Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 1998 Proceedings

International Conference on Information Systems (ICIS)

December 1998

# IDM: A Methodology for Intranet Design

Seung Lee
*Hofstra University*

# IDM: A METHODOLOGY FOR INTRANET DESIGN

**Seung C. Lee**
Hofstra University
U.S.A.

## Abstract

Organizations have been looking for efficient and effective information communication to create and sustain a competitive advantage. The recent surge in the Web has created new interest in intranets as an alternative for cost-effective and boundaryless information exchange. Amid the interest, one of the problems in developing intranets is a dearth of design methodologies. This article presents an intranet design methodology focusing on the navigation and user-interface designs. Based on various classification schemes such as attribution, the navigation design organizes information into global and local structures using the concepts of metainformation structure and information structure. The user-interface design utilizes Minsky's *frame* to map the elements of the metainformation structure and information structure onto the actual user-interface objects.

**Keywords:** Intranet, navigation design, global and local information structures, user-interface design.

## 1. INTRODUCTION

As the basis of competition has been shifting more and more to the creation and assimilation of information, efficient and effective information communication has become a critical source of competitive advantage for organizations (Porter 1990). This implies that information communications should be *timely*, *up-to-date*, *consistent*, and *cost-effective* as well as *fast* and *unlimited* (Welch 1993) in order for organizations to create and sustain competitive advantage. This comes from reducing barriers of information flows, which are frequently experienced with the traditional information systems, to achieve competitive advantage through the creation and communication of information (e.g., Jackson 1993).

Traditional information systems based on mainframes and terminals, which were developed primarily to support business functions, do not provide most information in the fashion described above because of inefficient user interfaces, platform-dependency, and multiple software licenses (Diebel and Greens 1995). Moreover, the linear way in which traditional information systems provide information does not fit into nonlinear human thought process (Waldorp 1992).

This contradiction between how the mind works and the way information is presented in virtually all media has long been recognized (Bush 1945). Two decades after Bush's vision, Ted Nelson coined the term *hypertext*, which he defined simply as nonlinear reading and writing (Nelson 1965). This paradigm was continued through Douglas Engelbart's NLS (Engelbart 1968). Although the work of these early researchers produced several innovations and has had definite impact on the field, computer hardware limitations impeded the development and widespread use of hypertext systems until recently (Perry 1987).

Recent advances in computer hardware and, especially, the recent surge in the Web have created new interest in hypermedia (Bieber and Isakowitz 1995). Organizations have begun to utilize the Web through intranets to share information with their employees in an efficient and effective manner (Challa and Redmond 1996). Amid the utilization, one of the problems in developing intranets is a lack of design methodologies (Bieber and Vitali 1997). This article presents a methodology for design of intranets called the intranet design methodology (IDM), focusing on the navigation and user interface design.

A problem statement and a review of the existing hypertext/hypermedia development methodologies are presented in the following two sections, respectively. Section 4 presents a detailed discussion of the IDM. Although it is necessary and desirable to present details of the implementation of the prototype intranet that the author developed using the IDM, only brief examples are provided due to the space limitations. Conclusions and opportunities for future research are discussed in the last section.

## 2. PROBLEM STATEMENT

In general, every non-trivial information system requires an application of a certain methodology and tools in its design and implementation. An intranet is an information system based on the Web technology and hypermedia paradigm. Thus, to truly realize the potential of hypermedia via an intranet within an organizational setting, we need a formal methodology and a set of tools. In this context, the following argument by Bieber and Isakowitz (1995, p. 26) is worth noting:

> We lack guidelines and tools to design and develop hypermedia applications. This is especially true for commercial scale systems which involve frequently changing information. Without such guidelines and tools, the ever-growing network of interlinked applications is becoming increasingly spaghetti-like and hard to maintain.

An intranet is by nature a large-scale and complex application because it lets users exchange a large amount of information across organizations, enterprises, and even global boundaries (McCune 1996). On the other hand, Bieber and Vitali (1997, p. 62) point out that "in the rush to acquire and retrofit Web applications, organizations risk bypassing the Web's greatest supplemental benefit—hypermedia." They argue that the main reason for this is a dearth of appropriate methodologies. Furthermore, as Garzotto (1995) points out, "it is no longer feasible to neglect design issues when managing the complexity of building large hypermedia, nor is it possible to build complex hypermedia features into our systems without doing preliminary analysis at the conceptual level." Again, "the range of hypermedia applications is so broad that no single formal design technique is relevant for designing all of them " (Nanard and Nanard 1995, p. 50).

## 3. A REVIEW OF THE EXISTING METHODOLOGIES

We have recently seen several hypermedia development methodologies. This section presents a review of two methodologies: hypertext design model (HDM) (Garzotto, Paolini and Schwabe 1993) and relationship management methodology (RMM) (Isakowitz, Stohr and Balasubramanian 1995). These methodologies were chosen because of their non-object-oriented nature.

HDM is the first step toward defining a general-purpose model for hypertext development. It is argued that HDM has some innovative features, including the notion of perspective, identification of different categories of links, and the distinction between hyperbase and access structures. HDM defines several design primitives such as entities to represent information elements and several link categories to connect them to each other.

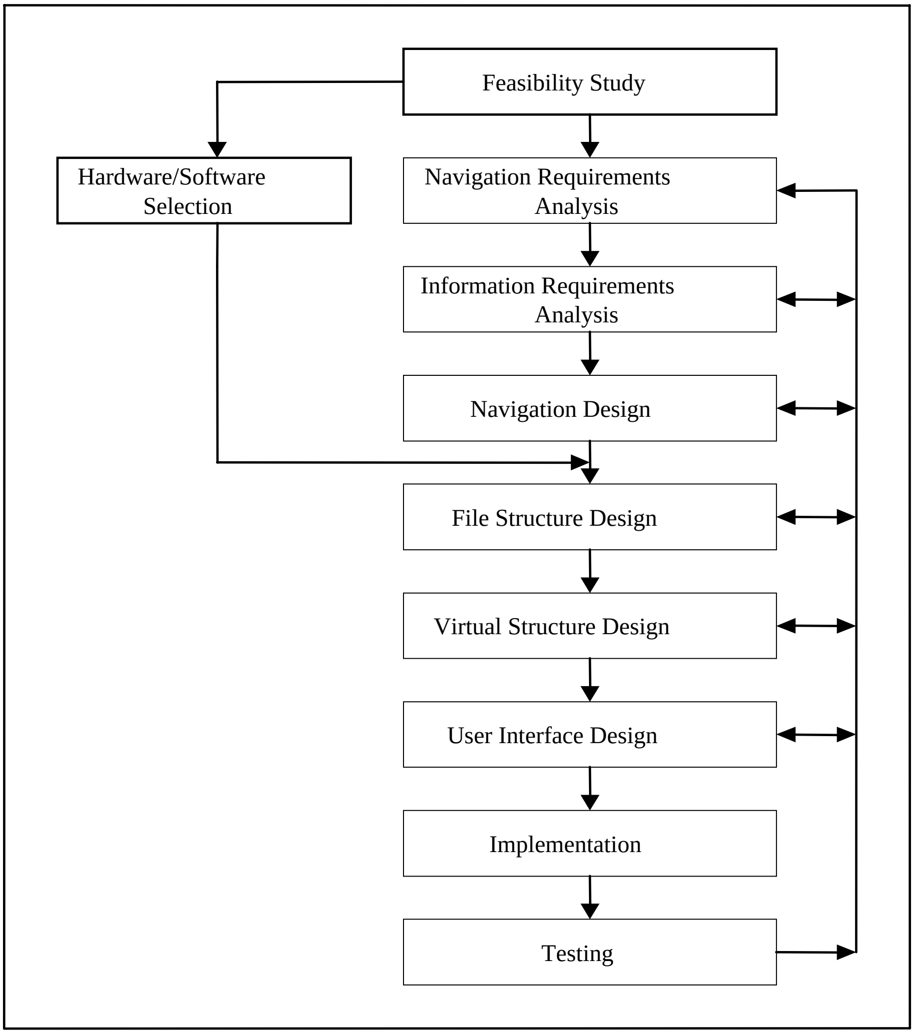
Even though HDM provides useful design primitives, there are some issues that should be noted. First, it simply provides a static set of structures of information chunks and a static set of link types. This hurts extensibility and flexibility of the model. HDM ignores the possibility that an entity type may have sub-entity types. In that case, the three information structure constructs and the three link types would be too limited. Second, HDM does not provide a mechanism for defining the application links among entity types and their instances. It just admits that application links can be identified by semantic properties of information structure constructs. Third, HDM has no specific guidelines for constructing outlines that provide users with navigation entry points.

