Towards Integrating Hypermedia on the Web

Chao-Min Chiu
Department of Information Management
National Kaohsiung First University of Science and Technology
cmchiu@ccms.nkfust.edu.tw

Michael Bieber
Information Systems Department
New Jersey Institute of Technology
bieber@njit.edu

Qiang Lu
Computer and Information Science Department
New Jersey Institute of Technology
q15@njit.edu

Abstract

A major goal of this research is to present a framework for discussing issues and questions around Web Information Systems (WIS). WIS dynamically generate their contents, and thus require some mechanism to automatically infer metadata about WIS objects, infer access to relationships (i.e., links) among information objects, and provide hypermedia functionality such as annotation and ad hoc (user-declared) linking. The framework focuses on integrating information systems into the Web and providing hypermedia functionality to them. This should result in new ways to view and manage the WIS' knowledge and information relationships. The framework has three layers: interface, integration and infrastructure. The interface layer highlights the importance of designing Web interfaces appropriate for system functionality. The integration layer concerns integrating information systems, hypermedia support and inter-application interoperability. The infrastructure layer concerns using the Internet and Web technologies to support interoperability, hypermedia, and overall development at the integration layer. This framework should help system developers think more fully about their designs.

1. Introduction

With an eye towards enhancing their competitive position, many organizations are rushing to develop Web information systems (WIS) for users to access information. Many system designers and developers focus on implementing business process and system functionality of individual information system and designing attractive Web pages by using most up-to-date or popular Web technologies and simple linking functionality. They might not concern the differences between Web page and WIS design [19], the importance of integrated information systems, the benefits of rich hypermedia functionality, and the sustainability of systems. This constitutes much of the motivation for our research.

A major goal of this research is to present a framework for discussing issues and questions around integrating information systems into the WWW, and providing hypermedia functionality to them. This framework should help system developers think more fully about their designs.

We view hypermedia as value-added support functionality [6]. Hypermedia structuring, annotation and navigational functionality can enrich business, scientific, engineering and personal applications, thereby making them more effective. People use these applications primarily for their underlying analytical functionality, i.e., not for their ability to produce hypermedia documents or displays. People will not abandon the applications they use everyday in favor of those that offer hypermedia. Therefore we need to bring hypermedia support to these applications.

Web information systems are systems built either from scratch or by reengineering existing information systems by using Web technology. These can include systems associated with, e.g., decision support, executive information, finance, data mining, on-line analytical processing (OLAP), knowledge management, and digital libraries. WIS dynamically generate their contents and thus require some mechanism to automatically infer metadata about WIS objects, infer access to relationships among information objects, and provide hypermedia functionality such as annotation and ad hoc (user-declared) linking [11, 12].

This paper is organized as follows. In Section 2 we present the hypermedia-aware Web information system framework. In Section 3 we discuss issues and questions for the interface layer. In Section 4 we discuss issues and questions for the integration layer and present guidelines for building mapping rules. In Section 5 we discuss issues and questions for the infrastructure layer. Section 6 concludes with an overview of our future research directions.

2. Hypermedia-Aware Web Information System Framework

Our hypermedia-aware Web information system (HWIS) framework has three layers: interface, integration and infrastructure. It focuses on integrating information systems into the Web and providing hypermedia functionality to them. The interface layer highlights the importance of designing Web interfaces appropriate for system functionality. The integration layer concerns integrating information systems, hypermedia support and inter-application interoperability. The infrastructure layer concerns using the Internet and Web technologies to support interoperability, hypermedia, and overall development at the integration layer. Given this framework, system developers can think more fully about interface and hypermedia design, and appropriate technologies for supporting integration and interoperability before they start to build their WISs either from scratch or by reengineering existing information systems.

