its uncertainty level. This is a consequence of the high information bandwidth that results from the inclusion of the different media– sound, video, graphics, etc.– in hypermedia systems. What levels of uncertainty does this multimedia functionality succeed in managing? Experience from other technologies such as teleconferencing, can provide some clues.

Hypermedia's ability to manage relationships among various objects, its *hyper* component, can help reduce equivocality. Users confronted with a confusing task can explore the problem domain to disambiguate it, by following embedded hypermedia links. Since asking a "yes-no question is not feasible [7]," navigation is an appropriate alternative. Moreover, through its hyperbases, hypermedia provides structural mechanisms to facilitate the processing of rich information, as required for equivocality reduction. A key problem is to determine the extent to which equivocality is reduced. Annotation aside, hypermedia can not help in a task where no structure exists. Nor would it be helpful at the other extreme, where tasks have a distinctive sequential organization.

Determining the reduction levels in the presence of both the multimedia and the hyper components of hypermedia is also problematic since there are interdependencies among both components. However, it would seem that the added benefits help reduce uncertainty and equivocality levels even more. Experiments should be devised to measure these levels.

In sum, classifying hypermedia in terms of uncertainty and equivocality reduction can help target hypermedia applications for success.

3 The Knowledge Perspective

A central issue arising from the knowledge-intensive nature of the organizations is their need to manage large knowledge bases. Hypermedia can help represent and access knowledge about information held in the organization.

3.1 Hypermedia and Organizational Memory

In their role as knowledge handlers, organizations accumulate various kinds of information. An organization can increase its efficiency by maintaining a well-organized repository with all this information. Such a repository is called an **organizational memory**.

Organizational memories contain various entities among which we can distinguish: facts (e.g., *the design of the wing was completed on 6/16/92*), positions (e.g., *node airfoil-inadequate objects to node wing-design-satisfactory*), and ephemeral events (e.g., *group meeting 6/26/92 at 5pm*). Memory entities are related in ways that are continuously updated as the environment evolves. In principle, the organizational memory lends itself to a hypermedia representation: entities are represented by nodes, relationships by links. One important aspect of organizational memory is its growth. As new memory entities are added, they have to be properly related to the memory base.

Various hypermedia systems deal, albeit implicitly, with organizational memories. Ted Nelson's Xanadu system [22] is, in fact, an organizational memory of all literary material. Engelbart [9] proposes a *Hyperdocument Library System* where all documents of an organization can be stored in a library-like service that provides linking capabilities. Answer Garden [1] is a tool to disseminate expert knowledge to end-users through a question-answer paradigm. In essence, the user navigates a network (actually a tree) by responding to questions. Answer Garden exhibits organic growth when experts respond to user's requests by adding or updating information. The Virtual Notebook System [13] has been developed for cooperative research activities. It is based on the notebook metaphor. A common interlinked repository is accessed through individual notebook pages. The system grows as users add information to this repository. CORE [8] deals specifically with hypermedia academic journals. The network represents an academic memory that grows as new articles are included.

3.2 Problems with Organizational Memory

Several problems arise when considering a hypermedia implementation for organizational memory.

How can consistency of an organizational memory hyperbase be ensured? **Pollution** is an important problem with hypermedia in general and organizational memories in particular. Memory entities are inter-related via links that are continuously updated as the environment evolves. If the hypermedia engine has no understanding of the relationships it represents, links have to be manually crafted by experts. Besides its high cost in terms of time, this approach becomes unsafe when we consider large networks that span across organizations, or even across groups within the same organization. Quality control of manual link creation is, at best, difficult to enforce, even in the presence of typed links and nodes, because there are no semantic constraints to enforce consistency, as in databases. Hence, instead of representing organizational memory, the hypermedia network may reflect the inherently biased personal perceptions of the individuals responsible for maintaining the links. This has the potential to dissociate the hypermedia network from what it represents. Important relationships among facts might be absent (e.g., *group-meeting-6/26/9 at 5pm* might not be linked to *group meeting canceled due to ...*), relationships might be erroneous (e.g., *group meeting 6/26/92 at 5pm* might be associated by a *canceled* link to *wing design does not withstand required loads.*) As a result, errors can pollute the network and turn it useless.

Another problem is memory **growth**. Although expansion is inherent in hypermedia systems, there are no set mechanisms to incorporate new information into the existing hyperbase. New entities are usually related to older ones already in the memory. These relationships are bidirectional. For example, a new regulation on employee's compensation should be related to older ones. And vice-versa, links should relate the older regulation to their newer versions so that readers can be made aware of any changes. How will these relationships be established? A manual method can not be effective because individuals may not be aware of the full contents of the hyperbase. Therefore, they will not be able to establish all required links.