Isakowitz, Stohr and Balasubramanian argue that RMM is the first full hypermedia design methodology. As pointed out by Schwabe and Rossi (1995), however, RMM focuses on navigation design while overlooking navigational requirements analysis, information requirements analysis, and user interface design. Navigation design begins with entity-relationship modeling and ends with a diagram called relationship management data model that defines access structures.

RMM seems very easy to use and understand, but it lacks some required features in developing hypermedia applications. Besides the indication by Schwabe and Rossi, RMM seems better suited for smaller applications because other aspects of the hypermedia, such as overall structure of information, are not considered. If RMM were implemented for a large document such as a training manual, it would be difficult to handle the large number of entities that could be found in the document. Furthermore, Isakowitz, Stohr and Balasubramanian suggest that RMM be used for domains that are updated on a relatively frequent basis. If a large document needs to be updated frequently, the designer must repeat the steps of the navigation design to update content of the hypermedia developed using the RMM. Finally, it is believed that the designer should divide a large document into several parts for easier navigation design. However, no such method is provided in RMM.

## 4. THE INTRANET DESIGN METHODOLOGY (IDM)

Figure 1 depicts the entire cycle of the IDM. As shown by arrows, this methodology is essentially iterative between various phases to provide feedback loops whereby design quality can be improved through the evaluation and process recursion. It supports abstraction and instantiation mechanisms that enable the designer to alternate between top-down and bottom-up approaches. At the abstract level, information to be put on an intranet is organized into the global and local structures to enhance the user's global and local coherence and to reduce cognitive overhead. At the instance level, application components can be objectively evaluated, subsequently initiating redesign and reconstruction activities. One of the useful mechanisms for moving between the abstract and instance levels is prototyping (Diaz and Isakowitz 1995). A prototype intranet was developed using the IDM and will be presented briefly in the implementation phase.




This methodology is essentially iterative and the feedback loops are shown on the right side of this figure. Hardware/software selection is important, since it may affect the overall performance of an intranet. Feasibility study is beyond the scope of this method.

**Figure 1. The IDM Methodology**

As shown in Figure 1, the IDM consists of ten phases. This article, however, focuses on the navigation requirements analysis, navigation design, and user-interface design. Feasibility study and hardware/software selection are beyond the scope of this article. The feasibility study is well summarized in Davis and Olson (1985). Hardware/software selection will be decided within a given organizational setting. Information requirements analysis, file structure design, and virtual structure design need more investigation. Only brief explanations for these phases and other phases are presented below.

### 4.1 Navigation Requirements Analysis

In this phase, both *navigation domains* and *navigation thresholds* are identified. Navigation domains are sub-application domains. For example, if a company is the application domain of an intranet and it has several branch offices, identifying navigation domains would result in a list of branch offices and headquarters. Note that if an application domain is small, the domain itself can be a navigation domain. Once the navigation domains are determined, navigation thresholds are identified for each navigation domain. A navigation threshold is a group of *related* information elements where users start navigation. For example, the navigation threshold *people* may contain information about employees for a branch. Other navigation thresholds may include *product*, *service*, and *document*.

### 4.2 Information Requirements Analysis

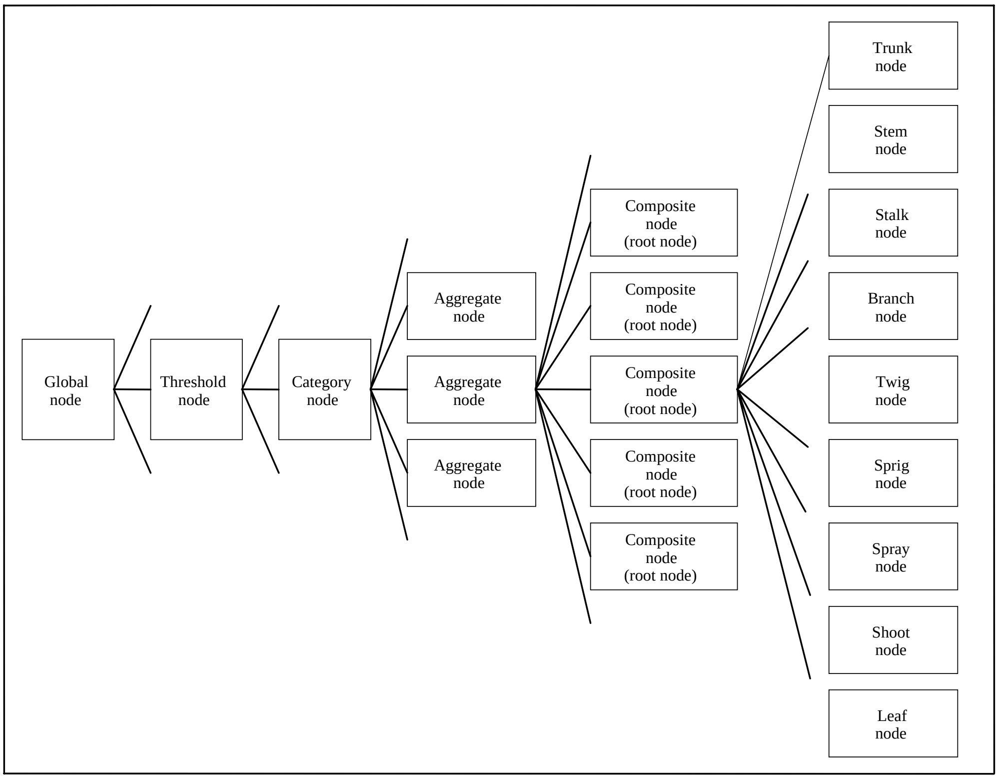
Based on the navigation requirements analysis, user information requirements for each navigation domain are identified and categorized into navigation thresholds in the information requirements analysis. Byrd, Cossick and Zmud (1992) summarized various requirements analysis techniques. For this phase, any of the techniques may be used. This phase concerns information screening where the designer may add missing information or remove redundancy. It also concerns information security where access privileges for each user can be defined based on the degree of confidentiality. Information screening involves (1) sorting information identified into navigation thresholds for each navigation domain, (2) removing redundancy, (3) adding missing information, and (4) classifying information elements for security. The last issue can be incorporated into the navigation design by marking a node.

### 4.3 Navigation Design

The information identified in the previous step should be organized into nodes with necessary links. For this purpose, two concepts are introduced: *metainformation structure* and *information structure*. The metainformation structure is used to organize information in the form of abstract nodes with their links. The information structure defines the local structure of each node of the metainformation structure using classes, subclasses, instances, attributes, attribute values, and links. They are derived from the notion of structure and attribution.

According to Simon (1962), hierarchical structure is a major facilitating factor enabling us to understand, to describe, and even to see complex objects and their parts. The notion of structure (usually a hierarchical network) has been a part of most hypertext systems since the time of the NLS (Engelbart 1968). The metainformation structure and information structure use the notion of generalization/specialization, whole/part, and other associations including spatial inclusion, temporal inclusion, and attribution.

Attribution plays an important role in developing the information structure. In this article, attributes of an object are divided into *standard* attribute and *custom* attribute. A standard attribute can be defined as an attribute that describes “standard” properties of an object such as *name* and *age*. A custom attribute is an optional or “customizable” attribute such as photograph. Custom attributes also include a special attribute “search” for multiple-instance classes such as *person*. Standard attributes are further divided into *intrinsic*, *associative*, and *conditional attributes*. An intrinsic attribute is an attribute that describes the very basic properties of an object. This sort of attribute has a single, primitive value such as *age*. An *associative attribute* is a composite attribute where the values are comprised of objects. A value of an associative attribute provides a link to a node that contains the description of the value. For example, if the value of *teaching* attribute is IS101 and detail of the course is defined as a node, IS101 becomes a link to the course node. This type of link is called the cross link. A *conditional attribute* of an object is an attribute that has a limited set of values whereby the object can be classified. The class *person* can be classified by its attribute *ethnic group* (e.g., Asian).




A composite node is a root node; the top most node is called a global node.