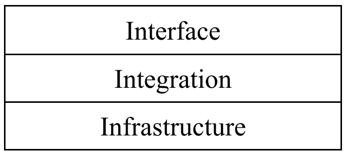


Figure 1. A three-layer framework for hypermedia-aware Web information systems

3. Interface Layer

The interface layer highlights the importance of designing Web interfaces appropriate for system functionality. This includes navigational mechanisms and access to a system's primary functionality. Web modeling languages exist for designing the entire Web site or just user interface, including Web Modeling Language (WebML) [10] and User Interface Markup Language (UIML) [1]. Some hypermedia design methodologies are also well suited to designing interfaces of Web applications, including Object-Oriented Hypermedia Design Methodology (OOHDM) [27] and Relationship Management Methodology (RMM) [20]. WebML is a modeling language for designing complex Web sites at the conceptual level. Its concepts are associated with an intuitive graphic representation and an extensible Markup Language (XML) syntax [10]. WebML enables the specification of a Web site under five orthogonal perspectives: at the content (structural model), the pages that compose it (composition model), the links between pages (navigation model), the layout and graphic appearance of pages (presentation model), and the customization of content delivery [10]. UIML is an XML-compliant language, which allows designers to describe the user interface in an appliance-independent manner and then map the interface to various operating systems and appliances through style sheets [1]. The abstract interface design stage of the OOHDM methodology involves defining perceivable interface objects and appearance of the navigational objects, and specifying the activation relationship between interface and navigational objects [27]. The user-interface design stage of RMM is for designing the look of anchors, button layouts, appearance of nodes and indices, positioning of video and images, and location of navigational aids [20].

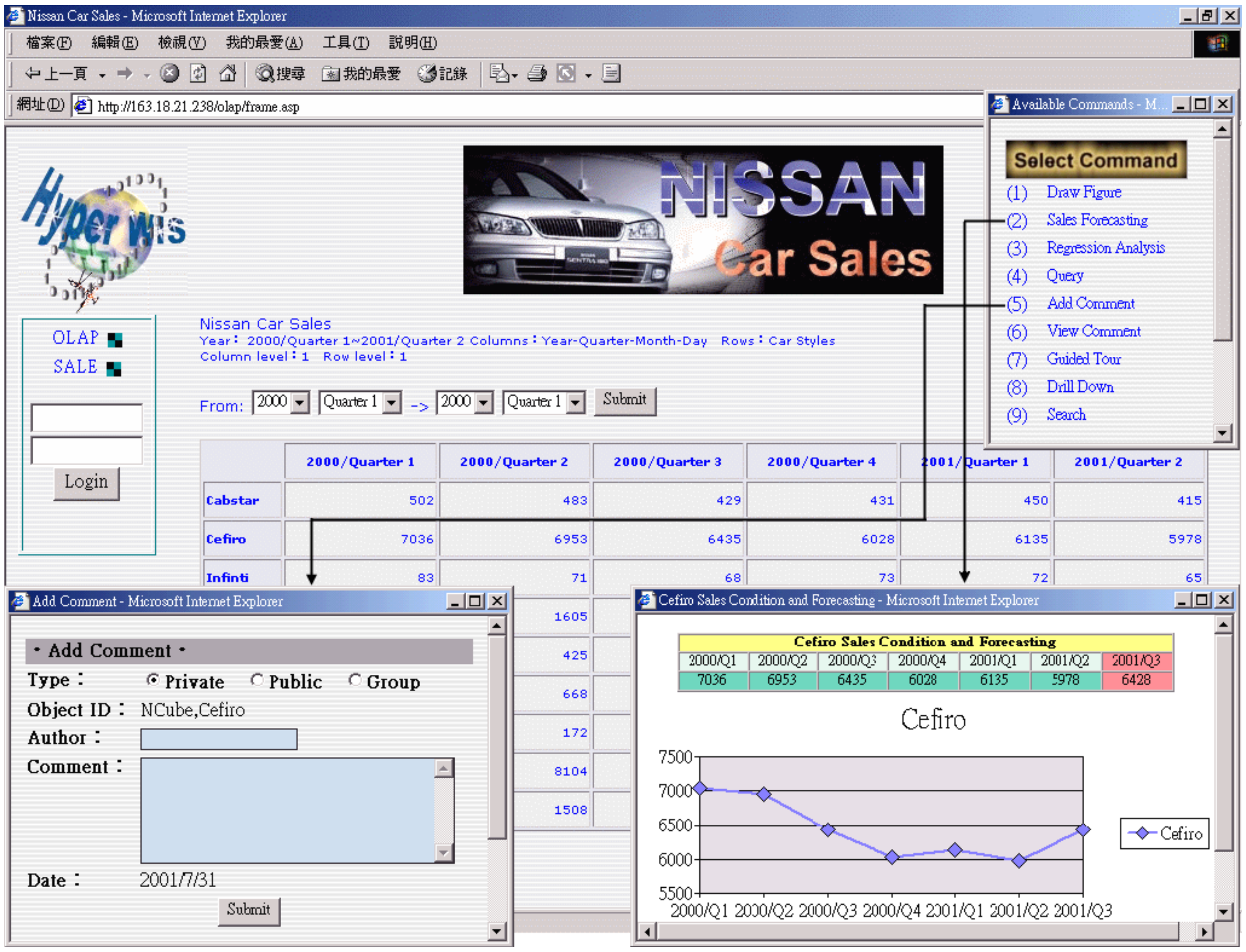
Table 1. Some useful issues and questions that arise for the interface layer.

