

A CASE STUDY IN SYSTEMATIC HYPERMEDIA DESIGN

V. BALASUBRAMANIAN⁽¹⁾, MICHAEL BIEBER⁽²⁾, TOMÁS ISAKOWITZ⁽³⁾

(1) E-Papyrus Inc., 1 Gramercy Road, Monmouth Junction, NJ 08852 USA; bala@e-papyrus.com; <http://www.e-papyrus.com>

(2) Collaborative Hypermedia Research Laboratory, Information Systems Department, New Jersey Institute of Technology, University Heights, Newark, NJ 07102 USA; bieber@njit.edu; <http://www-ec.njit.edu/~bieber>

(3) Research Department, Janney Montgomery Scott, 1801 Market St., Philadelphia, PA 19103 USA; isakowitzt@acm.org; <http://www.stern.nyu.edu/~tisakowi>

Abstract — Hypermedia structuring and navigation requires design methodologies different from those developed for standard information systems. This case study details our successful application of Relationship Management Methodology (RMM), a hypermedia systems analysis and design methodology, to LINKBase. LINKBase is a World Wide Web (WWW) application, which dynamically generates WWW pages from a relational database containing information about hypermedia-related events such as conferences, publications, authors, and sponsoring organizations. We describe our experience applying RMM in this case study, summarize design lessons we learned in the process, present extensions to RMM, discuss Human-Computer Interaction (HCI) aspects of RMM, and ground our work in the hypermedia design and HCI literature. Our experiences should encourage hypermedia and WWW developers to utilize systematic design techniques to build highly usable and useful WWW applications.

Key words: hypertext, hypermedia, design methodologies, design guidelines, entity-relationship diagrams, structure, database application design, navigation, user interface, browsing semantics, information access, indexes, guided tours, World Wide Web database gateways, LINKBase, usability evaluation, functionality evaluation

1. INTRODUCTION

The World Wide Web (WWW) and other hypermedia systems are characterized by three major design dimensions: structure, navigation and presentation [23]. Hypermedia authors exploit interrelationships (links) for structuring information in a non-linear network of information chunks. Hypermedia systems provide various access and navigational structures allowing users to move through an application, including indexes, trails and tours, indexes, history-based navigation, and various permutations of link traversal, among others [12, 48]. These structural features must be designed and displayed in a manner intuitive to users.

Hypermedia structuring and functionality requires design methodologies different from those developed for standard information systems. This article details our successful application of a hypermedia systems analysis and design methodology to a WWW application. We present a case study using the Relationship Management Methodology (RMM) by Isakowitz, Stohr and P. Balasubramanian [34], propose extensions to RMM, summarize the “RMM design” lessons we learned in this process, describe HCI implications of RMM, and ground our experience in the hypermedia design literature. Our experiences should help pave the way for other developers to take advantage of hypermedia design and development techniques, both on the WWW and in more traditional applications.

What makes hypermedia so useful? We view hypermedia as the science of relationships and hypermedia techniques as supporting relationship management. Hypermedia concerns structuring, presenting and giving users direct access to the content and interconnections within an information domain. A hypermedia vantage point encourages designers to consider a system in terms of the relationships among its objects and processes, focusing on what users might want to access and how users should access them [11,12,13].

Handcrafting hypermedia interrelationships is tedious at best, and impractical in cases of voluminous or frequently changing information. Furthermore, authoring a hypermedia interface and hypermedia links in an *ad hoc* format leads to inconsistent design [25], and the possibility of errors and omissions [14, 15]. Therefore, a systematic design approach with automatic page generation constitutes the only reasonable option in many cases.

Isakowitz et al., [34] focused on the first three of RMM’s seven stages and provided initial developer guidelines for these. The current article reviews RMM’s seven stages, extends some aspects of the first three stages, and describes the remaining four stages in detail. We also present the first full case study of a step-by-step application of this hypermedia design methodology. While many WWW applications also retrieve display contents from databases (see, for example, CommerceNet [18] and Johns Hopkins University’s [35] public domain bioinformatics databases), ours is the first published description of a

WWW application designed with a systematic hypermedia design methodology. Our emphasis in the previous sentence is twofold: on published and on systematic hypermedia design. We assume these other applications resulted from some degree of design. Yet without any published record we do not know how much. And, more important, others can not learn from them.

