

December 1998

Integrating Third-party Applications and Information Systems Into the World Wide Web

Chao-Min Chiu

National Institute of Technology at Kaohsiung

Rong-Shyan Chen

Rutgers University

Michael Bieber

New Jersey Institute of Technology

Follow this and additional works at: <http://aisel.aisnet.org/amcis1998>

Recommended Citation

Chiu, Chao-Min; Chen, Rong-Shyan; and Bieber, Michael, "Integrating Third-party Applications and Information Systems Into the World Wide Web" (1998). *AMCIS 1998 Proceedings*. 317.

<http://aisel.aisnet.org/amcis1998/317>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISEL). It has been accepted for inclusion in AMCIS 1998 Proceedings by an authorized administrator of AIS Electronic Library (AISEL). For more information, please contact elibrary@aisnet.org.

Integrating Third-party Applications and Information Systems Into the World Wide Web

Chao-Min Chiu

National Institute of Technology at Kaohsiung, Taiwan

Rong-Shyan Chen

Rutgers University

Michael Bieber

Institute of Integrated Systems Research
New Jersey Institute of Technology

Abstract

In this paper we propose a conceptual architecture that integrates with both third-party applications and information systems. We believe that integrating information systems with the Web will go a long way toward making information systems more understandable. We also believe that integrating third-party applications with the Web will lessen the problem that users have to discard applications they use everyday to navigate and publish in the cyberspace.

Introduction

Hypertext is an approach that allows authors to manage information relationships by creating links within or across small chunks of information (nodes or documents). Readers can easily and rapidly browse the nonlinear information network in the order most appropriate to the reader's task by activating links.

The overall goal of this paper is two-fold: (1) to integrate third-party applications with the WWW and (2) to provide hypertext functionality through the WWW to hypertext-unaware dynamically-mapped information systems (DMIS) with minimal changes to DMISs.

We adopt the WWW as the primary infrastructure for system integration because of the following reasons: (a) the WWW seems to be the major interface and development environment in the future, (b) all users will have access to it (through the Internet or Intranets) [BV97], and (c) the WWW architecture provides extensible interfaces (CGI, server-side API, etc.), which have the potential to build domain-specific wrappers to integrate with DMISs.

We call third-party applications display-oriented applications (DOAs). In DOAs authors create document contents and links manually. What benefit do users gain from integrating third-party applications with the WWW? Users can use applications that they use everyday to create documents and link document contents both within and among applications instead of using dedicated Web publish tools. Users can also use DOAs to browse document contents.

DMISs dynamically generate their contents and thus require some *mapping mechanism* to automatically map the generated content to hypertext constructs (nodes, links, and link markers) instead of hypertext links being hard coded over static content. DMISs include spreadsheets, statistical analysis systems, database management systems (DBMS), geographic information systems, expert systems and decision support systems. As we can not expect to change DMISs in any major way, we need to leave them "hypertext-unaware." A wrapper might help by providing a mapping mechanism on behalf of DMISs.

What benefit do users gain from providing information systems with hypertext support? Users can find it difficult to understand and take advantage of the myriad of inter-relationships in a information system's knowledge base (data, processes, calculated results, reports). Hypertext helps by streamlining access to and providing rich navigational features around related information, thereby increasing user comprehension of information and its context [BK95]. Augmenting an information system with hypertext support results in new ways to view and manage the information system's knowledge, by navigating among items of interest and annotating with comments and relationships (links) [BV97].

This paper contributes a conceptual architecture that integrates with both display-oriented applications and dynamically-mapped information systems.

Integrating DOAs with the WWW

In this paper, we propose a super-client and sub-client architecture (see Figure 1) which restructure the way Web clients integrate with the Web server. The main concept is that sub-clients (third-party applications) integrate with the super-client (wrapper administrator) which will then communicate with the Web server on behalf of sub-clients. This architecture also

emphasizes integrating other third-party applications with the Web server. Our goal is to provide a different approach for integrating the World Wide Web with third party applications under all levels of compliance.