Issues and Questions
1. How can one present hypermedia and other integrated systems in a useful and intuitive way? How should we display multiple links – in a pop-up dialog box? How could we display semantic types (e.g., semantic relationship between the link's destination and the current Web page [8]) and other metadata (i.e., data about data)? How can we implement hypermedia arrival and departure cues ("rhetoric") within WIS? In general, how do we make hypermedia intuitive and useful?
2. Are current Web modeling languages or hypermedia design methodologies appropriate for designing hypermedia-aware Web information systems?
3. Will users resist information systems converted with simple Web interfaces? Can training counter the danger of user resistance?
4. What kind of interface is appropriate for listing WIS functions - with tree-like structures or pull-down menus? What kind of interface is appropriate for listing WIS commands - in a frame or pop-up dialog box (window) (See Figure 2)?
5. How do we evaluate interfaces in terms of how well they support WIS?

4. Integration Layer

The integration layer concerns inter-application interoperability and integrating information services, such as hypermedia support. The former often requires standard technologies and incorporating Web standards. Integrating information services can bring the functionality provided by specialized systems to general systems on the Web. These can include the services associated with, e.g., hypermedia functionality, open hypermedia, digital libraries, document management, knowledge management, data mining, on-line analytical processing (OLAP), statistical analyses, decision support, and workflow [4, 7]. A digital library is an electronic collection of various fields of information. Document management systems provide various facilities to store, manage, and access either reference to document or whole documents. Data mining tools uncover relationships, patterns, and trends among huge volumes of data, and then transform them to valuable information. OLAP is a technology that performs data analysis using multidimensional and logical views of the data in the data warehouse. Statistical tools are for testing hypothesis, predicting values of variables, grouping variables, and finding variables relationships.

The integration layer should be an open architecture that consists of an open set of inter-operating components. Each component should support a hypermedia service (e.g., navigational, taxonomic, argumentation) or a class of information systems [31]. The integration layer should allow a new component to be easily added to the architecture.



Nissan Car Sales - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 記錄
網址(D) http://163.18.21.238/olap/frame.asp
Hyper WIS
NISSAN
Car Sales
OLAP
SALE
Login
Available Commands - M...
Select Command
(1) Draw Figure
(2) Sales Forecasting
(3) Regression Analysis
(4) Query
(5) Add Comment
(6) View Comment
(7) Guided Tour
(8) Drill Down
(9) Search

Nissan Car Sales
Year : 2000/Quarter 1~2001/Quarter 2 Columns : Year-Quarter-Month-Day Rows : Car Styles
Column level : 1 Row level : 1

From: 2000 Quarter 1 -> 2000 Quarter 1 Submit

	2000/Quarter 1	2000/Quarter 2	2000/Quarter 3	2000/Quarter 4	2001/Quarter 1	2001/Quarter 2
Cabstar	502	483	429	431	450	415
Cefiro	7036	6953	6435	6028	6135	5978
Infinti	83	71	68	73	72	65
		1605				
		425				
		668				
		172				
		8104				
		1508				

Add Comment - Microsoft Internet Explorer

• Add Comment •
Type : [✓ Private] Public Group
Object ID : NCube,Cefiro
Author :
Comment :
Date : 2001/7/31
Submit

Cefiro Sales Condition and Forecasting - Microsoft Internet Explorer

Cefiro Sales Condition and Forecasting						
2000/Q1	2000/Q2	2000/Q3	2000/Q4	2001/Q1	2001/Q2	2001/Q3
7036	6953	6435	6028	6135	5978	6428

Figure 2. The interface of an example system that integrates a navigational hypermedia service into an OLAP tool and a sales forecasting system

Let's take the integration of a navigational hypermedia service into an OLAP tool, a sales forecasting system, a statistical tool, and a document management system as an example. Assume the CEO of a company is browsing the product sales of each car based on sales in last six quarters, which have been enhanced with links which a hypermedia services engine intercepts. The hypermedia engine will send a command to the OLAP tool to retrieve the analysis result. The engine then maps the product name, car names, year/quarters, and car sales to links. After the CEO selects a link representing a car name, the hypermedia engine will infer available commands for the selected object and convert them to links. There could be commands for forecasting sales, analyzing regression relationship between sales and commercial expenditure, adding or retrieving comments, viewing a guided tour, accessing reports, drawing a figure, viewing explanation, accessing metadata, etc. If the CEO selects the "forecasting sales" link, the hypermedia engine will send a command to the sales forecasting system to forecast the sales for the selected car. If the CEO selects the "analyzing regression relationship between sales and commercial expenditure" link, the hypermedia engine will send a command to the statistical tool to perform regression analysis. If the CEO selects the "accessing reports" link, the hypermedia engine will send a command to the document management system to retrieve reports regarding to the past sales status of the selected car. The CEO can also add or retrieve comments on the selected car. Figure 2 shows an example system that integrates a navigational hypermedia service into an OLAP tool and a sales forecasting system.

