DISTRIBUTED DESIGN OF HYPERMEDIA APPLICATIONS

Tomás Isakowitz

Stern #IS-95-23

DISTRIBUTED DESIGN OF HYPERMEDIA APPLICATIONS

Tomás Isakowitz
Information Systems Department
Leonard N. Stern School of Business
New York University
New York, NY 10012-0266
E-mail: tomas@stern.nyu.edu
URL http://is-2.stern.nyu.edu/tisakowi

Working Paper Series
STERN IS-95-23

Distributed Design of Hypermedia Applications

Abstract

Hypermedia technology is experiencing a rapid growth due, in large part, to the WWW. Many hypermedia applications, especially those on the WWW have a distributed design besides being physically spread among many servers. A distributed design is a design that varies, albeit slightly, from instance to instance. However, such design variances can lead to undesirable inconsistencies that can render a hypermedia application useless. This paper explores this problem and presents a solution based on a methodological approach to hypermedia design and construction. The methods are illustrated via a sample application.

1 Introduction

The WWW has brought attention to a new kind of applications that exhibit a sort of heterogeneity rarely encountered in other environments. The loosely-coupled characteristics of the WWW platform consents heterogeneity in application design, that allows similar information to be portrayed in dissimilar ways. This is the problem of *hypermedia distributed design.*

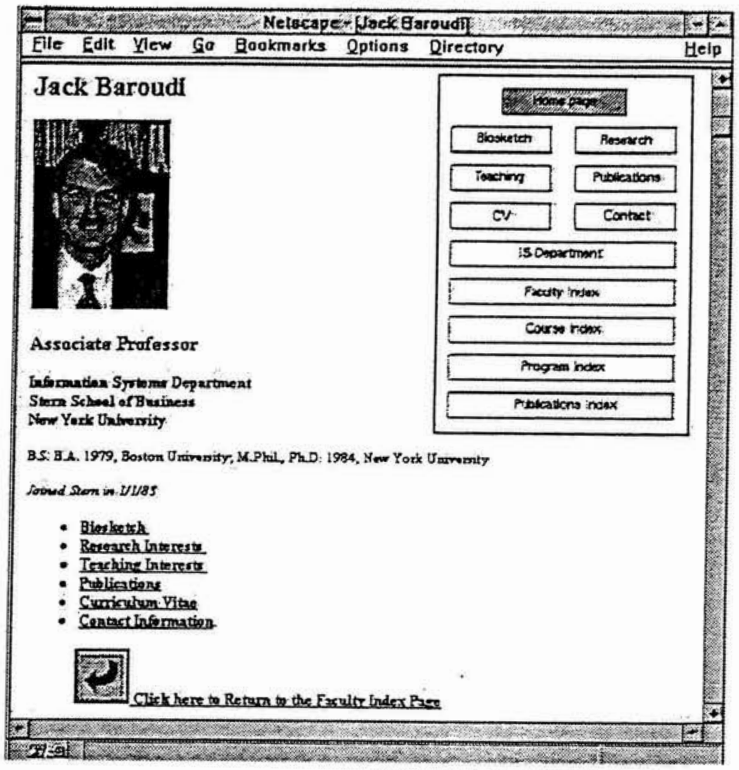
In this paper I study how to instill and maintain a consistent design within a distributed hypermedia application. I provide terminology and a framework that enable a clear articulation of the issues involved, and I present a general solution that brings forward conceptual and practical benefits.

Take, as an example, personal homepages of academicians. Although many embody similar kinds of information (e.g., a biographical sketch, teaching interests, research interests, publications and personal interests), these personal homepages exhibit great diversity, not only in terms of data contents, e.g., diverse biographical sketches and research interests, but also in terms of their design. Some homepages may include pictures and a few links, others may have no links at all and be mostly textual, while a third group consists of pages that are, themselves, a complex web.

Consider the example shown in figure 1. Both homepages contain very similar information, yet it is displayed in different ways. Navigation aids (e.g., links to "research") are also displayed differently (with a map in (a), and with text in (b)). Moreover, (b) contains some additional links to the "Wharton School", the "University of Pennsylvania", etc. In sum, with regards to the kind of information they contain, (a) and (b) are somewhat similar, but not identical. They are however, quite different in the way they present this information. Users should, nonetheless, still feel that they are within the bounds of the IS Department site as they inspect both pages. They also expect to be able to follow similar navigational cues throughout the entire application.

For another example, consider a firm that sells its products though the Web. Unless its web site exhibits a consistent design, users can become disoriented and confused as they attempt to shop electronically. As a result, customers may make undesirable errors, will feel frustrated, and, ultimately,

(A)



Jack Baroudi

Associate Professor

Information Systems Department
Stern School of Business
New York University

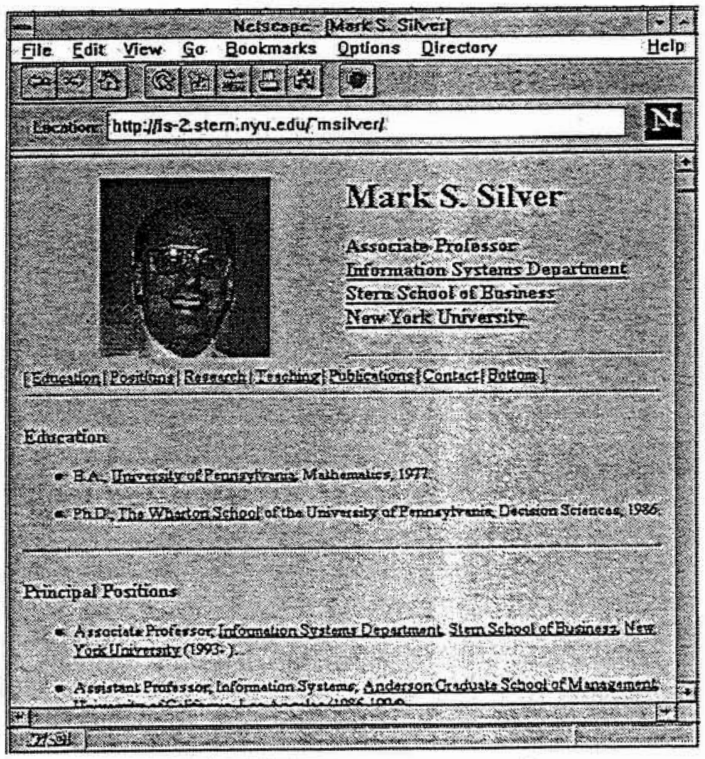
B.S. B.A. 1979, Boston University; M.Phil., Ph.D. 1984, New York University

Joined Stern in 1/1/85

- Biosketch
- Research Interests
- Teaching Interests
- Publications
- Curriculum Vitae
- Contact Information

Click here to Return to the Faculty Index Page

(B)


Netscape - [Mark S. Silver]
File Edit View Go Bookmarks Options Directory Help
Location: http://is-2.stern.nyu.edu/~msilver/

Mark S. Silver

Associate Professor
Information Systems Department
Stern School of Business
New York University

[Education | Positions | Research | Teaching | Publications | Contact | Bottom]

Education

- B.A., University of Pennsylvania, Mathematics, 1977.
- Ph.D., The Wharton School of the University of Pennsylvania, Decision Sciences, 1986.

Principal Positions

- Associate Professor, Information Systems Department, Stern School of Business, New York University (1993-).
- Assistant Professor, Information Systems, Anderson Graduate School of Management,

Figure 1: An example of *distributed design*. Two personal homepages displaying similar information. An example of semi-structured objects, which are somewhat similar, yet heterogeneous. However, and most importantly, both designs are compatible.

will abandon use of the system.