Growth brings along **waste**, i.e., information that looses its relevance over time, remains indefinitely in place. This results not only in a waste of space, but also in a waste of time when garbage nodes distract users during browsing. How can these nodes be removed? Can garbage collection techniques from programming languages be applied?

Memory restructuring is another challenge. Human memory can restructure itself in response to changing circumstances. This enables relevant events to be retrieved easier than less important ones. What kind of hypermedia restructuring mechanisms are needed to support this? Stotts and Furuta [28] describe time-based adaptation mechanisms of hyper-

text networks. Some links may appear or disappear according to the amount of time spent by the user on a node. Similarly, node contents can be re-evaluated based on timing parameters. More work in this area is needed to achieve the kind of restructuring required for organizational memories.

Engelbart [9] identifies **interoperability** between knowledge domains as key to the coordination of tasks within and across organizations. Knowledge domain interoperability is the ability to share and exchange information among systems that operate on different knowledge domains (marketing, manufacturing, etc.) Within the organizational memory, for example, domain interoperability translates into the need to establish relationships among entities originating in diverse knowledge domains. For example, the representations for a regulation in separate divisions of a law firm have to be compatible, if not identical, to support cooperative work. Since internal languages of various systems will usually be different, some kind of translation mechanism is needed. Such translation is only viable if each system is able to represent knowledge at a sufficiently high level of abstraction. Hypermedia constructs capable of representing knowledge, as opposed to simple node and link constructs, are required to support domain interoperability. If the hypermedia system only maintains nodes and links, no automatic information exchange across different domains can take place. It is not clear that the proposed hypermedia design languages [12, 19, 21] have the required expressive power to help with domain interoperability.

Organizational memories need to be flexible enough to accommodate new kinds of information and relationships. One problem with current hypermedia design models is their lack of flexibility. It is hard to change schema definitions or logic laws as systems evolve. However, such flexibility is needed, especially given the intra- and inter-organizational requirements. For example, as Malone et. al [17] report, users might realize that they need new kinds of links, or that previously defined fields should allow contents of different types. (Such fields are called *soft buttons* [29]). In [17] the need for semi-formal structures to deal with such problems is identified. The nature of these structures in the context of hypermedia needs to be further studied.

The challenges to hypermedia organizational memories just presented are many and difficult to tackle. Investigating these areas will shed light not only on this application of hypermedia, but on the hypermedia concept at large.

4   The Integration Perspective

Meyrowitz [20] argues that integrating hypermedia with other applications will clear the way to hypermedia's adoption within organizations. Integration requires more than a simple import/export facility because of its dynamic nature. Applications, e.g., a DSS, will run concurrently with a hypermedia engine, onto which they delegate hypermedia operations such as navigation and annotation.

Integrating hypermedia into other applications is a challenging problem because most hypermedia systems have been designed as standalone applications with minimal support for exchanging information with other software. Facing this challenge requires rethinking the basic hypermedia architecture and devising information exchange techniques. This section proposes research issues in both areas.

4.1   Architectures for integration

Most hypermedia systems are not appropriate for integration because they do not include dynamic exchange mechanisms to allow them to execute concurrently with other programs. This is exemplified by the HAM architecture [5], on which several hypermedia systems are based. HAM has no provisions for enabling and information exchange with other applications. Even recent hypermedia models that propose layered architectures [14, 10, 16, 30, 24, 30], do not contemplate exchange mechanisms. The toolkit approach to hypermedia [26, 27] and Sun's Link Service [25], although based on the service metaphor, suffer from the same shortcoming.

How can these exchange mechanisms be provided for? One possible approach, explored by Bieber ([4, 3] and the paper in this proceedings) and by Isakowitz and Stohr [15], proposes a layered architecture which differentiates among *back-end* servers responsible for managing the communication with back-end applications, *front-end* servers responsible for managing user interaction, and a hypermedia engine responsible for hypermedia functionality. The layered architecture requires messages between the external systems (backend application or front-end interface) to be routed thorugh the hypermedia layer. This might be problematic because it requires the definition of an exchange protocol that all parties involved are required to adopt. It is not clear what such a protocol should be. Should it involve direct requests for hypermedia actions as in Bieber's approach? Or should the protocol be used to communicate application state-changes, as in as in Isakowitz and Stohr [15]?

Another problem with the layered architecture is that it requires significant efforts from system software developers, i.e. those that develop the back and front-end shells, who need to supply their expertise for proper integration. Moreover, this approach to integration requires an active role from the back-end application, since it has to send messages to the hypermedia engine. Is there a way of automating the communication mechanisms so that messages do not have to be designed specifically for each application?