In the following subsections we discuss hypermedia support and interoperability, closing with a set of issues and questions for the integration layer.

4.1 Hypermedia Support

Many information systems, both on and off the World Wide Web (WWW), do not take full advantage of linking and navigation. This occurs for several reasons. Most designers and developers have a very rudimentary view of hypermedia from the WWW. In part, it has not occurred to them to incorporate more sophisticated hypermedia functionality (i.e., an extremely rich layer of links and other navigation, structuring and annotation functionality [8]). While they have seen fancy Flash animations and well-tailored applets, they and their users have seen few examples of sophisticated hypermedia functionality and do not demand this functionality yet. In part, developers do not have the time to reengineer existing applications, especially when migrating them to the WWW with other projects waiting in the wings. Web standards for some forms of sophisticated hypermedia are quite new and few have seen applications that use them to their full potential yet. Developers also have few tools and techniques for designing and incorporating hypermedia functionality easily. Developers will not do this until it is natural to conceive and easy to implement.

Hypermedia can be viewed as the science of relationship management – structuring, annotating and navigating information through relationships, i.e., links [9]. What benefit do users gain from providing information systems with hypermedia support? Users may find it difficult to understand and take advantage of the myriad of inter-relationships in an information system's knowledge base (among data, metadata, processes, calculated results, and reports). Hypermedia helps by streamlining access to, and providing rich navigational features around related information, thereby increasing user comprehension of information and its context [5, 6]. Augmenting applications with direct access and hypertext structuring, annotation and navigation functionality [8] should result in new ways to: view and manage an application's knowledge and processes conceptually; navigate among items of interest and task stages; enhance an application's knowledge with comments and relationships; and target information displays to individual users and their tasks.

Hypermedia researchers have developed several navigational hypermedia features to help users easily navigate information, and reduce cognitive overhead and disorientation (i.e., becoming "lost in the hyperspace" [15]). This support includes backtracking, history lists, paths, trails, guided tours, overview diagrams, structure-based queries, timestamps, footprints, fisheye views, annotations, bi-directional links, multi-destination ("n-ary") links, etc. [8]. A trail is a logical sequence of links through information spaces [29]. They provide a context for browsing a series of related documents. The trail allows detours from it. Unlike the trail, however, a guide tour prohibits detours. A structure-based query is text string search based on node or link attributes [24]. Overview diagrams are graphical maps for providing a view of the relationships among interconnected nodes. The fisheye view is a tree-like diagram that can show the hypermedia network in a balance of local detail and global context based on the degree of interest [17]. A bi-directional link allows users to initiate the link from both ends of the link. Backtracking is probably the most popular navigation facility, which allows users to go back to the previously visited node. An annotation (comment) facility allows users to add additional information to objects and exchange ideas. The timestamp facility displays the time since the user last visited the node in the hypermedia network.

Hypermedia design methodologies exist for modeling application structure and functionality. The Relationship Management Methodology (RMM) methodology [20] is strongly based on entity-relationship (E-R) abstracts. It comprises seven steps: E-R design, slice design (entity presentation), navigational design, conversion protocol design, user-interface design, run-time behavior design, and construction and testing. The Object-Oriented Hypermedia Design Methodology (OOHDM) [27] is a model-based approach for analyzing the process of building hypermedia applications. It consists of four different phases: conceptual design, navigational design, abstract interface design and implementation. OOHDM focuses in particular on the navigational design and abstract interface design phases. Enhanced Object-Relationship Model (EORM) consists of three frameworks: class, composition and graphical user interface (GUI) [22]. It represents a link as a first class object and provides a reusable link class library to facilitate the mapping of relation schematics into link class. EROM consists of four phases: information analysis, functional analysis, object modeling, and hypermedia mapping. Hypertext Structure Description Language (HSDL) is a schema-based approach for authoring large and structured hypertexts [21]. The core concepts of HSDL contain class schema that represents classes in the target domain and relationships among objects (i.e., links), and instance schema that represents instances of the classes and the relationships. After filling the empty instances with the content, an author can use the compilation program called expanders to map the schema to HTML. Scenario-Based Object-Oriented Hypermedia Design Methodology (SOHDM) is an approach for developing process-oriented hypermedia information systems for supporting organizational processes [23]. It consists of six phases: domain analysis, object modeling, view design, navigation design, implementation design, and construction. Indexed Driven Hypermedia Design Methodology (IHDM) is an approach for developing content-oriented hypermedia systems focusing on marketing or information services [28]. This methodology uses index nodes as the entrance to hypermedia contents and supports two types of links (structural and referential). It consists of five stages: content analysis, node structure design, node specification design, database design, and implementation. Strudel is a graph-oriented approach for modeling semi-structured and tuple-stream data available in the Web site [16]. It provides a query language, StruQL, for specifying the content and structure of a Web site and a template language for specifying a Web site's HTML representation. These current hypermedia design methodologies, however, are for designing stand-alone, retrieval-oriented hypermedia applications instead of full WIS applications.