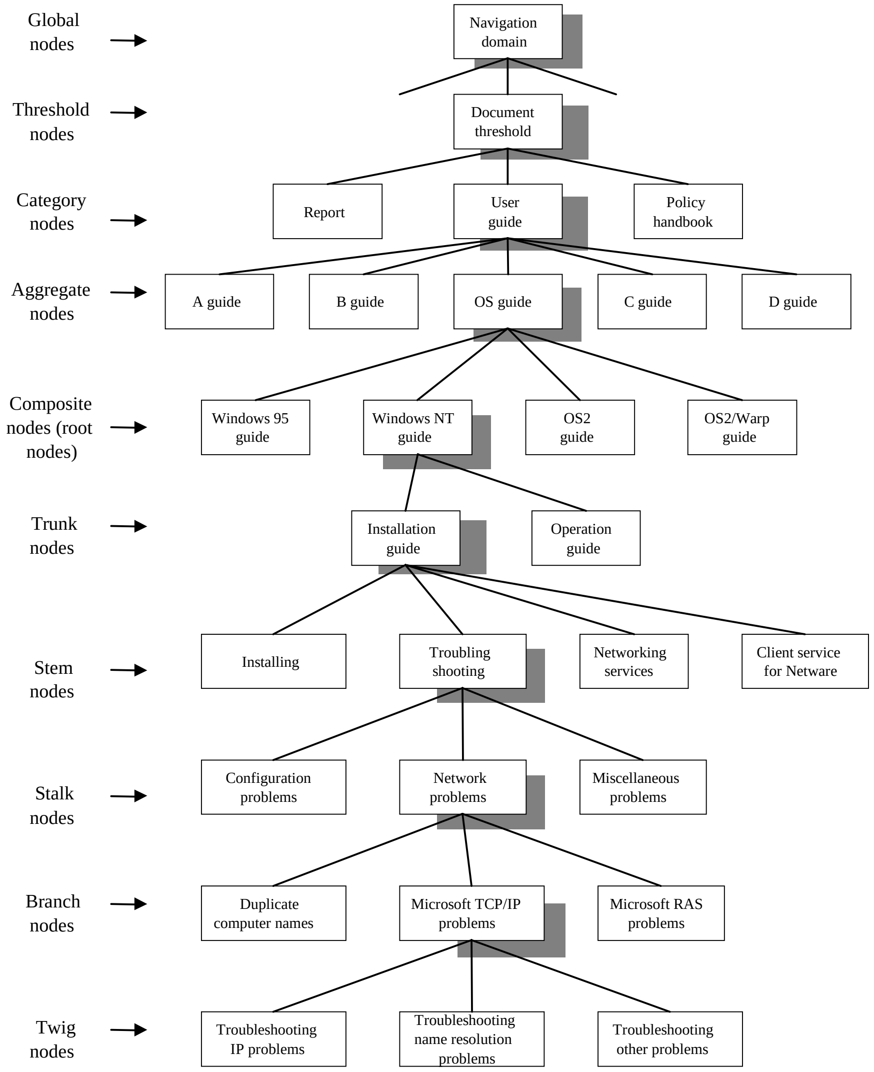
**Figure 2. Basic Elements of Metainformation Structure**

### *4.3.1 Metainformation Structure*

The metainformation structure uses a set of node types, as shown in Figure 2, to organize information. The actual “depth” of the *hierarchy-like* network depends on the structure of information. Note that in designing a hypermedia, the designer should limit “the fragmentation characteristic of hypertext” (Marshall and Irish 1989, p. 22) to increase the level of coherence. For example, dividing a document into individual words would not be meaningful.

The global node represents a navigation domain, and threshold node is a navigation threshold identified in the navigation requirements analysis. The network can grow beyond leaf node but may give the impression of fragmentation. In that case, other names may be attached to the new nodes. A very deep structure may, however, cause readers to become easily disoriented (Botafogo, Rivlin and Shnerderman 1992).

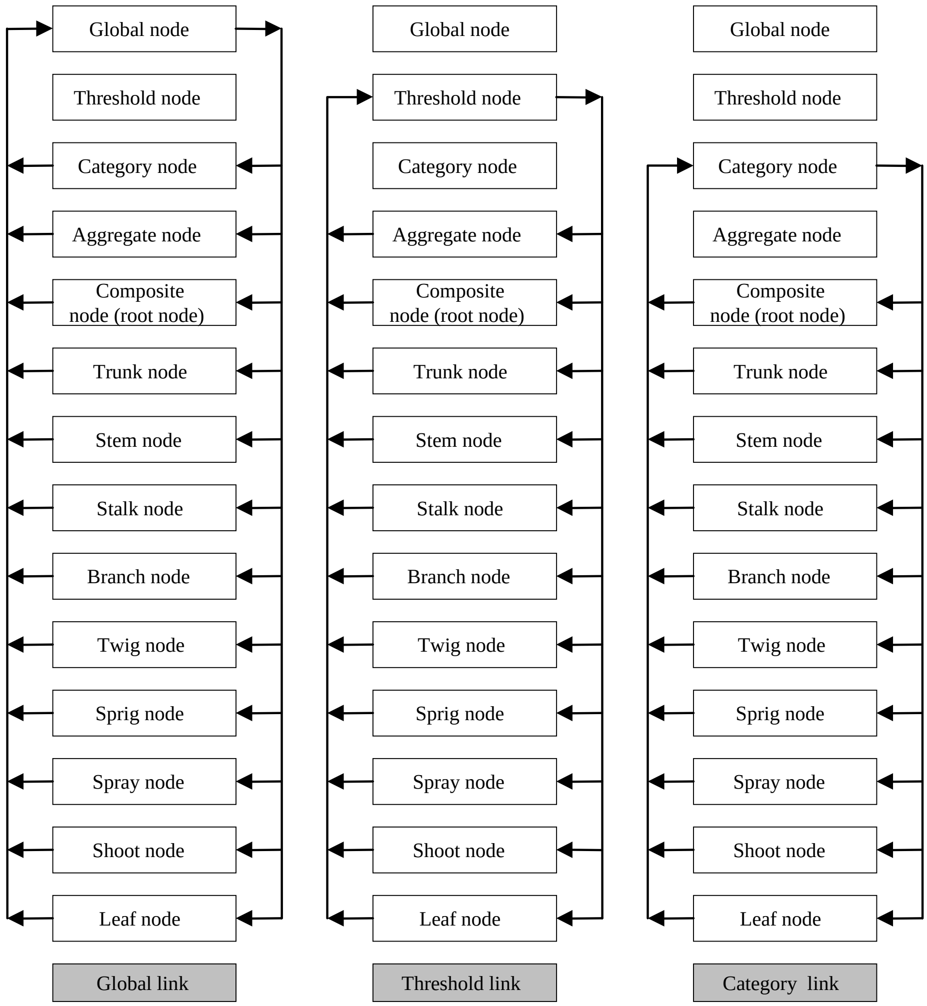
An example a metainformation structure is shown in Figure 3. It has been developed for a *document* navigation threshold. The “decomposition” continues until a reasonable structure is found while avoiding the impression of fragmentation. This requires the designer’s judgment and discretion of the subject. When information decomposes, each node should be positioned based on the spatial and temporal inclusions, if required. In other words, each node should have some semantic relation with adjacent or dangling nodes. The OS guide cannot be dangled from the Report node. At first glance, Figure 3 looks like a hierarchy, but it is not. It is a network of nodes and links. If necessary, a node can be combined into another or vice versa to avoid the fragmentation or achieve more meaningful nodes. This gives leeway to the designer and emphasizes iteration of the design process.




The expanding nodes are shaded. Each node or a group of nodes are named individually. The tree extends to twigs but it can be expanded to leaf nodes or beyond, depending on the application domain.

**Figure 3. The Basic Structure of Metainformation**

The next step is to define links among nodes. There are two link classes for a metainformation structure: hierarchical-link and referential-link. The former is divided into jump-link and step-link subclasses. The *jump-link* subclass defines "jumping" between non-immediate nodes. It may have up to 12 instance links: global through splay. Some instances of the jump-link subclass are shown in Figure 4. The upward jump links can be implemented using a table of contents. The s*tep-link* subclass is used for "step-by-step" traversal between immediate nodes. It has two instances of links: *specialization* and *inclusion*. The referential-link class in a metainformation structure has one instance called a *parallel* link that connects siblings of the same "level" in a metainformation structure. The parallel link and the instances of the step-link subclass are shown in Figure 5. The parallel link indicates spatial or temporal inclusions (semantic relationships). The two link classes provide useful mechanisms for identifying links. Although all of the nodes are linked bi-directionally, it may not always be the case. A metainformation structure provides a template for "possible" links among nodes. Implementing links depends on semantic relationships among nodes.