A related problem is the integration of old applications with hypermedia. The approaches described above would require extensive re-engineering. Since about 90 % of software budgets are devoted to maintenance, it makes sense to establish mechanisms to upgrade existing applications with hypermedia functionality. An alternative is the development of a shell that provides only partial-hypermedia functionality, but incorporates all applications (new and old).

Alternatives to the layered architecture approach should also be explored. A hypermedia software environment would encourage the development of hypermedia applications. The development environment would ensure that the applications are able to intercommunicate and exchange information. What should such an hypermedia software development environment be like? What kinds of features should it provide? Lange –in his paper in this mintrack– for example, proposes an object-oriented hypertext environment for the development of hypermedia applications.

4.2 Dynamic Information Exchange

As a back-end application communicates with a hypermedia engine, it has to be able to specify, dynamically, the nodes and links within the application. For example, as a DSS generates reports from models and data, it has to let the hypermedia engine know which new nodes and links should be created, and what links are to be traversed. One problem with the basic node and link model is that it requires manual node and link creation, and is thus unfit for dynamic information exchange. Even typed nodes and links are not helpful here, since what is needed is an automatic, real-time method to identify hypermedia structures in the application domain.

A related problem mentioned in §3.2 is that the basic node-and-link framework is not rich enough to represent domain knowledge. This has sparked research into knowledge-oriented hypermedia systems like Aquanet [19], HDM [12] and WebTalk [21] that are capable of representing domain knowledge via schemata. The schemata assist in mapping domain entities such as the models and reports of a DSS, into hypermedia entities such as nodes and links. However, the schemas only assist in the design of hypermedia applications, not in the communication between a hypermedia engine and an application. What method can support the dynamic mapping of domain entities into hypermedia entities? Probably different methods will be required, one for nodes, another for links. The following list provides some basic proposals.

To identify nodes, applications could provide *cues*, in the form of regular expressions. A pattern matcher would then determine the kind of node a message represents. For example, an object matching the pattern *Memo* would be recognized as a node of type Memo. Alternatively, the application might use an object-oriented approach whereby each object has the ability to determine the kind of hypermedia object it should be mapped to.

Identification of anchors, buttons and links, which require the identification of domain relationships, presents similar challenges. A possible solution proposed by Isakowitz and Stohr [15] uses schemata, as described above, as a basis for communication. Through the schemata, a special engine recognizes the links to establish among nodes and directs the hypermedia engine to create the links.

Once the nodes and links in a domain application have been identified, the hypermedia engine can present a hyperbase to the user. How should user-interaction be handled? When the user manipulates hypermedia objects, his intentions are to invoke back-end application functions. For example, clicking on a button labeled execute should trigger a DSS application to generate a new report. There is no straightforward way of determining what kind of application procedure to invoke.

Are there other ways, without using schemata, to enable the dynamic exchange between back-end applications and the hypermedia engine? Is it possible to adopt network packaging protocols, EDI methods or inter-process communication techniques? Alternatively, a hypermedia software development environment could solve all these problems by automatically incorporating hypermedia functionality within applications. The afore-mentioned problems, notwithstanding this approach would not successfully integrate all corporate applications unless they all were developed within the same software development environment.

5 Conclusion

Although the field of hypermedia has evolved significantly over the last decade, the fact remains that few organizational applications of hypermedia can be found. At the root of this phenomena lie two interrelated issues. First, hypermedia systems are for the most part standalone, isolated applications that users view more as curiosities than as useful productive applications. Second, hypermedia is not perceived to fulfill to real organizational needs. Organizations will only realize the need for hypermedia applications once hypermedia is applied to key work tasks where they can make a difference.

Once hypermedia systems become open systems that make its functionality readily available to other application software, we will observe increased levels of hypermedia adoption. A central organizational hypermedia information exchange will be at the foundation of software platforms in organizations. All applications and documents will participate in this exchange. Users will be able to access any program and any information through machine-created links that connect related objects. Hypermedia will follow Malone's three stage pattern for technological adoption [18]. First, the open systems architecture will facilitate the implementation of hypermedia applications. Second, as more applications participate in the hypermedia information exchange, the resources made available through the exchange will lure more applications to participate in it. Third, the hypermedia exchange will become so vital to information processes, that no application will be able to afford not being part of it. At this point we will have ubiquitous hypermedia functionality at all levels of the organization.

However, several important problems stand in the way of this developments. Most prominently, the following five research problems will dictate research in the hypermedia field over the coming years.