The two examples I have given are hardly an exception. They cover an important cross-section of Web sites: WWW presence and electronic commerce. Clearly, many other applications, e.g., financial services [11] [13], decision support [14], Executive Information Systems [4], searches, product information for customer support, require a degree of consistency that is hard enforce on the WWW.

1.1 Organizational Arguments for Design Consistency

There are organizational and technical motivations for solving the problem of distributed application design. Form an organizational perspective, it is important that an application to be used by the public exhibit a common "look and feel" [16] so that its users can (1) readily identify the application with the organization, and (2) easily learn to use the application.

Unless the organization keeps a tight, centralized control of its web site (in clear opposition to the basic philosophy of the WWW), such consistency is hard to enforce within existing WWW software development environments.

1.2 Software Engineering Arguments for Design Consistency

Maintenance of WWW sites is labor-intensive and error-prone, mainly because links are hard-coded. Deleting, renaming or moving an HTML file for example, can result in dangling links because links may still point to the old URL (source for the so common *"Not Found"* error messages). A solution, popular due to its simplicity, but annoying for users, is to place in the old URL a message with a link to the new one (the *" we have moved"* phenomenon).

Incorporating new information into a hypermedia application can also be problematic. It requires inter-linking new information items into the existing application by adding appropriate links (a) from the item to existing objects, and (b) from existing objects to new items. The first kind of links, although potentially labor-intensive, fall within the responsibility of the authors of the new additions. Thus, they are somewhat manageable. The (b) kind of links are more problematic, as it is not clear who is responsible for making the new additions, and how to do carry them out. Authors of older information items could certainly not have anticipated all possible links to (yet inexistent) documents. On the other had, authors of new information items may neither have the authority nor the knowledge to add links to older information items.

In sum, maintenance of distributed hypermedia applications is a complex and labor intensive task,for which an automated or semi-automated maintenance support system is required. A distributed design framework, as the one I present here, is quintessential to achieve the required levels of support for application maintenance because it maps out in sufficient detail the relevant aspects of the design of applications.

1.3 A Sample Application

I will use, as a sample application to illustrate the concepts of this paper, the ISWEB application [1], which is the WWW site for the Information Systems Department at the Stern School of Business. The application contains information about faculty, courses, research and other academic activities.

Figure 2 shows three screen-shots from ISWEB. In it, the home-page of a faculty member, Jack Baroudi, was reached through the index of "all people" in the IS Department. ISWEB was developed using a uniform design, but envisioning distribution requirements. As Figure 1 showed, it is important to accommodate for the needs of individuals who want to maintain their own web-pages. The challenge is to provide flexibility without compromising the overall look and feel of the application.

2 Distributed Hypermedia Design

Although the tag "look and feel" is widely utilized, its meaning for distributed hypermedia applications needs to be pinpointed more precisely to enable a better analysis. "Look and feel" is thought of in terms of *presentation* aspects, i.e., "look", and *interaction* aspects, i.e., "feel'. I adopt a more inclusive view of the "look and feel" problem by analyzing *design consistency.*

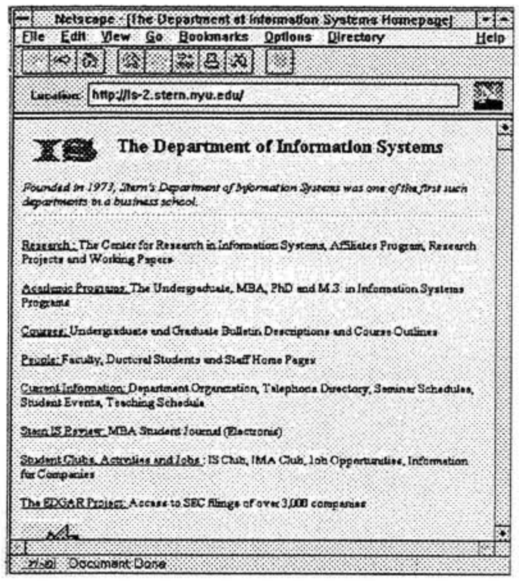
2.1 Design Consistency

Requirements for a consistent design can be divided into:

- *content consistency:* the kind of information presented
- *presentation consistency:* how the information is presented ("look")
- *interaction consistency:* how users interact with the application ("feel").

Content Consistency refers to the homogeneity in the *type of information* present in the application. It is expected, for example, that all faculty

(a)



Netscape - [The Department of Information Systems Homepage]

File Edit View Go Bookmarks Options Directory Help

Location: http://is-2.stern.nyu.edu/

The Department of Information Systems

Founded in 1973, Stern's Department of Information Systems was one of the first such departments in a business school.

Research : The Center for Research in Information Systems, Affiliates Program, Research Projects and Working Papers

Academic Programs: The Undergraduate, MBA, PhD and M.S. in Information Systems Programs

Courses: Undergraduate and Graduate Bulletin Descriptions and Course Outlines

People: Faculty, Doctoral Students and Staff Home Pages

Current Information: Department Organization, Telephone Directory, Seminar Schedules, Student Events, Teaching Schedule

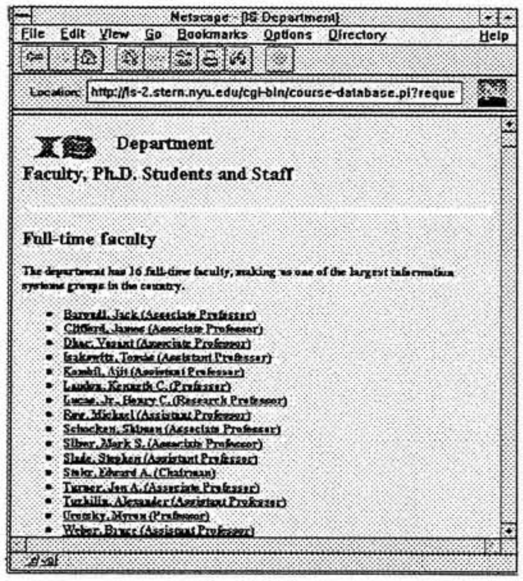
Stern IS Review: MBA Student Journal (Electronic)

Student Clubs, Activities and Jobs : IS Club, IMA Club, Job Opportunities, Information for Companies

The EDGAR Project: Access to SEC filings of over 3,000 companies

Document Done

(b)



Netscape - [IS Department]

File Edit View Go Bookmarks Options Directory Help

Location: http://is-2.stern.nyu.edu/cgi-bin/course-database.pl?reque

Department

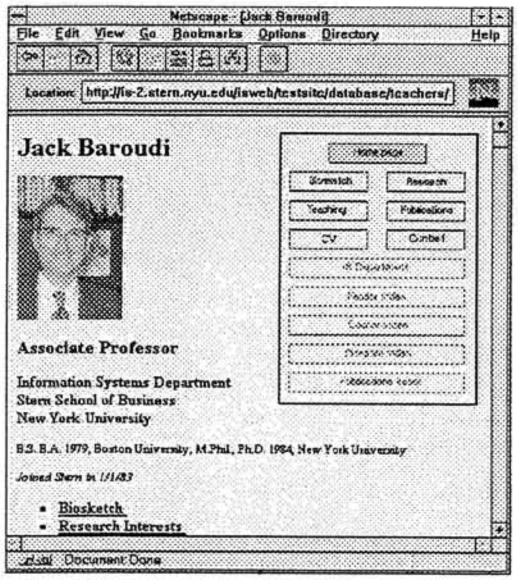
Faculty, Ph.D. Students and Staff

Full-time faculty

The department has 16 full-time faculty, making us one of the largest information systems groups in the country.