Many techniques exist for integrating WIS with hypermedia features, basically categorized into standard approaches and middleware approaches. In a standard approach, each component of a WIS should follow some standards therefore to archive interoperability among these complied components. In a middleware approach, we set up some intermediations between the components of WIS so as to provide the WIS with new features. The following discussion takes a middleware approach.

When providing WIS applications with hypermedia support, a facility that automatically infers useful links from information dynamically generated by information systems is required. To provide hypermedia services with minimal changes to WIS, we use wrappers and mapping rules [11, 12, 18]. Mapping rules convert WIS-generated information to hypermedia constructs (nodes, links, and anchors). Mapping rules infer useful links that give users direct access to the WIS' primary functionality, infer metadata about WIS objects, give access to relationships among information objects, and enable hypermedia functionality such as annotation and ad hoc (user-declared) linking. We divide the process of building mapping rules into ten steps: (1) identify users; (2) analyze tasks; (3) identify information objects; (4) identify objects' metainformation; (5) identify relationships among IS objects; (6) identify commands; (7) build metadata database; (8) identify possible hypermedia functionalities; (9) design user interface; (10) build mapping rules [13]. We identify two types of mapping rules: Command Rules and Object Rules. Bieber implemented mapping rules (or bridge laws) using Prolog [3] and in a simpler database format [2]. However, for clarity we discuss mapping rules in terms of functional procedural calls.

1. Command Rules

When a user selects a link representing an IS object, mapping rules in the hypermedia engine will be invoked to infer commands for operating upon the selected object based on the system name and object type. For this process, mapping rules should provide the following functions:

(1) Search the knowledge database for commands accessing various relationships on the selected WIS object.
(2) Map commands to links.
(3) Form a HTML document that includes mapped links and send the document to the Web server.

2. Object Rules

When a user selects a link representing a command, mapping rules in the wrapper will be invoked to execute the command and then infer links from the output generated by the WIS. For this process, mapping rules should provide the following functions:

(1) Map commands to actual WIS commands.
(2) Send actual commands and other parameters to the WIS.
(3) Receive display output from the WIS.
(4) Infer links from the output generated by the WIS.
(5) Create the HTML document with inferred links and send the document to the Web server.

A WIS wrapper is a middleware program that translates and routes messages between the hypermedia engine and the WIS, provides any additional information necessary for mapping links and nodes from WIS output, maps WIS commands to user selections, and passes these user selections to the WIS for processing. Some mapping rules (object rules) reside in and become invoked by the WIS wrapper. A comprehensive WIS wrapper will allow us to integrate an existing IS with few or no changes. Note that a WIS wrapper can be implemented as a Common Gateway Interface (CGI) program, Java servlet, Active Server Pages (ASP) script, Hypertext Preprocessor (PHP), Java Server Pages (JSP), dynamic link library (DLL), etc. The Common Gateway Interface (CGI) is an interface to the Web server that enables users to extend the functionality and capability of the Web server. Java servlets are CGI-like technology that allow user to extend server functionality. ASP, PHP and JSP are server-side scripting languages that allow users to write programs that contain combination of HTML statements, text, and server-side scripts.