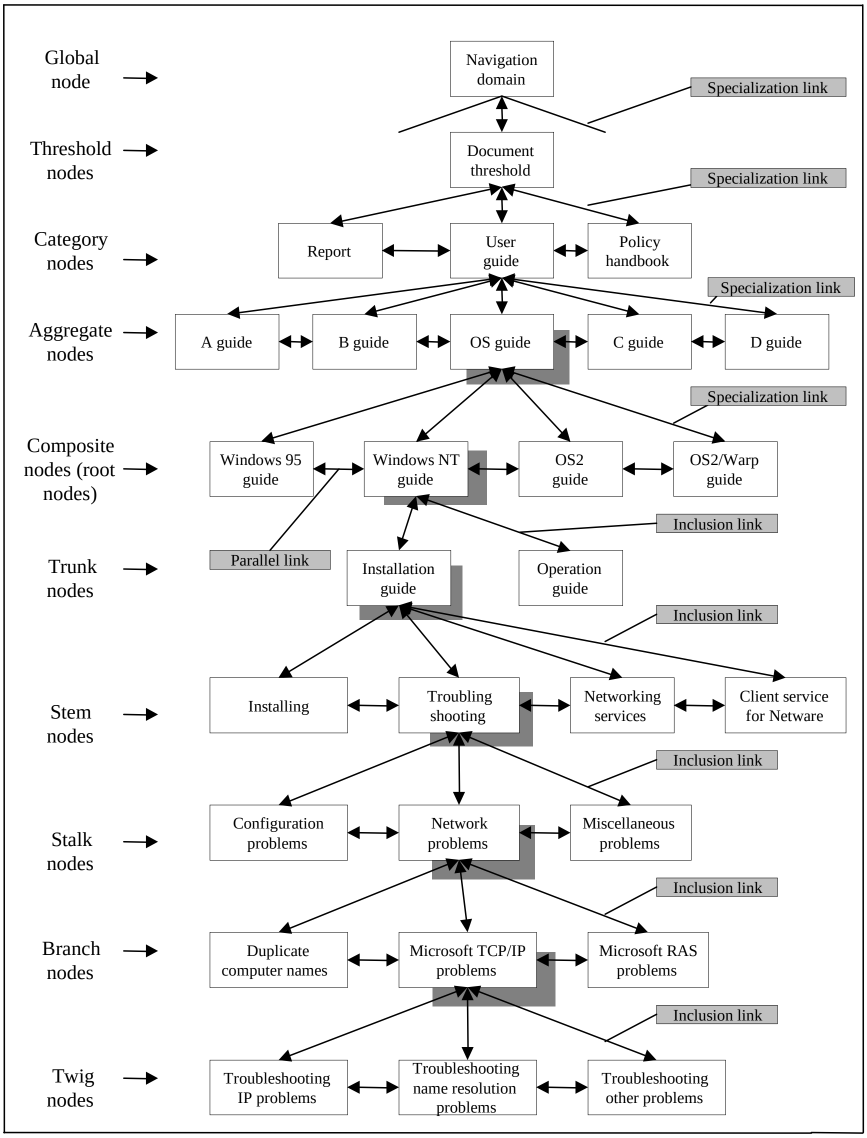

Jump-links are used to jump between non-immediate nodes. This figure shows only three instance links of the jump-link subclass (global, threshold, and category link). In this methodology, a total of 12 links are possible for the jump-link subclass, depending on a metainformation structure.

**Figure 4. Instance Links of the Subclass Jump-link**

### *4.3.2 Information Structure*

As mentioned earlier, the information structure organizes each node of a metainformation structure using classes, subclasses, instances, attributes, and attribute values. Figure 6 shows an example information structure for a graduate handbook. The first step in developing an information structure is to identify *classes* from the contents of a node. The next step is to identify *conditional attributes* for each class. This step is called subclassing. After the subclassing, other standard attributes—*intrinsic attributes* and *associative attributes*—are identified. The intrinsic attributes are combined into a special attribute called "Overview." The links are established in the last step. All identified classes, subclasses, instances, and attributes are again organized by a *hierarchy-like* network structure.

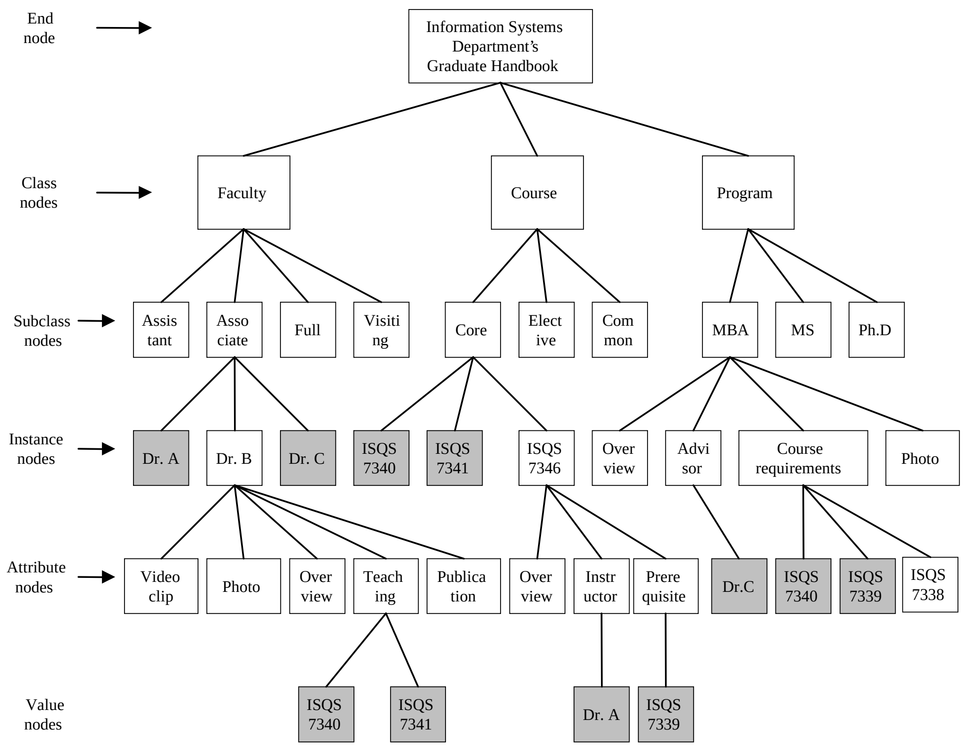
Like the metainformation structure, the information structure also has two link classes: hierarchical-link and referential-link. The custom attributes, special attribute Overview and Search are regarded as objects. However, standard attributes have their corresponding values. In Figure 6, Dr. B has two associative attributes (i.e., *teaching* and *publication*). The values of the




Specialization links indicate links for the structural relationships “is a threshold of,” “is composed of,” “is a category of,” and “is a.” Inclusion links show “is part of” relationships. Parallel links—instances of the referential link class—show spatial or temporal relationships.

**Figure 5. Instances of the Step Link Subclass Showing Two Instances**

attributes provide cross links—the instance of referential link class in an information structure. They are shown by shaded boxes. For example, the link between Dr. A under *associate* is cross-linked to Dr. A under *instructor* attribute of the instance ISQS 7346. In this way, Dr. A can be reached either from *faculty* or *course*. The description of Dr. A is defined only once, but has an incoming link from the shaded box under *instructor*. Other instance links are shown in Figure 7. Unlike the metainformation structure, the information structure has no jump-link subclass. When establishing sibling links—an instance of referential link class—nodes should also be arranged semantically. There is no sibling link between attribute values nor between attributes. Nonetheless, sibling links may exist between classes, between subclasses, and between instances. To avoid cluttering, Figure 7 does not show the sibling links.




Three classes and their subclasses are shown in this example. Some attributes and their values are also indicated. Note that certain attributes may currently have a single value. However, the possible number of values for an attribute may be one or more (e.g., Instructor attribute of ISQS 7346 instance). The shaded instances are occurring within this node or across other nodes.