- Baroudi, Jack (Associate Professor)
- Clifford, James (Associate Professor)
- Dhar, Vasant (Associate Professor)
- Isakowitz, Tomás (Assistant Professor)
- Kambil, Ajit (Assistant Professor)
- Laudon, Kenneth C. (Professor)
- Lucas, Jr., Henry C. (Research Professor)
- Rice, Michael (Assistant Professor)
- Schocken, Shimon (Associate Professor)
- Silver, Mark S. (Associate Professor)
- Slade, Stephen (Assistant Professor)
- Stohr, Edward A. (Chairman)
- Turner, Jon A. (Associate Professor)
- Tuzhilin, Alexander (Assistant Professor)
- Uretsky, Myron (Professor)
- Weber, Bruce (Assistant Professor)

(c)



Netscape - [Jack Baroudi]

File Edit View Go Bookmarks Options Directory Help

Location: http://is-2.stern.nyu.edu/isweb/testsite/database/teachers/

Jack Baroudi

Associate Professor

Information Systems Department
Stern School of Business
New York University

B.S. B.A. 1979, Boston University, M.Phil., Ph.D. 1984, New York University

Joined Stern in 1/1/83

- Biosketch
- Research Interests

Document Done

Figure 2: The ISWEB application's home-page (a) leads to an index of the people at the IS department (b). By clicking on an individual's name, one reaches the corresponding personal home-page (c).

home-pages contain information about research, teaching, etc. Users will become frustrated if one kind of information, e.g., telephone number, is given in one home-page but absent in another one.

Presentation Consistency requires that similar information be consistently conveyed in similar ways. Besides screen layouts, fonts and colors, this also includes media. For example, pictures are commonly expected on home-pages. Similarly, sound, video and animation should be used consistently.

The use of icons throughout the WWW and the emphasis placed on their consistency is symptomatic of the concerns about presentation consistency. For example, back arrows usually mean "back", up arrows mean "up one level", a home icon, "go to the home-page". Another common practice comprises the use of organizational logos to identify WWW pages within an application. This practice also addresses the *membership requirement,* to be discussed later.

Interaction Consistency refers to the aspects of the interface that determine how users communicate with the application. In hypermedia applications this refers mainly to navigational controls, such as buttons and click-able images.

Of particular importance is *predictability* [7]: when users encounter a new feature, or button, they should be able to induce, from their experience with other parts of the application, the approximate behavior of this new feature.

Design consistency violations burden users, resulting in user-dissatisfaction. If content consistency is violated, for example by including telephone contact information on some people's home-pages, but not on everyone's, users may become frustrated when searching for someone's phone number. They have a strong feeling the information is there, but they just can not get to it.

Presentation inconsistencies can overwhelm users when, for example, they have to adapt to different screen layouts for the same information, e.g., telephone numbers are placed in different screen locations.

Interaction inconsistencies occur, for example, when the same icon is used to represent different kinds of links. It happens at times on the WWW that a "left arrow" means back to the previous page, while in other places it means "back to the root home-page". As a result, users may feel confused, and may catalog the application as unreliable. For other examples of design inconsistencies, I refer the reader to the article by Garzotto *et al.* [7] where Microsoft's "National Gallery" CD-ROM is used to illustrate the evaluation of hypermedia designs.

The navigational ease conferred by the World Wide Web can be problematic. It is important for organizations to be able to delimit their Web sites so that users are aware of site boundaries. Otherwise users may not know when they are leaving one domain and entering a new one. This can be particularly problematic whenever security and privacy are of concern, as in electronic commerce. Imagine the following scenario: vendor B has planted a link to his own web site (B's site) in vendor A's site. As a result, and unnoticed to A, all orders intended for A are actually placed with vendor B.

Although this is an extreme scenario, it serves to illustrate the point. Similar situations can arise in the context of consumer shopping services that facilitate product comparisons across vendors [9].

The need to differentiate itself from other vendors has to be fulfilled within the organization's own Web site by ensuring that every application component is readily identifiable as belonging to the organization. I call this the *membership requirement.* This requirement can be met, for example, through the use of visual cues such as a common look and feel, logos, etc.

2.2 Distributed Design

An application is said to have a *distributed design* when instances of application components of the same type are presented in dissimilar ways, while the overall navigation patterns, which determine how information is accessed, are homogeneous. Thus, there are three aspects to distributed design:

(a) "Instances of the same type". Here *type* refers to the kind of information more in a social sense, e.g., personal information such as *name, photograph, interests,* etc., than in the computer science sense of the word "type", e.g., string, integer, etc..

(b) "Presented in dissimilar ways" means that different instances of the same information type are presented differently. This is a kind of *local heterogeneity.*

(c) "Homogeneous navigation patterns." Despite such local heterogeneity, there is a global sense in which interaction styles are consistent throughout application.

2.3 Semi-Structured Objects

Since the purpose of distributing application components is to enable uncoordinated updates, a satisfactory solution needs to support semi-structured objects. That is, objects whose design consists of a parts that follow central guidelines, and of variable parts, whose maintenance is left to individual persons or departments within an organization. The need for such objects was first identified by Malone [12], who identified them in the context of systems to support management of work-flow in knowledge-related tasks.

The general solution I advocate promotes a simple view of semi-structured objects. Each such object is composed of *central* parts and *distributed* parts. The two components are managed and maintained by different methods, and by different people or departments. The objects themselves are the result of merging the various parts.

3 A General Solution

A generic solution for the distributed design problem is shown in Figure 3. There are three components to the solution. (1) The application is designed and constructed using a semi-structured approach. (2) Individual components of the application are distributed as required to ensure proper ownership. (3) Design updates are broadcasted as needed. For example, an organization may require a new event to be announced, via a link, from every home-page.

The WWW does not include built-in facilities to do manage this process. As a matter of fact, the WWW lacks even simpler facilities for changing a URL address (all links to the prior URL are left dangling).

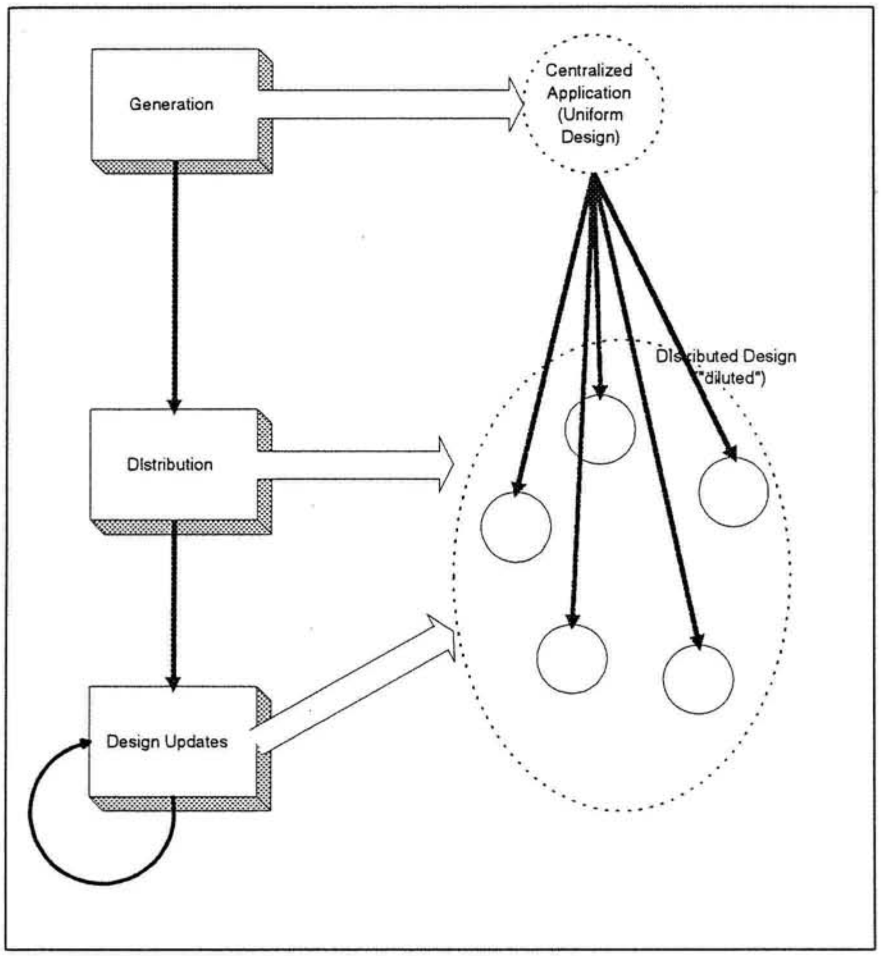

Generation
Centralized Application (Uniform Design)
Distributed Design ("diluted")
Distribution
Design Updates