We begin in section 2 by reviewing RMM and our extensions to it. Section 3 discusses related hypermedia design research. In section 4 we introduce our application, LINKBase. Section 5 presents our case study applying RMM to LINKBase. Section 6 takes a larger view of RMM, its strengths and limitations, and discusses issues such as hypermedia requirements analysis and retro-fitting legacy applications. We conclude in section 7 with some final motivation.

2. THE RELATIONSHIP MANAGEMENT METHODOLOGY (RMM)

The Relationship Management Methodology (RMM) addresses the design and construction of hypermedia applications. We begin this section by briefly presenting RMM and its data model, RMDM. A more detailed discussion can be found in [34].

2.1 Methodological Steps

RMM consists of the following seven steps, some of which can be conducted in parallel: (1) Entity-Relationship Design: models the information domain and its relationships, (2) Slice Design: how information units are sub-divided for display, (3) Navigational Design: how users will access information, (4) User-Interface Design: how information will be presented, (5) Conversion Protocol Design: how abstract constructs are to be transformed into physical-level constructs, e.g., what kind of WWW page corresponds to an index, (6) Run-time Behavior Design: how to populate the application with data and how to provide interaction behavior, and (7) Construction and Testing: actual development of programs and testing.

RMM was conceived to be flexible by supporting rapid feedback loops as prescribed in [44]. This is embodied in its software design tool, RMCASE [21]. Although not explicit in the methodology, HCI aspects are an integral part of RMM and we will present these in the discussion of each of the seven steps.

2.2 The RMDM Data Model

The Relationship Management Data Model (RMDM) is the cornerstone of the RMM methodology. Figure 1 presents its elements. RMDM includes elements for representing information domain concepts (such as entities and relationships), and navigation mechanisms (such as links). An application's design is described via an RMD diagram (see Figure 9). The RMDM model is based on the Entity-Relationship model [22], and on HDM [27] and HDM2 [26].

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, RMM groups attributes into slices. Splitting an entity into slices also reduces clutter of the information display. Only the most essential information is presented upfront with details being available as links.

RMDM specifies navigation via the six access primitives at the bottom of Figure 1. RMDM's most significant access structures are indices, guided tours, indexed guided tours and groupings. An index acts as a table of contents. A guided tour implements a linear path through a collection of items, allowing the user to move forwards or backwards on the path. Indexed guided tours combine the functionality of indices and guided tours. Logical conditions qualify these access structures. For example, attaching the condition "type='panel'" to an index of an event entity denotes an index to panels from that event. The grouping mechanism serves as a major access gateway to other parts of the system, as often found on many applications' home pages or initial screens.

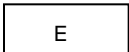
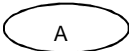
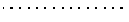
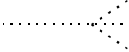
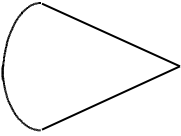
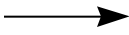
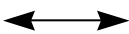
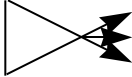

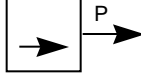
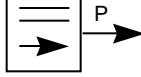
E-R Domain Primitives	Entity	
	Attribute	
	One-One Associative Relationship	
	One-Msny Associative Relationship	
RMD Domain Primitives	Slice	
Access Primitives	Uni-directional Link	
	Bi-directional Link	
	Grouping	
	Conditional Index	
	Conditional Guided Tour	
	Conditional Indexed Guided Tour	

Figure 1: The elements of the RMM Data Model (RMDM)

2.3 Extensions to RMM

Our experience enabled us to identify two useful additions to RMDM (1) minimal slices, and (2) hybrid slices. A minimal slice is a minimal collection of attributes of an entity type that enable users to identify entity instances uniquely. Although conceptually similar, RMDM minimal slices differ from database keys, because the latter usually do not convey relevant information (at a cognitive level) to users. For example, although an employee’s social security number may serve as a database key, a minimal slice may instead comprise the employee’s name and department. In hypermedia applications, minimal slices are used as anchors to entity instances.