**Figure 6. An Information Structure**

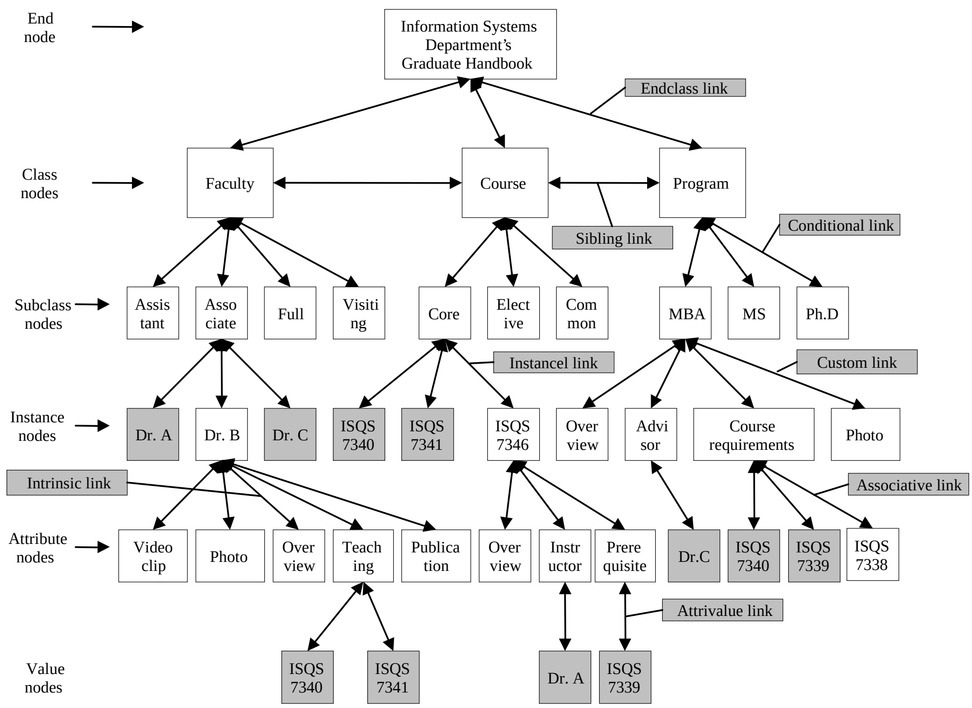
## 4.4 File Structure Design and Virtual Structure Design

A file structure is a storage structure of a node/link network (Halasz 1988). The Dexter Model (Halasz and Schwartz 1994) calls it the storage layer. In previous hypermedia systems, information has been stored using various units such as NoteCards (Halasz 1988) and Components (Halasz and Schwartz 1994). However, the storage unit of the intranet is a file. Thus, the file structure design should be considered in terms of size and content. This issue needs more investigation.

The virtual structure design is related to retrieving "reader" views from the network of nodes and links, which is similar to creating views from a database. The problem arises from the fact that the hypermedia network does not reconfigure itself in response to rapidly changing information (Halasz 1988). Although this problem is partly solved by search engines of the Web, it still remains a problem in terms of the number of "hits."

## 4.5 User Interface Design

Many hypermedia design methods overlook user interface design. Rossi et al. (1995) claim that a formal design model should be used prior to implementation of user interfaces for hypermedia. The proposed user interface design method is based on the frame concept (Minsky 1975). The two immediate advantages of frames include (1) a number of different objects may share the same frame and (2) frames provide a natural hierarchy through the subframe structure (Winston 1977). The following description is based on the metainformation structure and the information structure of the previous section.



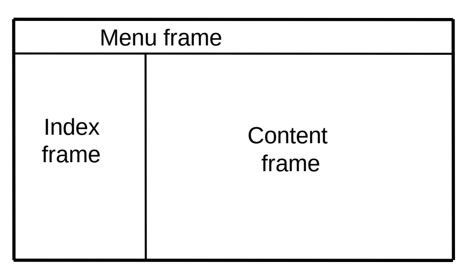

Six instances of the step link subclass are shown. Again, the individual nodes should be arranged on the basis of spatial or temporal inclusion when establishing a sibling link, which is an instance of referential link class. The cross link is not shown but indicated by shading. If more than one subclass, then there will be more than one conditional link.

**Figure 7. Instances of the Step Link Subclass and Referential Link Class**

### *4.5.2 Metainformation Structure Frame Systems*

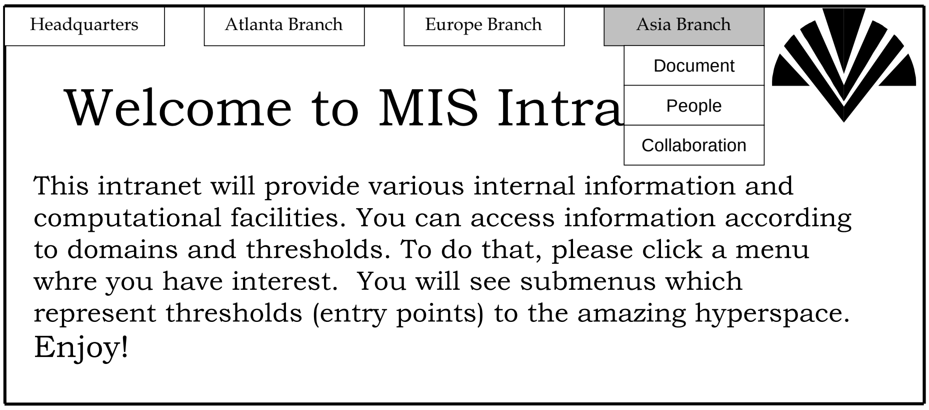
User interface is a collection of metainformation structure frame systems. A metainformation structure frame system consists of menu, index, and content subframe systems. Figure 8 shows a metainformation structure frame system. The menu frame system displays menu options that are global node names such as *headquarters* and *Europe branch*.

Each menu has submenus containing threshold node names for a specific navigation domain. The index frame system reflects the structure of a metainformation structure. When an intranet starts, no distinction between index and content frame systems exists. The combined frame will display an overall introduction such as a welcome message. Figure 9 shows an initial screen for an intranet. An intranet may have many navigation domains. Assume that the navigation domains are Headquarters, Atlanta, Europe, and Asia branches: they all are global nodes. These navigation domains are *menu* options as seen in Figure 9. A navigation domain may have many thresholds such as Document, People, and Collaboration: they all are threshold nodes. They are also *submenus* for the navigation domain of, say, Asia branch. Under the Document threshold in Figure 5, there are several document categories such as user guide: they all are category nodes and become submenus, and so on. An example is shown in Figure 10 when the Document option of the Asia branch in Figure 9 is selected. The index frame shows various submenus. As we can see, menu frame system contains global and threshold node




The metainformation structure is divided into three subframe systems. The contents of each frame system are implied by the name of each subframe system.

**Figure 8. A Metainformation Structure Frame System**




There is no distinction between index frame and content frame. Menu items and one of their submenus are shown.

**Figure 9. The Initial Screen when an Intranet is First Connected**

names. The index frame system has various index subframes to represent node names other than global and threshold node names. The content frame system also has various content sub-frames depending on the menu item selected. In this way, many subframes of both index and content frame systems can be constructed. In Figure 10, the arrow is a button, and the underlined text is a hotspot. A button is a *stand-alone* instance of a jump-link or step-link subclass, while a hotspot is an *embedded* instance of a step-link or jump-link subclass.

Figure 11 shows a formal metainformation structure frame system. Five media types are specified for the content frame system. An HTML file may contain these five media. Subframes for the media can be defined like the text frame. After defining the metainformation structure frame system, the information structure frame system is defined. It is based on the information structure for each node. The only difference between the metainformation structure and the information structure frame systems is that there is no index frame for the information structure. In constructing an information structure frame system, the instances of the step-link subclass have the focus.