Figure 3: A General Model for Distributed Hypermedia Design

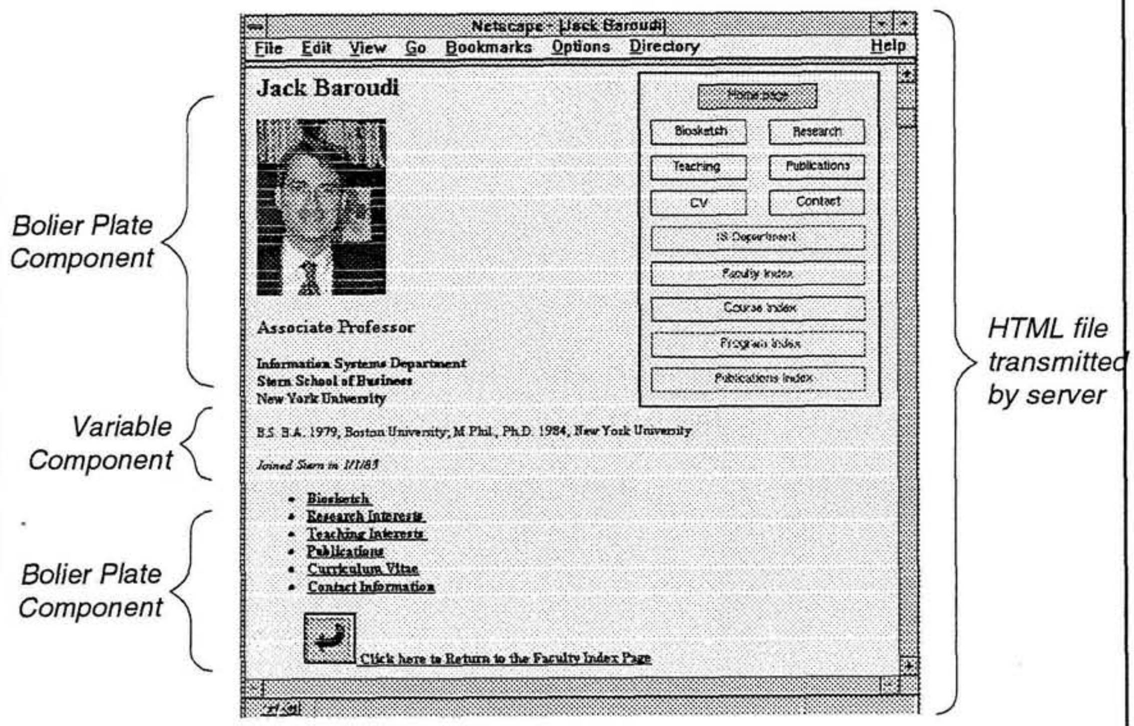
3.1 The "Boiler-Plate" Approach

The framework I propose is based on semi-structured objects. An example may help clarify this view. As shown in Figure 4, personal home-page are separated into two components: (1) a *boiler-plate*, centrally owned, component that contains the elements of the design that are necessary for overall consistency, and (2) a *variable* part, containing specific information, that can be freely changed without incurring in violations of design consistency. In the application the boiler-plate and variable parts are combined into HTML pages. This process can either be batch, i.e., the combination takes place every so-often producing static HTML files, or dynamic, i.e., HTML files are produced as needed by dynamically combining the two components.

It is important that ownership of the boiler-plate component lie within the scope of a central administrator, and that individual users are unable to alter, either deliberately or by mistake, important aspects of an application's design. I propose a system to manage distributed design that has the ability of globally enforcing and maintaining a consistent design without severely limiting the ability of individuals to alter content information, and to incorporate design variances, such as additional links.

3.2 Application Design

Without a good representation of the design of an application, it is difficult to analyze, criticize and suggest improvements to it. An effective management of the software development process requires a tangible representation of an application's design. Thus, a methodological approach to distributed hyper-



Jack Baroudi

Associate Professor

Information Systems Department
Stern School of Business
New York University

B.S. B.A. 1979, Boston University; M.Phil., Ph.D. 1984, New York University

Joined Stern in 1/1/85

- Biosketch
- Research Interests
- Teaching Interests
- Publications
- Curriculum Vitae
- Contact Information

Click here to Return to the Faculty Index Page

Figure 4: The *boiler-plate* architecture separates centrally managed from individually managed components (the *variable* components).

media design (a) promotes communication among the parties involved, and (b) enables more effective management of the software development process.

I use RMM [10], a methodology for the design and construction of hypermedia applications, as the foundations for a system to manage distributed designs. As shown in Figure 5, RMM consists of seven steps, some of which can be conducted in parallel. Figure 5 shows a rather linear progression. Since rapid feedback loops have been identified as crucial to the hypermedia design process [15], a software development environment that supports rapid feedback loops within RMM is under development [6].

Briefly explained, the first three steps of RMM are concerned with design, the remaining four with construction. The first design step results in an E-R diagram. It is similar to traditional E-R techniques, except that the objects of analysis are screens (also called presentation units) instead of information objects or database records. The second step, *Slice Design*, groups the attributes of each entity into groups, called *slices*, to be shown as a unit to the user. During the third step, *Navigational Design*, the designer decides what paths the user will be able to use to access these slices. The output of this step is an RMDM diagram, depicting all navigational paths in the application.

Figure 6 shows an RMD diagram for the ISWEB application. In contrast to an entity-relationship diagram that represents the design of a database, an RMDM diagram describes how users will navigate a hypermedia application. To avoid cluttering, slices are not included in Figure 6. At the top of Figure 6 the grouping mechanism implements a "main menu". Access into the *Faculty* and *Courses* information is provided via guided tours; access into