The information system oriented hypermedia engine (ISHE) is the core component within the integration architecture. It is like a service manager that provides navigational hypermedia support and coordinates relationship mapping and message passing among different WIS application domains, thus aiding WIS-to-WIS integration [4]. Relationships and other hypermedia functionality are often mapped to or accessible from link anchors added dynamically over the display screen contents. Commands underlying the link anchors can invoke the hypermedia engine to execute appropriate mapping rules. The hypermedia engine provides the following functionality: (1) decodes attributes (e.g., object type, object ID and command) that underlie a link anchor, (2) searches the knowledge base for commands that implement various relationships from the selected IS object based on the system name and object type, (3) maps commands to link anchors, and (4) forms an HTML document that includes the mapped link anchors and sends the document to the WWW server. Note that the master wrapper can be implemented with CGI programs, Java servlets, ASP, PHP, JSP, etc.

Hypermedia engine component, WIS and wrappers are software-intensive systems, and thus we suggest use a consistent and coherent way to model and abstract them. Modeling languages exist for specifying Web applications such as Unified Modeling Language (UML) [14]. System developers should consider how to use a Web application modeling language to specify hypermedia engine component, WIS and wrappers.

Identifying explicit and implicit relationships for system objects forces developers to consider which information users are interested in and then build mapping rules to access this information. Sometimes meaningful relationships cannot be accessed directly from WISs, so developers have to declare those relationships and store them in the knowledge base. Bieber and Yoo [33, 34] identify the following types of relationships for system objects. Each gives the user easy access to some aspect of an object.

(1) Configuration/Aggregation relationship: connects a part to other parts or a whole functionally or structurally.
(2) Membership/Grouping relationship: connects a member of a collection to other members or a whole collection.
(3) Classification relationship: connects an item of interest to its instance or class.
(4) Equivalence relationship: connects instances of the exact same object to a given item.
(5) Similar/Dissimilar relationship: connects all items that share some positive or negative degree of similarity.
(6) Ordering relationship: provides access to objects sequentially related to the object of interest.
(7) Activity relationship: deals with relationships that exist among elements that are involved in some kind of activity.
(8) Intentional relationship: connects an item of interest to the goals, arguments, issues, decisions, opinions, and comments associated with the item.
(9) Influence relationship: provide access to the item over which the item of interest has some kind of influence.
(10) Socio-organizational relationship: connects an item of interest to the position, authority, alliance, role, and communication associated with the item in a social setting or organizational structure.
(11) Temporal relationship: provides access to items temporally related to the item of interest.
(12) Spatial relationship: provides access to objects spatially related to the item of interest.

4.2 Interoperability and Open Hypermedia Systems

One of the obstacles to integration has been the lack of standards to enable interoperability across various Web information system. The current work in providing hypermedia functionality to applications has been explored primarily in the context of open hypermedia systems. The open hypermedia system (OHS) community has been concerned with interoperability between third-party applications and hypermedia servers, and interoperability between open hypermedia systems/servers [25, 32]. It is suggested that a component-based open hypermedia system (CB-OHS) environment should help reduce the effort to build new standards for interoperability and the effort for system developers to comply with these standards [30]. The CB-OHS framework aims to support a wider range of structural services, including navigational, taxonomic, collaborative, argumentation, and spatial [26]. Integrating the kinds of information systems discussed above is to a large extent still unexplored in the CB-OHS research. Some OHS researchers start to expand the CB-OHS architecture to deal with multiple open services [31]. Although the move towards a Common Object Request Broker Architecture (CORBA) compliant CB-OHS approach still cannot resolve the problems in integration of legacy applications and back-end storage interoperability, we can still learn a lesson from the CB-OHS's approach to solving other interoperability problems. However, we believe that wrappers complying with multiple communication middleware should resolve the WIS integration and interoperability problems.

Table 2. Some useful issues and questions that arise for the integration layer.

	Issues and Questions
1.	How can one apply or extend hypermedia design methodologies for modeling information dynamically generated by WIS and mapped by the hypermedia engine?
2.	Designing protocols to accommodate different internal structures of information services is a critical issue to the integration layer. Is a standard backend protocol sufficient? What else do we need?
3.	How can hypermedia service components and information service components communicate with each other completely and efficiently?
4.	Could we apply machine learning algorithms or statistical models to building mapping rules with good inference capability?
5.	How can one determine which information systems will benefit from integrating hypermedia service into them?
6.	What would allow one to freely interchange metadata across Web information systems?
7.	How could one best integrate different hypermedia services (e.g., navigational, spatial, augmentation, collaborative and taxonomic) into different information services?
8.	How does one evaluate the desirability and effectiveness of information services support within WIS?