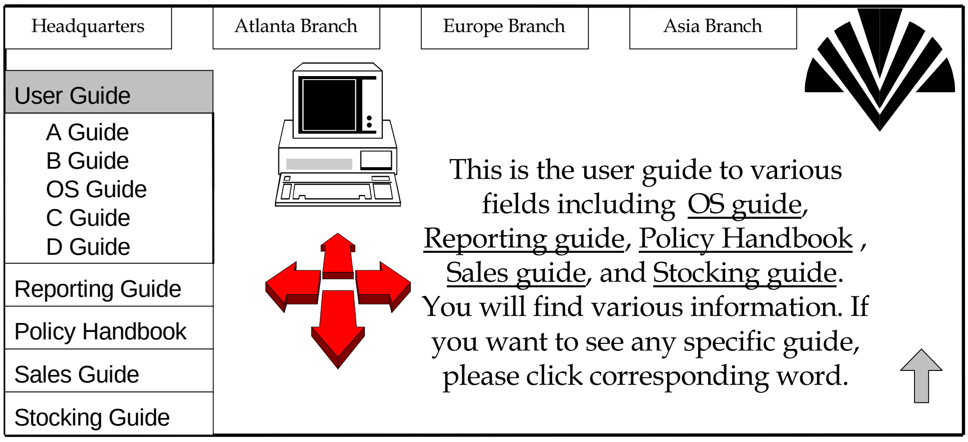
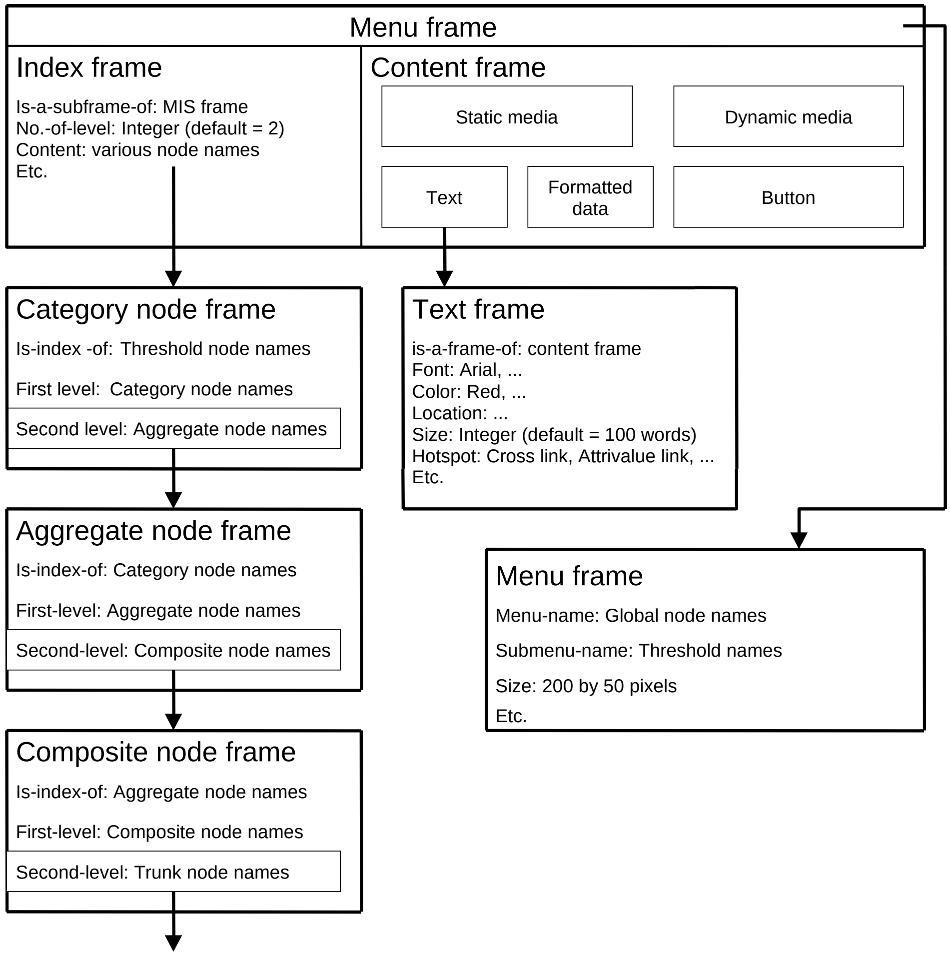



**Figure 10. An Example of an Intranet Second Screen**




Several subframes for the index frame system are shown. In the context frame system, there are five subframes. The menu frame is shown for clarity (not a subframe). A text frame may have slots for hotspots. The static media frame contains images and graphics. Formatted data contains forms or spreadsheet data. The dynamic media frame contains audio, video, or animation.

**Figure 11. A Formal Metainformation Structure Frame System**

## 4.6 Implementation and Testing

The design specifications can be implemented in a straightforward manner by mapping the nodes and links identified in various phases into actual files and links. This section illustrates the prototype the author developed using the IDM to show the utility of the approach.

The application domain for the prototype intranet was the department of Business Computer Information Systems and Quantitative Methods (BCIS/QM) of Hofstra University, Hempstead, New York. The application domain is so small that the BCIS/QM itself is the navigation domain. For this domain, several navigation thresholds were identified (Table 1).

**Table 1. The Results of the Navigation Requirements Analysis**

| | |
|---|---|
| Application Domain | BCIS/QM Department |
| Navigation Domain | BCIS/QM Department |
| Navigation Thresholds | People, Documents, Classes, Communications, Search |

Based on the results, several interviews were conducted to identify user information needs. Information about faculty, staff, and students was identified for the People threshold; faculty handbook and policy series for the Documents threshold; student handbook for the Classes threshold; and various departmental memo formats and opinion submission forms for the Communications threshold. Several other node types were identified for each threshold except for the Search threshold (this can be regarded as an attribute of the global node). These nodes were arranged into a metainformation structure (Figure 12).