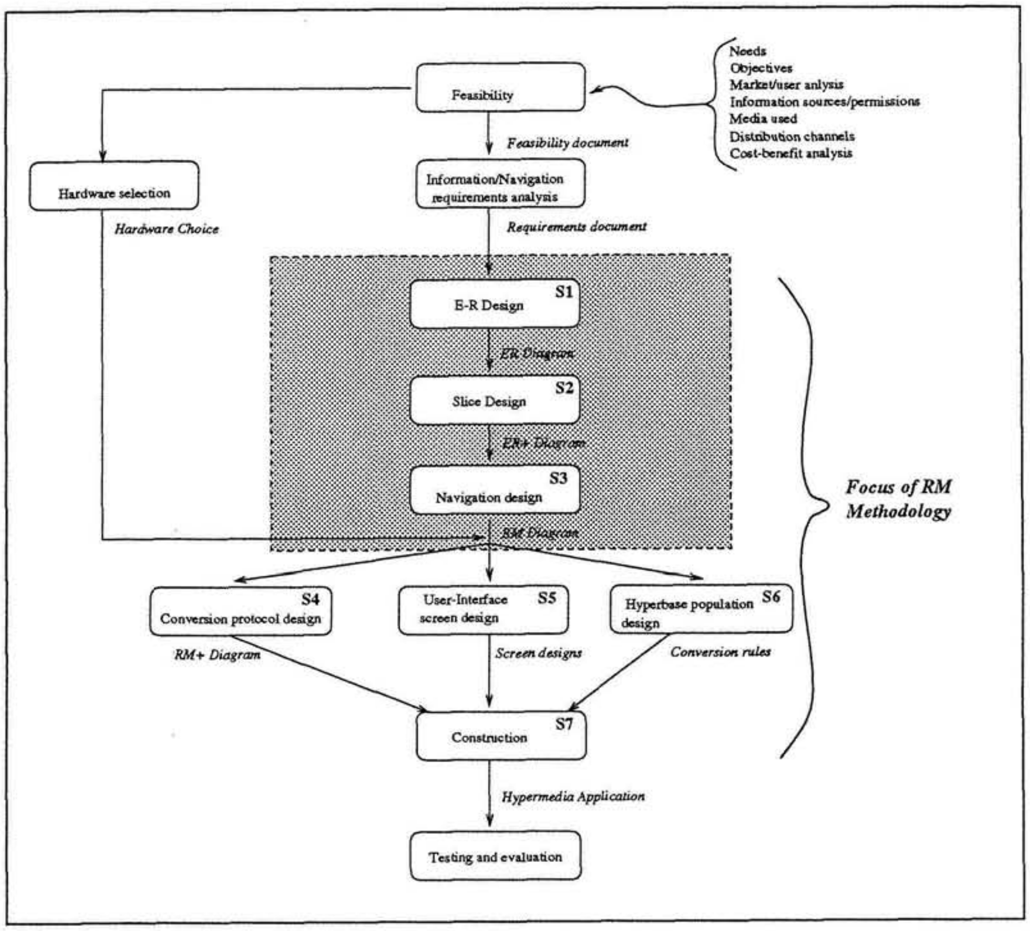


Figure 5: A linear depiction of the RMM methodology. The shaded area represents conceptual design activities. Numerous feedback loops are to be supported in a software development environment.

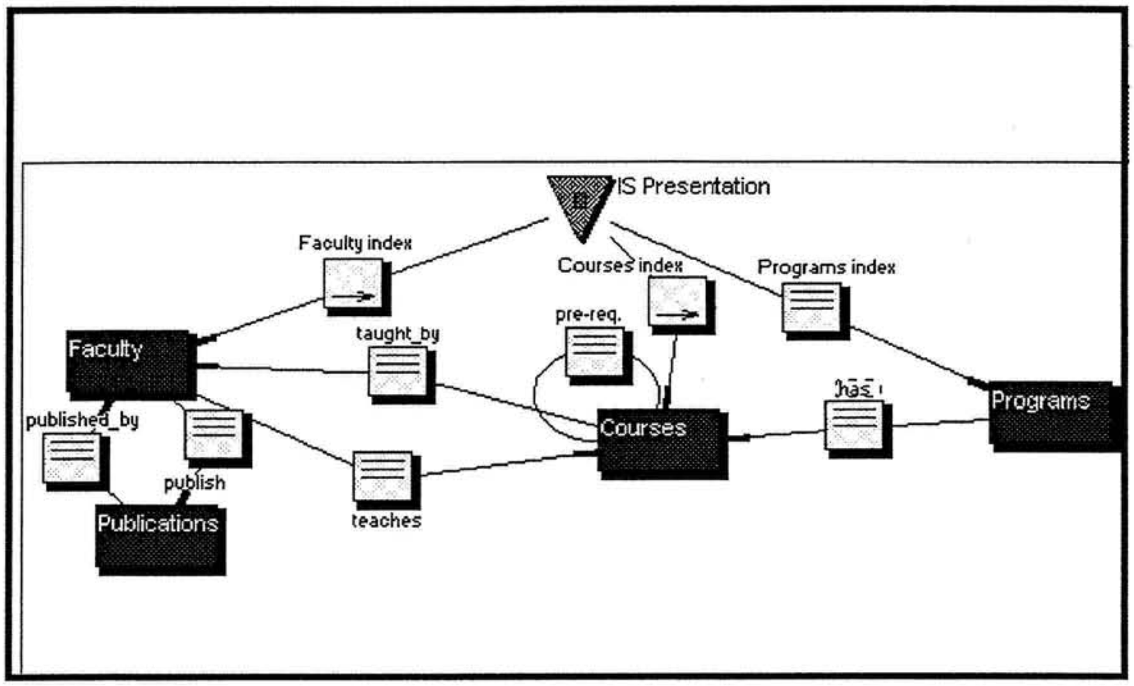


Figure 6: An RMD diagram representing the design of the ISWEB application.

Programs by means of an index. On choosing the guided tour to the *Faculty* entity, the user can move back and forth among all faculty members (ordered alphabetically). There is a conditional index from the *Faculty* entity into *Courses* with predicate *teaches(F,C)*. The reciprocal index *taught_by(C,F)* can be accessed from *Courses.*

The remaining steps of RMM involve designing the user-interface, (for WWW applications, the design of HTML templates), application population, and testing. I only provide a brief discussion in this article.

Most current hypermedia building kits such as Toolbook, Hypercard, Macromind Director, as well as tools used to create HTML documents, offer some degree of support for software development. The *conversion protocol design* step, uses a set of *conversion rules* to transform each element of the RMDM diagram into an object in the target platform. In the case of the WWW, HTML files.

The *user-interface design* step involves the design of screen layouts for every object appearing in the RMDM diagram. This includes button and text layouts, and location of navigational aids. Decisions about the population of the application with data are taken in the *run-time behavior design* step. Finally, *construction and testing* take place, much like in traditional software engineering projects. Special attention however, is to be given to a thorough test of all navigational paths.