Hybrid slices combine attributes from different entities and access structures. RMM’s original slices are “pure” in that they combine attributes of the same entity type. Since every screen in an RMM application originates in a slice or in an access structure, the purity of slices severely restricts the allowable pages. For example, using pure slices, it is impossible to combine into one screen the name of a paper in a conference and an index of its authors. Hybrid slices can be used to that effect. We illustrate these new kinds of slices in section 5.4. Isakowitz et al. provide a formal description of hybrid slices (called “Matrjeska slices”), along with some further extensions to RMM which we did not use in this case study [33].

3. RELATED RESEARCH

Hypermedia applications have a strong focus on user navigation and a structure that exploits the interrelationships among application elements. Links implement relationships in WWW and other hypermedia applications. Hypermedia structuring and functionality requires design methodologies different from those developed for standard information systems. Hypermedia applications involve many different components, such as content preparation, structure, storage, user-interface, navigation, and retrieval [23]. As a consequence, data models such as data flow diagrams, entity-relationship (E-R) diagrams and object-oriented hierarchies cannot represent the design intricacies hypermedia applications encompass.

Garzotto, Paolini, and Schwabe's HDM data model [27] and its successor HDM2 [26] describe the structure of a database application domain adequately to support hypermedia access. HDM and HDM2 describe representation schemes, but provide little guidance on using those representations in the design process. In other words, while they describe an application domain; they do not constitute a hypermedia design and development methodology. RMM builds on HDM and HDM2 to provide this full methodology. OOHDM [57, 63] extends HDM by adding additional stages onto the process: conceptual design, navigational design, abstract interface design and implementation, also making it a more comprehensive design methodology. Lange [37] has proposed a hypermedia design methodology based specifically on the object-oriented paradigm. For database domains RMM has the advantage of using tools such as E-R diagrams, with which designers already are familiar. In fact, as we note in section 5.10 designers using RMM especially have lauded this familiarity. Christodoulou et al. compare several hypermedia design methodologies [17].

At the other extreme, Marshall and Shipman et al. [41, 43, 42, 65, 66] discuss authoring in spatially-oriented hypermedia environments. In spatial hypertext, relationships may be left implicit, inferable only from the proximity of content with each other on the display. In addition, authors do not have to commit to a structure and its enforced consistency in advance. One could view this as an excuse to bypass consistency guidelines proposed by RMM and HDM; alternatively, time may show that the best spatial applications evolve to a state consistent with these guidelines. It is possible, in principle, to produce spatial designs from an underlying RMD diagram (section 5.5), as does RMCASE [21]. The interaction between the structured RMM methodology and spatial hypertexts remains to be explored.

Hypermedia designers are adopting the software engineering concept of design patterns [24], which capture similarities of execution and use across different applications and domains, especially to promote reuse. Hypermedia design patterns focus on relationship-based structure, navigation and presentation within applications [7, 56, 57, 58, 59, 69]. The community has devoted a Web site identifying and collecting hypermedia design patterns [30].

In the software engineering life cycle, analysis generally precedes design. A hypermedia analysis would produce hypermedia-oriented requirements for structural and navigational elements, and a preliminary set of elements and relationships to include in an application. Whichever design methodology is used would then help the application designer to determine the best way to present these to the user. Most hypermedia design methodologies, including RMM, only implicitly help analyze the application environment, to a large degree assuming the designer knows what she or he wants to include in the application. There are many popular analysis techniques in the software engineering field, but none focus specifically on the relationships that characterise a hypermedia-supported application. Yoo and Bieber have recently begun work on Relationship-Navigation Analysis (RNA), which analyzes an information domain explicitly in terms of the interrelationships among its elements [13, 74, 75, 76]. Performing an RNA would provide an initial set of elements and relationships, for example, to include in the entity-relationship diagram, with which RMM would then work.

Our experience follows Nanard et al.'s observation that hypermedia design "is an incremental and opportunistic human activity that takes place in a two axes space," in which one axis represents the technical aspect and the other axis represents the design process [44]. Employing RMM certainly had both aspects. Nanard et al. also identified fast feedback loops as a fundamental requirement of a hypermedia development environment. As we shall describe starting in the following section, throughout LINKBase's design and development, we reiterated through design-implementation loops. Lowe, Buckness and Webby also study the hypermedia development process, proposing a reference model-based assessment method [39].

We direct the reader to our prior research in systematic hypermedia user interface design [3] and in automatically generating hypermedia based on an application's internal structure [7, 8, 10]. These do not have the breadth of RMM. Also we note that Isakowitz et al. describe the design of the Web site of the Journal of Management Information Systems, which was based on an RMM design, though not in the detail presented in this case study [33].