5. Infrastructure Layer

The infrastructure layer concerns how to use the Internet and Web technologies to support the information services and interoperability and WIS development issues at the integration layer. Many approaches exist for building applications to interface the WWW server and external resources (e.g., databases) such as CGI, Java servlets, ASP, PHP, and JSP, among others. There are also various communication interfaces and middleware to support system components to communicate with each other. The Web technology is changing rapidly. Server-side programming approaches and languages keep emerging. When developing a hypermedia engine or new WIS, the system developer should chose an appropriate network infrastructure, operating system, communication interface, middleware, Web server and server-side programming language to ensure minimal or no cost for adapting to the rapidly changing Web environment.

Table 3. Some useful issues and questions that arise for the infrastructure layer.

	Issues and Questions
1.	Which communication protocols, middleware and server-side programming languages are appropriate for developing the hypermedia engine and wrappers for integrating various information services?
2.	How can we apply or even change Web technology to add more hypermedia and information services functionalities currently not supported in the WWW.
3.	How can one maintain the sustainability and extensibility of information services in WIS?

6. Conclusion

Developing an integrated WIS infrastructure is not just a matter of using up-to-date or popular technologies and standards to build WISs either from scratch or by reengineering existing information systems. It also involves providing user-friendly Web interfaces and easy mechanisms for accessing applications' knowledge and information relationships. We believe that integrating hypermedia services with WIS should go a long way toward making WIS more easily accessible and understandable. This paper aims to use a framework to discuss the benefit of integrating hypermedia into WISs, provide some suggestion, discuss various issues and questions, and call people's attention to this opportunity.

The framework currently only helps developers think more systematically about the issues and questions about integrating information systems into the WWW, and providing hypermedia services. We intend to extend the framework in the following ways: (1) include detailed guidelines for WIS development and building mapping rules, taking appropriate existing hypermedia and Web engineering techniques into account; (2) provide detailed guidelines for presenting hypermedia and other integrated services in a useful and intuitive way and designing an interface appropriate for listing WIS functions; (3) propose a conceptual architecture and the standards for implementing the architecture.

7. Acknowledgements

We gratefully acknowledge funding by the National Science Council (NSC) of Republic of China, by the United Parcel Service, by the NASA JOVE faculty fellowship program, by the New Jersey Center for Multimedia Research, by the National Center for Transportation and Industrial Productivity at the New Jersey Institute of Technology (NJIT), by the New Jersey Department of Transportation, and by the New Jersey Commission of Science and Technology.

8. References

[1] M. Abrams, C. Phanouriou, A.L. Batongbacal, and S.M. Williams, "UIML: an Appliance-independent XML User Interface Language," Computer Networks, Vol. 31, 1999, pp. 1695-1708.

[2] A. Bhaumik, D. Dixit, R. Galnares, M. Tzagarakis, M. Vaitis, M. Bieber, V. Oria, A. Krishna, Q. Lu, F. Aljallad, and L. Zhang, "Towards Hypermedia Support for Database Systems," Proceedings of the 34th Hawaii International Conference on System Sciences, 2001, [CD-ROM].

[3] M. Bieber, "Automating Hypermedia for Decision Support, " Hypermedia, Vol. 4, No.2, 1992, pp. 83-110.

[4] M. Bieber, D. Engelbart, R. Furuta, S.R. Hiltz, J. Noll, J. Preece, E. Stohr, M. Turoff, and B. Van De Walle, "Virtual Community Knowledge Evolution," Proceedings of the 34th Hawaii International Conference on System Sciences, 2001, [CD-ROM].

[5] M. Bieber, and C. Kacmar, "Designing Hypertext Support for Computational Applications," Communication of the ACM, Vol. 38, 1995, pp. 99-107.

[6] M. Bieber, H. Oinas-Kukkonen, and V. Balasubramanian, "Hypertext Functionality," ACM Computing Surveys, 1999.

[7] M. Bieber, R. Galnares, and Q. Lu, "Service Integration for Virtual Communities," Fourth International Workshop on Web Engineering, World Wide Web 10 Conference, Hong Kong, May 2001.

[8] M. Bieber, F. Vitali, H. Ashman, V. Balasubramanian, and H. Oinas-Kukkonen, "Fourth Generation Hypermedia: Some Missing Links for the World Wide Web," International Journal of Human Computer Studies, Vol. 47, 1997, pp. 31-65.

[9] M. Bieber, and J. Yoo, "Hypermedia: A Design Philosophy," ACM Computing Surveys, 1999.

[10] S. Ceri, P. Fraternali, and A. Bongio, "Web Modeling Language (WebML): a Modeling Language for Designing Web Sites," Computer Networks, Vol. 33, 2000, pp. 137-157.