Sidebar The RMDM Data Model

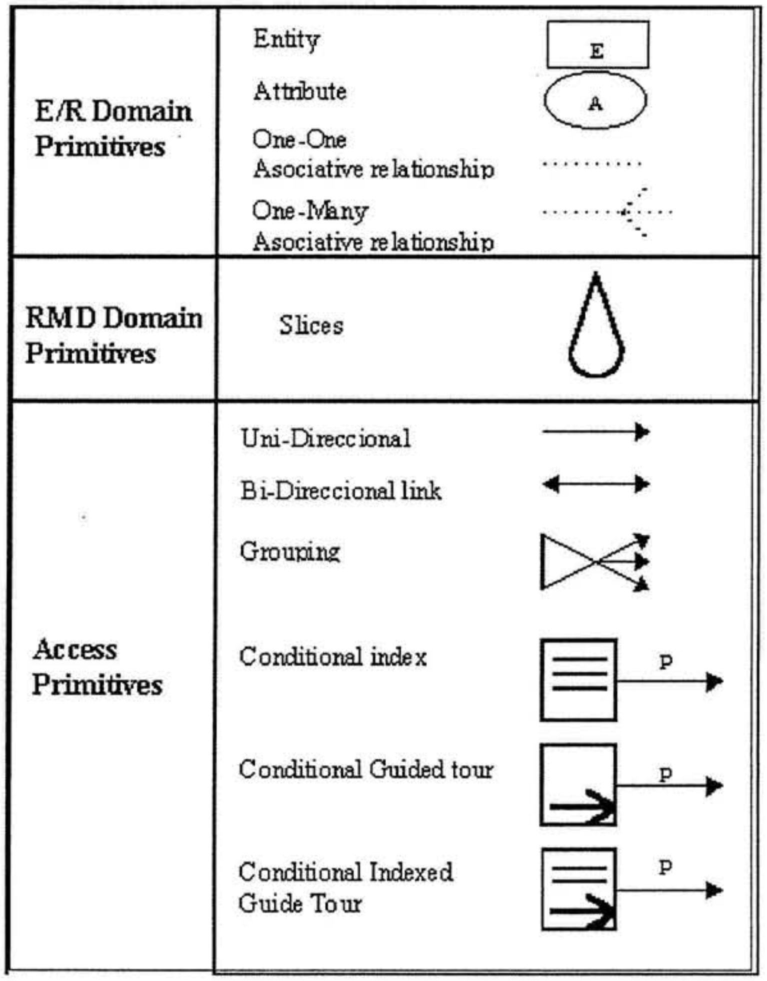
A cornerstone of the RMM methodology is its data model, the Relationship Management Data Model (RMDM), whose elements are shown in Figure 7. RMDM provides a language for describing information objects and navigation mechanisms in hypermedia applications. An application's design is specified with an RMD diagram, constructed from RMDM's elements (see Figure 6 on page 18). The RMDM model is based on the Entity-Relationship model [5] and on HDM[8]. ER relationships can be on-one, one-many, and many to many, representing associations between entities. As in database modeling, many-many relationships are factored into pairs of one-many relationships.

Because entities may have a large number of attributes of a different nature (e.g., salary information, biographical data, photograph), it may be impractical or undesirable to present all the attributes of an entity instance in one screen. Thus, attributes are grouped into slices. For example, a person entity with attributes name, age, photograph and biosketch, may have a General slice, containing name, age and photograph and a Biosketch slice, with name and biosketch. The notation for slices is shown in the middle portion of Figure 7 (it is supposed to resemble slice of a pizza).

Navigation is supported in RMDM by the six access primitives shown at the bottom of Figure 7. Uni- and bi-directional links are used to specify access between slices of an entity. The most significant access structures supported by RMDM are indices, guided tours and groupings. An index acts as a table of contents to a list of information items, providing direct access to each listed item. A guided tour implements a linear path through a collection of items allowing the user to move either forwards or backwards on the path. Index Guided Tours combine the functionalities of indices and guided tours. These three access structures are augmented with logical conditions that act as select statements specifying the set of instances being accessed. For example, a condition *rank=Associate* on an index into a faculty entity denotes an index to all associate professors. The grouping mechanism serves as an access point to other parts of the hypermedia document. For example, the initial screen of many applications contains a menu or set of buttons that provide access to different functions or classes of information.

Sidebar end

—



E/R Domain Primitives	Entity	E
	Attribute	A
	One-One Asociative relationship	
	One-Many Asociative relationship	
RMD Domain Primitives	Slices	
Access Primitives	Uni-Direccional	→
	Bi-Direccional link	↔
	Grouping	
	Conditional index	P
	Conditional Guided tour	P
	Conditional Indexed Guide Tour	P

Figure 7: The elements of the RMM Data Model (RMDM).

3.3 Application Generation

Once the design has been completed, the application can be tested with sample data and a prototype can be constructed [6] to aid in fine-tuning the various components of the application.

Once all the screen templates and access mechanisms have been defined, data needs to be incorporated. There are various alternatives that can be taken here. The manual approach, for example, requires typing-in the information, or cutting and pasting from other applications. It may involve a database whose contents are to be "poured" into the hypermedia application, or it may require developing computer programs that pipe the database contents into the hypermedia application as needed. Regardless of the details, it is important that there be a systematic procedure to generate the application. The RMM methodology contemplates a separate component, the Relationship Management System *(RMS)*, to manage this process. The RMS sets up gateways to enable the flow of information between databases and hypermedia applications.

3.4 Application Distribution

The distribution component consists of a broadcasting system that is in charge of disseminating the pre-generated hypermedia application elements to their owners (people, or organizations). In order to function effectively the distribution mechanisms must be able to (a) determine where the various parts of the application are to reside, (b) perform the broadcasting function, and (c) confirm that the broadcasts was successfully completed.

3.5 Design Updates

The distribution of design updates is quite complex. There are in fact, two sides to it: updates to the boiler-plate component, and updates to the variable portion. Let us briefly analyze these.

(1) A boiler-plate component is modified, for example, if we decide to add a new area, e.g., research seminar schedule, to the ISWEB application. Since in ISWEB, every page contains links to the top-most level pages, this change requires updating the design of all the distributed pages.

(2) The update is on the variable part. For example, a doctoral student decides to include information about "music" as a top-level section. The problem here is how to ensure that the changes made by the doctoral student do not interfere with other aspects of the central design .

These constraints on updates indicate that the "boiler-plate" and "variable" parts need to be kept separate. This ensures that individuals can not effect changes on the boiler-plate, and symmetrically, that central design updates leave variables parts intact.

A "clean room" approach suggests a system architecture where each Web page (or hypermedia node) is created dynamically by combining the variable and central parts. This would adequately support design updates. However, this approach reduces freedom because it forces every page to be generated through some kind of script. It is unlikely that individuals will find it attractive to struggle with script languages such as Perl to update their home-pages.

Alternatively, and this is the approach taken in ISWEB, special programs are executed whenever design updates are required. These programs inspect the required files and proceed to make necessary changes. The disadvantage of this approach lies in that others may inadvertently, or on purpose, remove or change the boiler-plate component in their files. As a result, the desired design updates may not be successfully implemented.

The updating programs can either be executed in a scheduled fashion, or on demand by users. We found the second approach to be rather intrusive, and hence we settled for the latter one. Upon login, users are advised when design updates are due, and, upon their approval, updates are performed.

4 An Example of Distributed Design

The ISWEB application [1] called for an implementation of the distributed design ideas presented earlier. Accordingly, the following three tiered approach was adopted:

1. **Generation:** a centralized design, based on the RMM [10] is used to centrally generate the application using a common design.

2. **Distribution:** once generated, individual components are distributed over the network as needed. Thereafter, the application becomes vulnerable to a distribution of design, because individual owners are able to modify at will the HTML files they own.

3. **Design updates:** design updates are propagated as they needed with a semi-automated method that broadcasts update scripts.

4.1 Challenges faced in Designing ISWEB

ISWEB presented a number of challenges, some related to the WWW and others to the nature of the application. For starters, since personal home-pages contain personal information, it seemed sensible to assume that most individuals would be interested in updating them. Therefore, we devised a mechanism to allow the transfer of ownership of homepages.

We encountered three kinds of individuals. (1) The *hands-off*, who didn't feel comfortable with HTML, and therefore preferred their home-page to be centrally owned, and also maintained. (2) The *doers*, technically sophisticated individuals that prefer to build their own homepages, and are somewhat reluctant to incorporate centralized design directives. (3) The *eager*, individuals who want the ability to customize and update their homepages, but prefer to work on pre-generated home-pages.

The ISWEB approach caters to the needs of the first and third groups. In general, individuals in the third group tend to keep the design intact, updating only their personal information. The development team was surprised at the second group's resistance. Being technically savvy, group 2's individuals, we thought, would be readily to adopt the design guidelines of ISWEB.

Group two however, was quite small (10%). The first group, "hands-off", proved to be, by far, the most popular (60%), many of which migrate to group three, the "eagers". (initially 30%).

From a technical perspective, the most significant challenge lies in updating designs. The three groups require different methods.