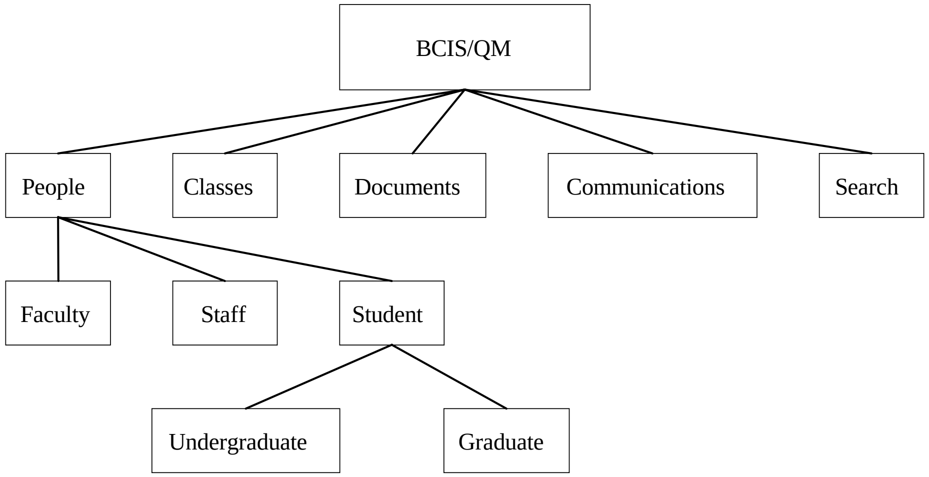


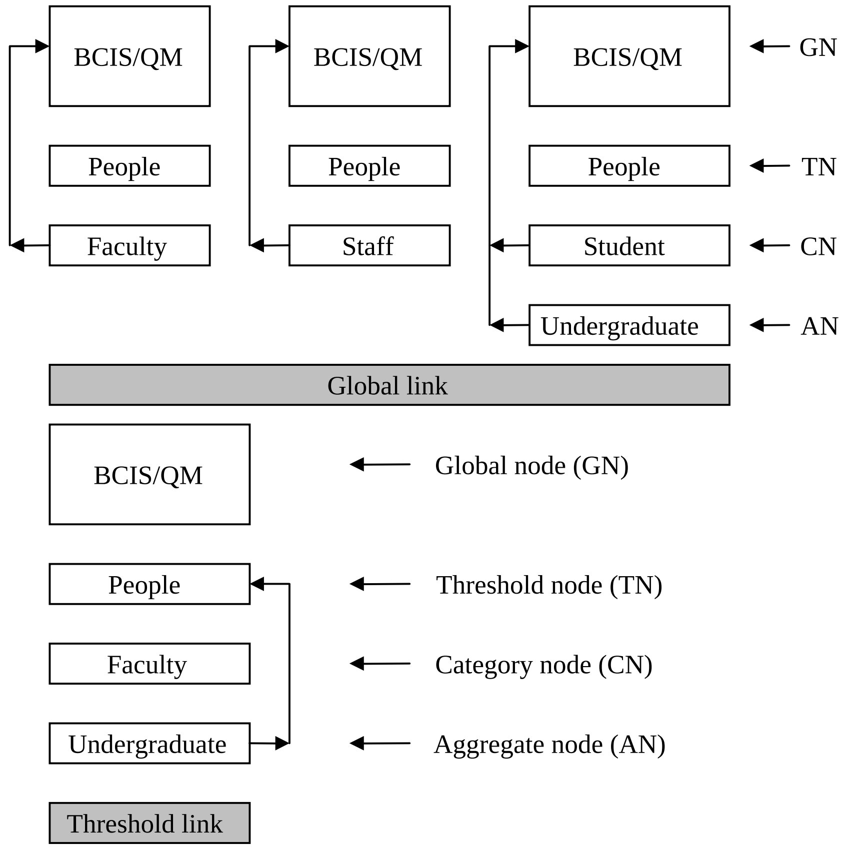

**Figure 12. Part of the Metainformation Structure Showing the Decomposition of the People Navigation Threshold**

The instances of jump-link subclass are shown in Figure 13. As stated earlier, there are up to 12 instances of the subclass. The four levels of nodes in Figure 12 lead to two instances. As can be seen in Figure 13, the instances of the jump-link subclass provide access between non-immediate nodes. Instances of step-link subclass and referential link class are shown in Figure 14. It shows the access structure between immediate nodes.

The next step is to develop information structures for each node of the metainformation structure. An information structure for the Faculty node is shown in Figure 15 with various link instances. Each faculty group is classified by the conditional attribute Rank. The shaded boxes represent cross links to the pages that contain information about BCIS10, Operations Management, and IT in Business respectively.

The next step is to convert the metainformation structure and information structure into the user-interface specification using the concept of the frame. Figure 16 shows an example of a content frame of the user interface.

The last step is to map the nodes and links onto the actual objects. The design specification was implemented using Microsoft FrontPage 97 with an image editor and HTML 3.0. The initial screen for the prototype is shown in Figure 17. It shows navigation thresholds as menu titles. The prototype is a small application, and hence there are no pull-down menu options. Large applications may include these options. The illustration shows only how the IDM can be utilized in developing an intranet. Before the prototype was deployed, it was tested against two browsers in the Hofstra LAN environment. It did not reveal any problems. The prototype was deployed using Microsoft Personal Web Server. It still runs fine.




There are two instances of the jump-link subclass: global link and threshold link. It shows "jumping" links between non-immediate nodes of the People navigation threshold of the metainformation structure in Figure 12.

**Figure 13. Instances of Jump-link Subclass**

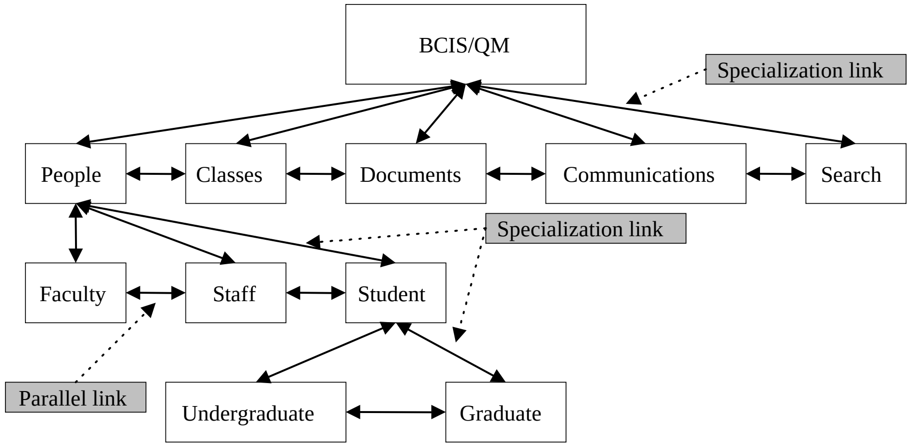



**Figure 14. Instances of Step-link Subclass and Referential Link Class of a Metainformation Structure**

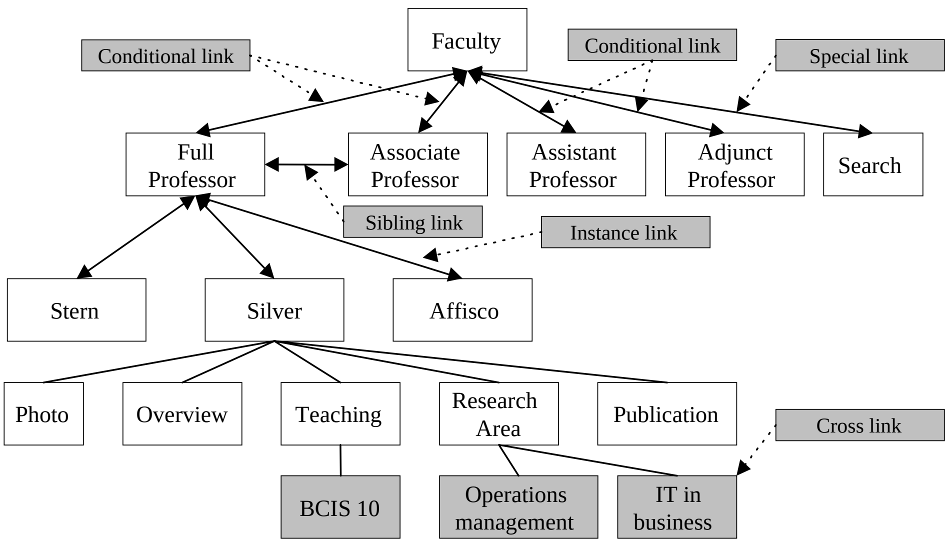



**Figure 15. The Information Structure for Faculty Node with Various Link Instances**

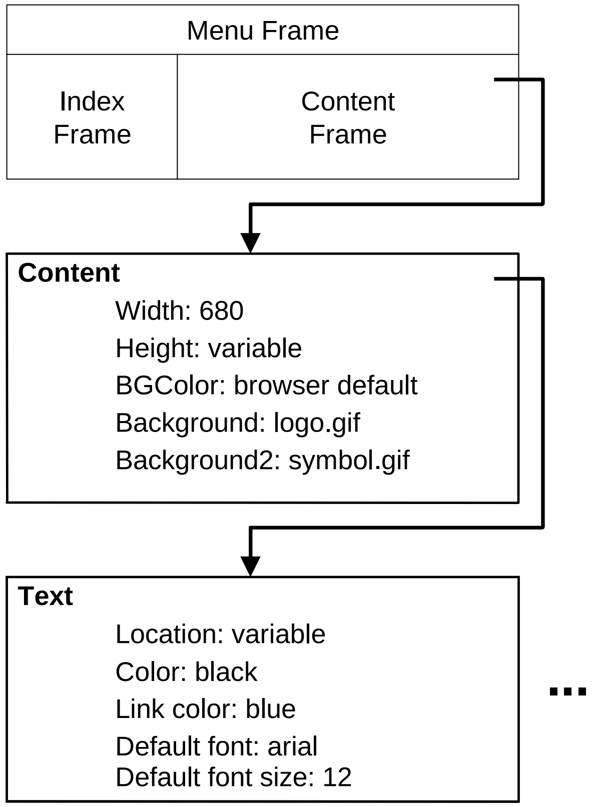
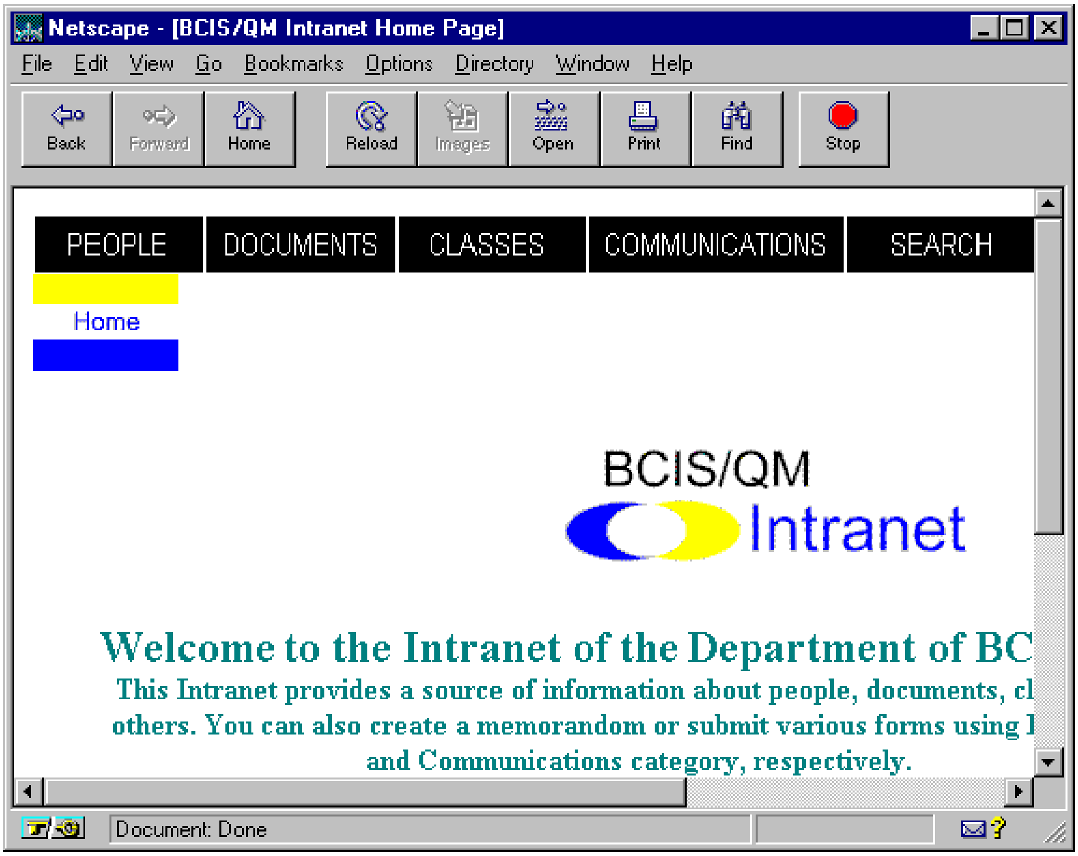