[11] C.M. Chiu, "Reengineering Information Systems with XML," Information Systems Management, Vol. 17, No. 4, 2000, pp. 40-55.

[12] C.M. Chiu, and M. Bieber, "A Dynamically Mapped Open Hypermedia System Framework for Integrating Information Systems," Information and Software Technology, Vol. 43, No. 2, 2001, pp. 75-86.

[13] C.M. Chiu, and M. Bieber, "Toward Hypermedia Support for Information Relationship Management," Journal of Information Science, Vol. 27, No. 2, 2001, pp. 93-100.

[14] J. Conallen, "Modeling Web Application Architectures with UML," Communications of the ACM, Vol. 42, No. 10, 1999, pp. 63-70.

[15] J. Conklin, "Hypertext: An Introduction and Survey," IEEE Computer, Vol. 20, 1987, pp. 17-41.

[16] M. Fernandez, D. Florescu, J. Kang, A. Levy, and D. Suciu, "Catching the Boat with Strudel: Experiences with a Web-site management system," ACM SIGMOD, 1998,.

[17] G.W. Furnas, "Generalized Fisheye Views," Proceedings of the ACM CHI'86 Conference on Human Factors in Computing Systems, 1986, pp.16-23.

[18] R. Galnares, "Augmenting Applications with Hypermedia Functionality and Meta-Information," Dissertation, CIS Department, New Jersey Institute of Technology, 2001.

[19] T. Isakowitz, M. Bieber, and F. Vitali, "Special Section: Web Information Systems," Communications of the ACM, Vol. 41, No. 7, 1998, pp. 78-80.

[20] T. Isakowitz, E. Stohr, and P. Balasubramanian, "RMM: A Methodology for Structuring Hypermedia Design," Communications of the ACM, Vol. 38, No. 8, 1995, pp. 34-44.

[21] M. Kesseler, "A Schema-Based Approach to HTML Authoring," W3 Journal, 1995.

[22] D.B. Lange, "An Object-oriented Design Method for Hypermedia Information Systems," Journal of Organizational Computing & Electronic Commerce, Vol. 6, No. 3, 1996.

[23] H. Lee, C. Lee, and C. Yoo, "A Scenario-based Object-oriented Design Methodology," Information and Management, Vol. 36, 1999, pp. 121-138.

[24] Y.K. Lee, S.J. Yoo, K. Yoon, and P.B. Berra, "Querying Structured Hyperdocuments," Proceedings of the 29th Annual Hawaii International Conference on System Sciences, 1996, pp.155-164.

[25] D.E. Millard, L. Moreau, H.C., Davis, and S. Reich, "FOHM: A Fundamental Open Hypertext Model for Investigating Interoperability between Hypertext Domains," ACM Hypertext'00 Proceedings, ACM Press, 2000, pp. 93-102.

[26] P.J. Nürnberg, J.J. Leggett, and U.K. Wiil, "An Agenda for Open Hypermedia Research," ACM Hypertext'98 Proceedings, ACM Press, 1998, pp. 198-206.

[27] D. Schwabe, G. Rossi, and S. Barbosa, "Systematic Hypermedia Application Design with OOHDM," ACM Hypertext'96 Proceedings, ACM Press, 1996, pp. 116-128.

[28] W. Suh, and H. Lee, "A Methodology for Building Content-oriented hypermedia systems," The Journal of Systems and Software, Vol. 56, 2001, pp. 115-131.

[29] R. Trigg, "Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment," ACM Transactions of Office Information Systems, Vol. 6, No. 4, 1988, pp.398-414.

[30] U.K. Wiil, P.J. Nürnberg, D.L Hichs, and S. Reich, "A Development Environment for Building Component-Based Open Hypermedia Systems," ACM Hypertext'00 Proceedings, ACM Press, 2000, pp. 266-267.

[31] U.K. Wiil, "Multiple Open Sercices in a Structural Computing Environment," Proceedings of the 1st Workshop on Structural Computing, Technical Report, Aarhus University, 1999.

[32] U.K. Wiil, and K. Østerbye, "Using the Flag Taxonomy to Study Hypermedia System Interoperability," ACM Hypertext'98 Proceedings, ACM Press, 1998, pp. 188-197.

[33] J. Yoo, and M. Bieber, "Towards a Relationship Navigation Analysis," Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000, [CD-ROM].

[34] J. Yoo, and M. Bieber, "Finding Linking Opportunities through Relationship-based Analysis," ACM Hypertext'00 Proceedings, ACM Press, 2000, pp. 181-190.