For the hands-off group (1), we designed a system to solicit updates and incorporate them into the home-pages. Group 2, the *doers,* requires frequent oversight to enforce the overall ISWEB design guidelines. Frequent reminders need to be send over e-mail to members of this group to get them to comply with design updates. The third group, the *eager*, requires close follow-up due to their inexperience with HTML.

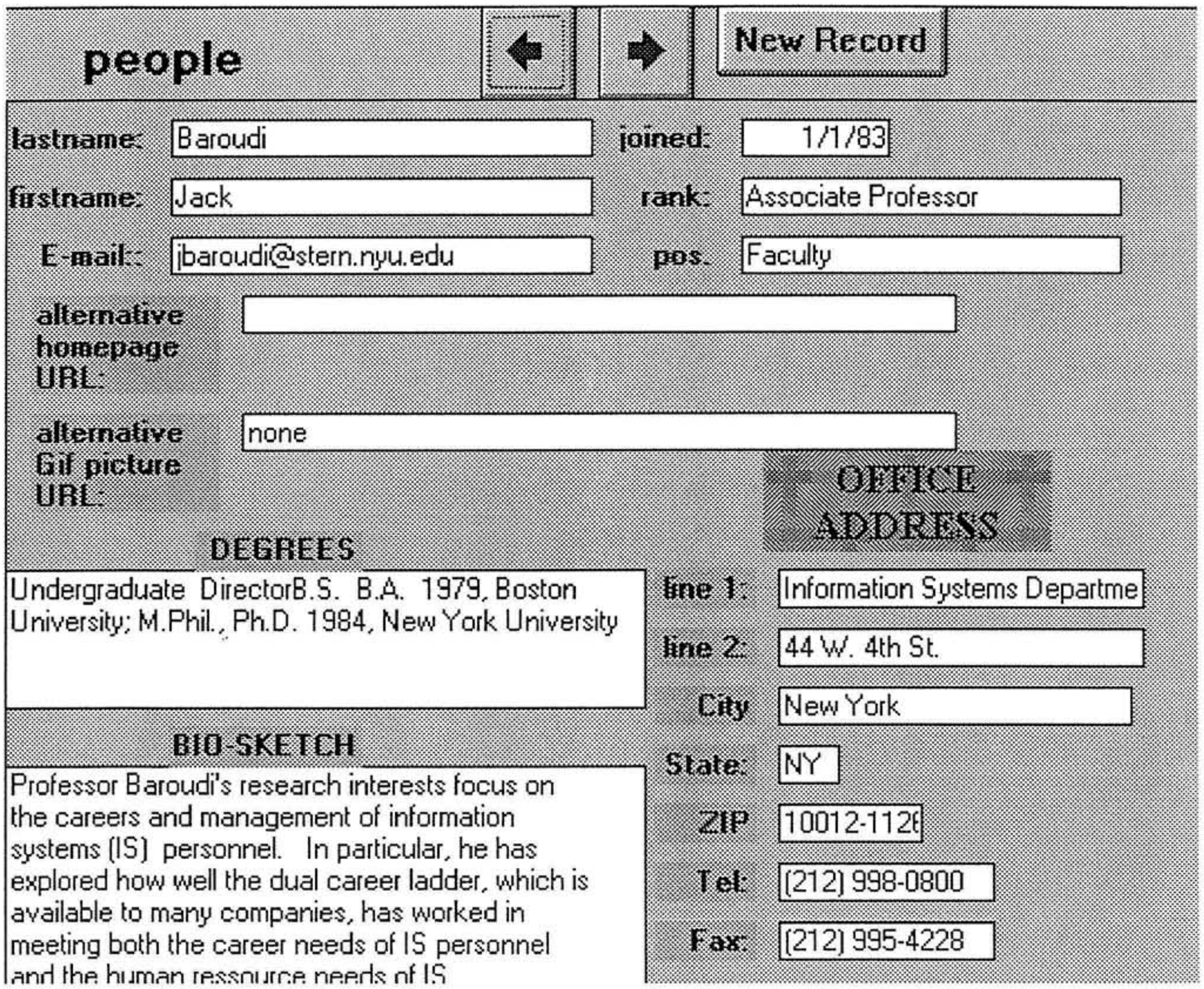
The main challenge, once the initial construction of ISWEB is complete, remains with the design updates. Part of my efforts, on which I report here, are geared towards the development of tools and methods to facilitate content updates without compromising central design guidelines.

4.2 Generation of ISWEB

The RMM methodology was used to design ISWEB, as reported earlier. Data to populate the application was funneled from various existing documents and databases. For example, an Access database was created to contain information about all faculty in the department. To facilitate data updates, this database was setup on a LAN. A special form, shown in Figure 8 was designed to facilitate entering updates.

We created a sample template for the personal homepages of faculty based on the RMDM design of ISWEB, with six slices per person. The design was tested with sample data, and iterated until we were satisfied with the outcome. The HTML templates and the data from the database were combined using a mail merge facility to produce a long file containing HTML code for all persons in the database, six interlinked slices per person. Figure 9 illustrates this process.

The file was then uploaded to the server machine, form where it is acces-



lastname:	Baroudi	joined:	1/1/83
firstname:	Jack	rank:	Associate Professor
E-mail::	jbaroudi@stern.nyu.edu	pos.	Faculty

alternative homepage URL:

alternative Gif picture URL: none

DEGREES

Undergraduate DirectorB.S. B.A. 1979, Boston University; M.Phil., Ph.D. 1984, New York University

BIO-SKETCH

Professor Baroudi's research interests focus on the careers and management of information systems (IS) personnel. In particular, he has explored how well the dual career ladder, which is available to many companies, has worked in meeting both the career needs of IS personnel and the human ressource needs of IS

OFFICE ADDRESS

line 1:	Information Systems Departme
line 2:	44 W. 4th St.
City	New York
State:	NY
ZIP	10012-112
Tel:	(212) 998-0800
Fax:	(212) 995-4228

Figure 8: Data is placed into an Access database. The form shown here facilitates data entry. The database contents are used to populate the hypermedia application.