1. *Proper task identification:* to take the *hype* out of hypermedia, hypermedia should be applied only to those tasks that really require it. The identification of the appropriate levels of uncertainty and equivocality that hypermedia can manage is an important step towards finding the right match. In addition, the interdisciplinary nature of this research will prompt collaboration with researchers in other areas, increasing hypermedia's exposure.

2. *Management of organizational memories:* as hypermedia systems become the repository of information across the organization, several management issues arise. Among these, the growth, restructuring and consistency of the hyperbase will take precedence.

3. *Knowledge representation:* the shortcomings of the node-and-link model for hypermedia will become even more obvious as research in the two previous areas evolves. At the heart of the matter lies the problem of automatically mapping domain concepts into hypermedia entities. Knowledge management tools of this nature will, in turn, ease the implementation of more hypermedia applications.

4. *Software integration:* to overcome the isolated nature of hypermedia applications and to enable the envisioned organizational hypermedia exchange, new hypermedia architectures are required. Innovative design proposals should be investigated.

5. *Exchange protocols:* hypermedia information exchange requires not only novel architectures, but innovative information exchange protocols with the ability to establish dynamic communications among the hypermedia engine and the various applications in an organization.

Within a decade we will see as a result of these research directions, an organizational-wide hypermedia exchange in which various kinds of applications participate by infusing and extracting information on a dynamic basis.

References

[1] Mark S. Ackerman and Thomas W. Malone. Answer Garden: A Tool for Growing Organizational Memory. In *Proceedings of the ACM Cpnference on Office Information Systems*, pages 31–39, 1990. (Also as MIT-Sloan working paper # 108.

[2] Hemant Bhargava, Michael Bieber, and Steven O. Kimbrough. Oona, Max and the WYWWYWI Principle: Generalyzed Hypertext and Model Management in a Symbolic Programming Environment. In Janice I. DeGross and Margarethe H. Olson, editors, *Proceedings of the Ninth ICIS*, pages 179–192, 1988.

[3] Michael P. Bieber. Issues in Modeling a *Dynamic* Hypertext Interface. In *Hypertext' 91 Proceedings*, pages 203–218. ACM, ACM Press, December 1991.

[4] Michael P. Bieber. Automating Hypertext for Decision Support. *Hypermedia*, (forthcoming).

[5] B. Campbell and J. M. Goodman. HAM: A general Purpose Hypertext Abstract Machine. *Communications of the ACM*, July 1988.

[6] Jeff Conklin. Hypertext: An Introduction and Survey. *IEEE Computer*, 20(9):17–41, September 1987.

[7] Richard L. Daft and Robert H. Lengel. Organizational Information Requirements, Media Richness and Structural Design. *Management Science*, 32(5):554–571, 1986.

[8] Dennis E. Egan, Michael E. Lesk, R. Daniel Ketchum, Carol C. Lochbaum, Joel R. Remde, and Thomas K. Landauer. Hypertext For the Electronic Libaray? CORE Sample Results. In *Hypertext' 91 Proceedings*, pages 299–312. ACM, ACM Press, December 1991.

[9] Douglas Engelbart. Knowledge Domain Interoperability and an Open Hyperdocument System. In *Proceedings of the Conference on Computer-Supported Cooperative Work*, pages 143–156, Los Angeles, CA, October 7–10 1990.

[10] Richard Furuta and P. David Stotts. The Trellis Hypertext Reference Model. In Judi Moline, Dan Beningni, and Jean Baronas, editors, *Proceedings of the Hypertext Standartization Workshop*, pages 83–93, Gaithersburg, MD 20899, March 1990. National Institute of Standards and Technology, NIST special publication 500-178.

[11] Jay Galbraith. *Designing Complex Organizations*, chapter 2 - Information Processing Model, pages 8–19. Addison-Wesley Publishing Company, Inc., 1973.

[12] Franca Garzotto, Paolo Paolini, and Daniel Schwabe. HDM - A Model for the Design of Hypertext Applications. In *Hypertext'91 Proceedings*, pages 313–328. ACM Press, December 1991.

[13] G. Anthony Gorry, Kevin B. Long, Andrwe M. Burger, Cynthia P. Jung, and Barry D. Meyer. The Virtual Notebook System: An Architecture for Collaborative Work. *Journal of Organizational Computing*, 1(3):233–250, 1991.

[14] Frank G. Halasz and Meyer Schwartz. The Dexter Hypertext Reference Model. In Judi Moline, Dan Beningni, and Jean Baronas, editors, *Proceedings of the Hypertext Standartization Workshop*, pages 95–133, Gaithersburg, MD 20899, March 1990. National Institute of Standards and Technology, NIST special publication 500-178.