4. LINKBASE

LINKBase is a bibliographic application using information stored in a relational database management system. LINKBase contains information on events (conferences, workshops, special issue of journals, etc.), event items (actual articles), event publication(s) containing the articles (conference and workshop proceedings, books, URL addresses), the organization(s) sponsoring the Event (e.g., ACM, INRIA, etc.), and people (authors, event managers, and organization officers). LINKBase is more than a database, it is a full-fledged hypermedia application which processes and reformulates data extracted from a relational database. The WWW was chosen as the delivery platform so the widest possible audience could access LINKBase easily and free-of-charge.

The following scenario illustrates LINKBase. A user interested in information about hypermedia reaches LINKBase. After traversing some introductory material, he or she selects "conferences" from a list of event types. The user arrives at the dynamically generated page shown in Figure 2a, which contains an index of hypermedia related conferences. Upon selecting "ECHT '94" (European Conference on Hypermedia Technology 1994), the system generates the WWW page shown in Figure 2b, containing basic conference information including name, date, location, sponsor ("ACM"), and related publications ("ECHT '94 Proceedings"). Other navigation choices include "previous conference" and "next conference" for the index. Other links (not visible here) can lead to photographs taken at the conference, trip reports, a list of committee members, etc. At any time, the user has access to all major system entry points ("groupings") for events, authors, organizations and publications.

The user then traverses the link "Table of Contents" to the page shown in Figure 3a, containing a list of papers, posters and demos, etc., presented at ECHT '94 along with author names. Note that the "Navigation Path" reflects the user's movement deeper into the information space. Selecting "Adding Multimedia Collections to the Dexter Model," the user arrives at Figure 3b. From here the user can select author names and navigate to details about their affiliation and contact information.

5. LINKBASE CASE STUDY

Like many Web applications, LINKBase's original version was designed in an *ad hoc* manner without following a methodology. In retrospect, this resulted in a system too limited in content and access mechanisms, containing only conference-related event items and co-author cross references. Source information was kept in flat files. The system administrator executed utilities to periodically update WWW pages in Hypertext Markup Language (HTML) and to re-generate navigation indices.

This section describes LINKBase's design and development using RMM. We will present the relevance of HCI principles with respect to the seven design stages, where applicable.

5.1 Requirements Analysis

We, the designers, played the role of users (both content providers and readers) while discussing the requirements for the new LINKBase. The second author who implemented the original LINKBase knew its deficiencies: primarily, it lacked a lot of content relevant to hypermedia researchers. Access and update mechanisms were also very limited. Although usability aspects were not of primary importance while re-designing the system, as we discuss in section 5.12, the discipline encouraged by RMM results in a system with a high-degree of usability and utility. The second author along with the first author and two students taking a course in hypermedia discussed the functionality and content required for a comprehensive bibliographic system on hypermedia. This approach is similar to other design approaches reported by Salomon [60], that is, the designer taking the role of the user. Based on discussions amongst ourselves, we arrived at the following new requirements for the system:

- Improve the information content of LINKBase by incorporating missing information. The original LINKBase had a list of conferences, list of papers and their abstracts and some information on authors. Also, all this information was restricted to events organized by the ACM. A lot of hypermedia related information was available in the form of workshop reports, other conferences, books, technical reports, etc. In addition to these other sources, we also wanted to provide complete author information (such as affiliation, contact address, email, etc.), more information about the event (location, dates), event managers, sponsoring organizations (ACM, IEEE, universities, etc.), and resulting publications. Being researchers ourselves, we felt that these would be additional pieces of information in which other researchers would be interested.
- Provide easy mechanisms for content providers to update information about themselves and the organizations they represent.
- Provide an easy-to-use and intuitive interface using the World Wide Web as the medium so that the information is easily accessible by researchers all over the world.
- Develop the system in a systematic manner so that it is easy to maintain and update.
- Generate HTML documents dynamically from an up-to-date source so that HTML documents need not be created and maintained manually.

Hence, the primary purpose of re-implementing LINKBase was to provide additional information to researchers interested in the field of hypermedia, increased functionality in terms of various ways to access the information, a simple-to-use user interface, and an easy update mechanism.