**Figure 16. An Example of a Content Frame**

## 5. CONCLUSIONS AND OPPORTUNITIES FOR FUTURE RESEARCH

In summary, this article has presented an intranet design methodology based on the concepts of the metainformation and information structures. The underpinning paradigm is "divide and conquer" utilizing structural classification and attribution, which is reflected in the navigation design. To provide a formal user-interface design, the frame concept has been employed. The IDM differs from the existing hypermedia design methods in two respects. First, it can be used for any size of hypermedia application and, second, it is flexible, extensible, and easy to apply. The benefits include (1) simplified design method by concentration on the global and local structures of information; (2) ability to incorporate security issues into the navigation design; (3) easy-to-modify design; (4) applicability to other Web-based application development; (5) ability to reduce user disorientation and cognitive overhead; (6) ability to increase the global and local coherence of users.

Currently, many advanced technologies are incorporated into intranets, including dynamic page generation, database access, and integration with legacy systems. The IDM is basically intended for static information. Although it gives insight into the areas, it does not cover the dynamic and function-oriented applications. Moreover, as discussed earlier, the IDM lacks details regarding information requirements analysis, file structure design, and virtual structure design. These areas should also be further investigated.




The figure shows menu items (PEOPLE, DOCUMENTS, CLASSES, COMMUNICATIONS, and SEARCH) that are navigation threshold names.

**Figure 17. The Initial Screen for the Prototype**

## *References*


Bieber, M., and Isakowitz, T. "Designing Hypermedia Applications" *Communications of the ACM* (38:8), 1995, pp. 26-29.

Bieber, M., and Vitali, F. "Toward Support for Hypermedia on the World Wide Web." *IEEE Computer*, 1997, pp. 62-70.

Botafogo, R.; Rivlin, E.; and Shnerderman, B. "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics," *ACM Transactions on Office Information Systems* (2:10), 1992, pp. 142-180.

Bush, V. "As We May Think," *Atlantic Monthly* (176:1), 1945, pp. 101-108.

Byrd, T. A.; Cossick, K. L.; and Zmud, R. W. "A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques," *MIS Quarterly* (16:1), 1992, pp. 117-138.

Challa, C. D., and Redmond, R. T. "Is it a Lot of Hype? Hypermedia Approach to Document Processing," *Journal of Systems Management* (47:3), 1996, pp. 12-21.

Davis, G. B., and Olson, M. H. *Management Information Systems: Conceptual Foundations, Structure, and Development*, 2nd ed., McGraw-Hill, New York, 1985.

Deibel, S., and Greenes, R. "An Infrastructure for the Development of Healthcare Information Systems from Distributed Components," *Journal of the American Society for Information Science* (46:10), 1995, pp. 765-771.

Diaz, A., and Isakowitz, T. "RMCase: Computer-Aided Support for Hypermedia Design and Development," *Proceedings of the International Workshop on Hypermedia Designs*, 1995, pp. 3-15.

Engelbart, D. C. "Toward Augmenting the Human Intellect and Boosting Our Collective IQ," *Communications of the ACM* (38:8), 1995, pp. 30-33.

Engelbart, D. C. "Authorship Provisions in Augment," *Proceedings of FJCC*, 1968, pp. 395-410.

Garzotto, F. "Welcome Address," *Proceedings of the International Workshop on Hypermedia Design (IWHD'95)*, 1995.

Garzotto, F.; Paolini, P.; and Schwabe, D. "HDM—A Model-based Approach to Hypertext Application Design," *ACM Transactions on Information Systems* (11:1), 1993, pp. 1-26.

Halasz, F. "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems," *Communications of the ACM* (31:7), 1988, pp. 836-855.

Halasz, F., and Schwartz, M. "The Dexter Hypertext Reference Model," *Communications of the ACM* (37:2), 1994, pp. 30-39.

Isakowitz, T.; Stohr, E. A.; and Balasubramanian, P. "RMM: A Methodology for Structured Hypermedia Design," *Communications of the ACM* (38:8), 1995, pp. 34-44.

Jackson, T. *The Next Battleground*, Houghton-Mifflin, New York, 1993.

Marshall, C. C., and Irish, P. M. "Guided Tours and Online Presentations: How Authors Make Existing Hypertext Intelligible for Readers," *Proceedings of Hypertext '89*, 1989, pp. 15-26.

McCune, J. C. "The Intranet: Beyond e-mail," *Management Review* (85:11), 1996, pp. 23-27.

Minsky, M. "A Framework for Representing Knowledge," in P. Winston (ed.), *The Psychology of Computer Vision*, McGraw-Hill, New York, 1975, pp. 211-277.

Nanard, J., and Nanard, M. "Hypertext Design Environment and the Hypertext Design Process," *Communications of the ACM* (38:8), 1995, pp. 49-56.

Nelson, T. H. "A File Structure for the Complex, the Changing and Indeterminate," *Proceedings of the ACM Twentieth National Conference*, 1965.

Perry, T. S. "Hypermedia: Finally Here," *IEEE Spectrum*, 1987, pp. 38-39.

Porter, M. E. "The Competitive Advantage of Nations," *Harvard Business Review*, 1990, pp. 73-93.

Rossi, G.; Schwabe, D.; Lucena, C. J. P.; and Cowan, D. D. "An Object-Oriented Model for Designing the Human-Computer Interface of Hypermedia Applications," *Proceedings of the International Workshop on Hypermedia Design*, 1995, pp. 123-143.

Schwabe, D., and Rossi, G. "The Object-Oriented Hypermedia Design Model," *Communications of the ACM* (38:8), 1995a, pp. 45-46.

Schwabe, D., and Rossi, G. "Building Hypermedia Applications as Navigational Views of Information Models," *Proceedings of the Hawaii International Conference on Systems Sciences*, 1995b.

Simon, H. "The Architecture of Complexity," *Proceedings of the American Philosophical Society* (106:6), 1962, pp. 467-482.

Welch, J. "A Master Class in Radical Change," *Fortune* (28), 1993, pp. 82-83.

Waldorp, M. M. *Complexity: The Emerging Science at the Edge of Chaos*, Simon and Schuster, New York, 1992.

Winston, P. H. *Artificial Intelligence*, Addison-Wesley, Reading, MA, 1977, pp. 181-187.