[15] Tomas Isakowitz and Edward A. Stohr. Hypertext-based Relationship Management for DSS. Working paper IS-92-22, Center for Research in Information Systems, New York University, Information systems Department, New York, NY 10003, 1991.

[16] Danny B. Lange. A Formal Model of Hypertext. In Judi Moline, Dan Beningni, and Jean Baronas, editors, *Proceedings of the Hypertext Standartization Workshop*, pages 145–166, Gaithersburg, MD 20899, March 1990. National Institute of Standards and Technology, NIST special publication 500-178.

[17] Thomas W. Malone, Keh-Chiang Yu, and Jintae. Lee. What Good are Semistructured Objects? Adding Semiformal Structure to Hypertext. Working Paper 102, Massachusetts Institute of Technology, Cambridge, MA, June 1989.

[18] T.W. Malone, J. Yates, and R.I. Benjamin. Electronic Markets and Electronic Hierarchies. *Communications of the ACM*, 30:484–497, 1987. (Reprinted in I. Greif, ed., Computer Supported Cooperative Work, Los Altos, CA: Morgan Kaufmann Publishers, 1988).

[19] Catherine C. Marshall, Frank G. Halasz, Russell A. Rogers, and William C. Janssen Jr. Aquanet: a hypertext tool to hold your knowledge in place. In *Hypertext'91 Proceedings*, pages 261–275. ACM Press, December 1991.

[20] Norman Meyrowitz. The Missing Link: Why we're all doing hypertext wrong. In Edward Barrett, editor, *The Society of Text*, pages 107–14. MIT Press, Cambridge, MA, 1989.

[21] Jocelyne Nanard and Marc Nanard. Using Structured Types to incorporate Knowldege in Hypertext. In *Hypertext'91 Proceedings*, pages 329–342. ACM Press, December 1991.

[22] Theodor H. Nelson. Replacing the Printed Word: A Complete Literary System. In S. H. Lavington, editor, *IFIP Proceedings*, pages 1013–1023. North Holland, 1980.

[23] Jakob Nielsen. *HyperText & HyperMedia*. Academic Press, 1990.

[24] H. Van Dyke Parunak. Toward a Reference Model for Hypermedia. In Judi Moline, Dan Benigni, and Jean Baronas, editors, *Proceedings of the Hypertext Standardization Workshop*, pages 197–211, Gaithersburg, MD 20899, January 1990. National Institute of Standards and Technology, NIST Special Publication, U.S. Government Printing Office.

[25] Amy Pearl. Sun's Link Service, A Protocol for Linking. In *Hypertext'89 Conference Proceedings*, pages 137–146. ACM Press, November 1989.

[26] J. J. Puttress and N. M. Guimaraes. The Toolkit Approach to Hypermedia. In A. Rizk, N. Streitz, and J. André, editors, *Proceedings of the European Conference on Hypertext*, pages 25–37, France, November 1990. INRIA, Cambridge University Press.

[27] Mark Sherman, Wilfred J. Hansen, Michael McInerny, and Tom Neuendorfer. Building Hypertext on a Multimedia Toolkit: An Overview of Andrew Toolkit Hypermedia Facilities. In A. Rizk, N. Streitz, and J. André, editors, *Proceedings of the European Conference on Hypertext*, pages 13–24, France, November 1990. INRIA, Cambridge University Press.

[28] P. David Stotts and Richard Furuta. Dynamic Adaptation of Hypertext Structure. In *Hypertext' 91 Proceedings*, pages 219–232. ACM, ACM Press, December 1991.

[29] Norbert Streitz. The Role of Hypertext for CSCW Applications - Panel discussion. In *Hypertext'91 Proceedings*, pages 369–378. ACM Press, December 1991.

[30] Craig W. Thompson. Strawman Reference Model for Hypermedia Systems. In Judi Moline, Dan Beningni, and Jean Baronas, editors, *Proceedings of the Hypertext Standartization Workshop*, pages 223–246, Gaithersburg, MD 20899, March 1990. National Institute of Standards and Technology, NIST special publication 500-178.

[31] Linda Klebe Trevino, Richard L. Daft, and Robert H. Lengel. *Organizations and Communication Technology*, chapter 5 - Understanding Manager's Media Choices: A Symbolic Interactionist Perspective, pages 71–94. Sage Publications, 90?

[32] Michael L. Tushman and David A. Nadler. Information Processing as an Integrating Concept in Organizational Design. *Academy of Management Review*, pages 613–624, July 1978.