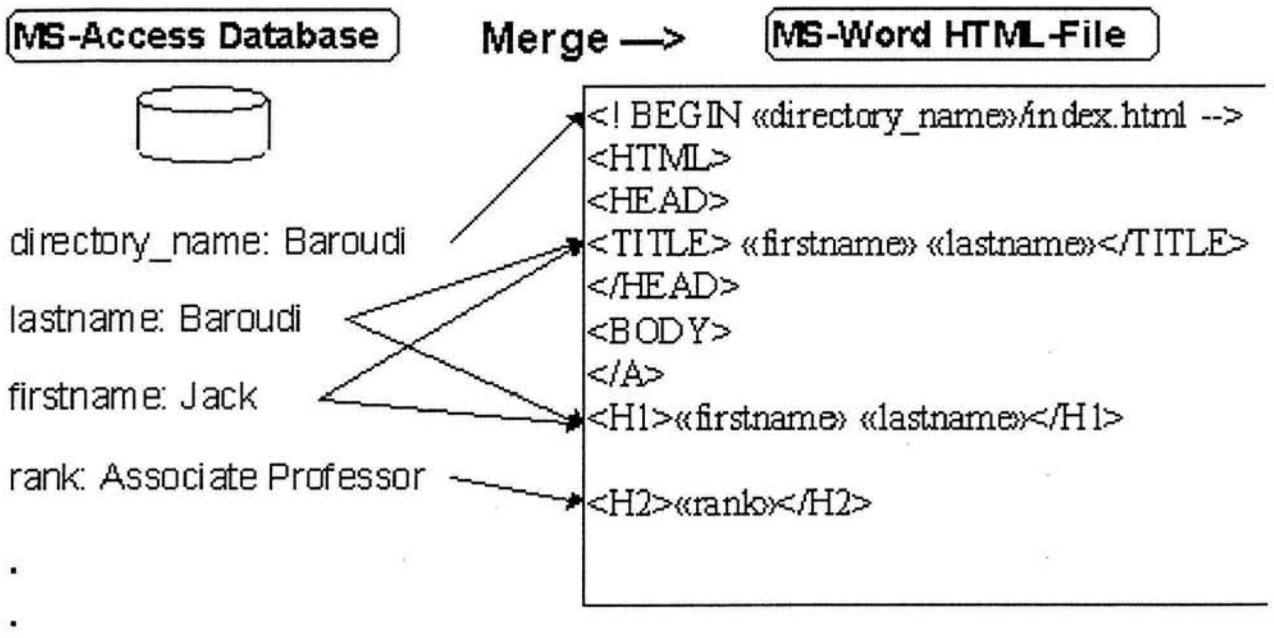


Figure 9: Merging the database records with HTML templates to generate HTML pages

sible on the WWW. The initial design called for a single, albeit long, HTML file per person, containing all the slices, and using internal anchors to support navigation, e.g., from the "teaching interests area" to the "biosketch". This turned out to be inefficient because lengthy files take too long to load. So I reverted to a design that called for storing each slice in a separate HTML file. A Perl script was used to split the long file into six files per person. A separate directory was created for each person, using the person's account id as the directory name. Each directory contains six files, one per slice.

4.3 Generation in ISWEB

A Perl script was written to enable distribution of home-pages to the owners. The script operates by copying the six HTML files from the centralized ISWEB directory into the user's "public_html" sub-directory, and by creating a pointer that indicates to the ISWEB page server the new location of the HTML page. As a precaution, the script first checks if the user already has a "public_html" directory. If so, it creates a copy of this directory and ports the HTML pages over. Once copied over, the HTML pages are owned by their real owners, who can proceed to edit them at their own will.

4.4 Distribution in ISWEB

Numerous tuning changes that had to be made to the design as the application evolved. Many of these changes were undertaken before the homepages were distributed, while residing in a central account. The most significant change was the addition of a "home-page" box to the navigational map, and the shading of the slices in it.

4.5 Design updates in ISWEB

Updates were carried out via PERL scripts that globally replaced the GIF file used as navigational map in all homepages. Thanks to our use of the enhanced Imagemap, navigation itself was coordinated through a common Imagemap program for all the home-pages. Hence only one program had to be updated.

Distribution of updates once the homepages had been acquired by the owners required a slightly different approach. Upon login, users who had downloaded their pages receive a message indicating that updates are due. If they wish so, they can run the update process then.

The contribution we implemented in Imagemap 2.0 enables sharing the same map program among multiple pages. In fact, there is only one map program used for all of 180 HTML files that constitute personal home-pages in ISWEB.

4.6 Enhanced Imagemap

Our approach to distributed design lead to the development of simple but useful tools. One such example is Imagemap 2.0 [3]. To understand its impact, let me observe that whenever a graphical map is used in a web page, an accompanying program is needed to specify what areas in the drawing are to be linked to what URLs. One such program is needed per map usage. Now, our navigational aids called for a shared navigational map among all pages. Using the existing version of Imagemap we would have needed one map program per HTML page. Literally hundreds of them! Besides the obvious space inefficiencies this duplication carried with it, there are significant maintenance considerations involved should a design update be required. Adding for example, a new slice would require updating $6 * n$ programs (where n is the number of personal homepages).

5 Future Work

The architectural details described in this paper are generalizable into a complete design management system. I plan to investigate the components of such a plan and incorporate these into hypermedia software development environments.

Probably most future distributed hypermedia applications will relate to electronic commerce. This is an area that requires integrating hypermedia applications with traditional information systems (e.g., accounts receivable). Many additional challenges need to be faced regarding a seamless integration of hypermedia information systems with such information systems used in organizations [2].

6 Summary

In this paper I described the problem of distributed hypermedia design, presented a general scheme for a solution and illustrated it via specific examples in ISWEB. The framework I put forward consist of a system able to manage a three-tiered approach (1) generation, (2) distribution, and (3) design updates. Specific implementation details were given for the ISWEB application. These are generalizable. In fact, the ISWEB development lead to to the construction of tools, such as Enhanced Imagemap 2.0, to support distributed design on a large scale.

Clearly, there are many shortcomings to this approach, most of which lie within the update stage. Further experimentation with actual usage will help determine more thorough solutions to this problem.

What is clear however, is that a systematic approach to support distributed hypermedia design is needed and, indeed, possible.

References

[1] URL: http:\\www.stern.nyu.edu.

[2] Michael P. Bieber and Charles J. Kacmar. Designing Hypertext Support for Computational Applications. *Communications of the ACM*, 38(8):99–107, August 1995.

[3] Victor Boyco, Mark Ginsburg, and Tomás Isakowitz. Enhanced Imagemap 2.0: Web Design Advantages Conferred by Parameterization. Working Paper Series IS-95-10, Center for Research in Information Systems - New York University, Stern School of Business, 44 West 4th St., New York, NY 10012, July 1995.

[4] Minder Chen. A Model-Driven Approach to Accessing Managerial Information: A Repository-Based Executive Information System. *Journal of Management Information Systems*, 11(4):32–63, Spring 1995.

[5] Peter P. S. Chen. Database Design Based on Entity and Relationship. In S. Bing Yao, editor, *Principles of Database Design, Volume 1: Logical*

Organizations, chapter 5, pages 174–210. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, USA, 1985.

[6] Alicia Díaz and Tomás Isakowitz. RMCase: Computer-Aided Support for Hypermedia Design and Development. Working paper IS-95-3, Center for Research in Information Systems, New York University, Information Systems Department, New York, NY 10012, 1995.

[7] Franca Garzotto, Paolo Paolini, and Luca Mainetti. Hypermedia Design, Analysis and Evaluation Issues. *Communications of the ACM*, 38(8):74–86, August 1995.

[8] Franca Garzotto, Paolo Paolini, and Daniel Schwabe. HDM - A Model-Based Approach to Hypertext Application Design. *ACM Transactions on Office Information Systems*, 11(1):1–26, January 1993.

[9] James B. Baty II and Ronald M. Lee. InterShop: Enhancing the Vendor/Customer Dialectic in Electronic Shopping. *Journal of Management Information Systems*, 11(4):9–31, Spring 1995.

[10] Tomas Isakowitz, Edward. A. Stohr, and P. Balasubramanian. RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, 38(8):34–44, August 1995.

[11] Ajit Kambil and Mark Ginsburg. The Edgar Project. URL: http:\\edgar.stern.nyu.edu, 1994.

[12] 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.

[13] Jiye Mao, Izak Benbasat, and Jasbir Singh Dhaliwal. Enhancing Explanations in Knowledge-Based Systems with Hypertext. *JOC, forthcoming*, 1995.

[14] R.P. Minch. Applications and Research Areas for Hypertext in Decision Support Systems. *Journal of Management Information Systems*, 6(3):119–138, Winter 1989-90.

[15] Jocelyne Nanard and Marc Nanard. Hypertext Design Environments and the Hypertext Design Process. *Communications of the ACM*, 38(8):49–56, August 1995.

[16] Christine A. Quinn. From Grass Roots to Corporate Image - The Maturation of the Web. . In *Proceedings of the Second WWW Conference: Mosaic and the Web*, Chicago, IL, 1994.