We classify display-oriented applications into three categories: **fully Web compliant**, **partially Web compliant**, and **non Web compliant**. Fully Web compliant applications are those that developers have built or modified on purpose to support full Web functionality. Those applications include Netscape, MS Internet Explorer. Partially Web compliant applications are programs that support basic hypertext functionality (i.e., linking) with minor modification (e.g., using a scripting language, macro, etc.) or add-ins. Partially compliant Web Applications include MS-Word with Internet Assistant, MS-Excel with Internet Assistant, etc. Non Web compliant applications are those that are not aware of Web functionality and only invoked by other Web compliant applications to display special types of files. Helper applications belong to this category. Helper applications are programs that are invoked to display specific types of files when browsers encounter unknown file types. Helper applications include WHAM (for sound), Xing Player (for MPEG audio), MPEGPlay (for MPEG video), etc. Based on the classification, the World Wide Web is currently concentrating on integrating with fully Web compliant applications (dedicated Web browsers) and non Web compliant applications (Helper applications).

Integrating DMISs into the WWW

Web database development is a hot new field. Most database applications handle queries and generate HTML from query results without inferring links. Those database applications do not meet our definition of integration with DMISs since they do not infer useful links that give users more direct access to the DMIS's primary functionality, give access to meta-information about DBMS objects, and enable annotation and ad hoc links.

The WWW has radically increased the awareness of the concept of hypertext. The WWW is mainly used to browse predefined HTML files. However, the WWW has the opportunity to integrate hypertext support into information systems. There are many potential approaches for integrating Web servers with information systems: traditional CGI, server API, Java applets, ActiveX technology, Active Server Page, etc. No systematic approach exists, however, for moving an analytical information system to the WWW and giving users direct access to its interrelationships. Chiu [C97] proposed a systematic dynamic-mapping mechanism and a set of guidelines for DMIS integration on the WWW.

Web search engines do perform mapping using simple mapping rules to map the query result to a dynamically-generated Web page with links underlying the "URL address" and "page title" for each hit. The search engine has a simple dynamic mapping mechanism. To augment a DMIS with hypertext functionality, we need a more complicated and generic mapping mechanism.

System Architecture

In this section, we propose a conceptual architecture with nine logical components. This architecture emphasizes the integration of Web servers with DMISs and DOAs, providing hypertext functionality to each. Note that actual architectures may physically implement the functionality in different modules than those shown here. Since the Web server is a normal Web component, we just describe the functionality of other components.

1. Display-Oriented Applications

Fully Web Compliant Applications: They should support full Web functionality. They also will send a message to the Wrapper Administrator to ask for invoking partially compliant applications or non Web compliant applications to display a file that they can not handle.

Partially Web Compliant Applications: A partially compliant application can provide the capability of parsing and rendering HTML documents partially or completely or leave it to the DOA wrapper.

Non Web Compliant Applications: Whenever an application encounters a unknown file type it will ask the wrapper administrator to launch the appropriate non-compliant application to display the information. The DOA wrapper for non Web compliant application might be optional.

2. DOA Wrappers: Sometimes a fully compliant Web application does not need a DOA wrapper. A DOA wrapper can be installed in the Wrapper Administrator or embedded in the appropriate partially Web compliant application. We can also remove the parsing functionality from the DOA wrapper and put the burden on the partially compliant application. Note that our architecture is only conceptual. A DOA wrapper should ignore or substitute the HTML elements that an associated partially compliant application can not render.

3. Wrapper Administrator should provide the following functionality:

Coordinate the message passing among three types of display-oriented applications.

Pass HTML and non-HTML documents to the appropriate DOA wrapper.

Communicate with the Web server on behalf of partially or fully compliant applications.

Invoke appropriate applications based on file types.

Provide the interface for users to install wrappers and register applications associated with file types.

Provide networking capability to communicate with the Web server.

Support HTTP protocol.

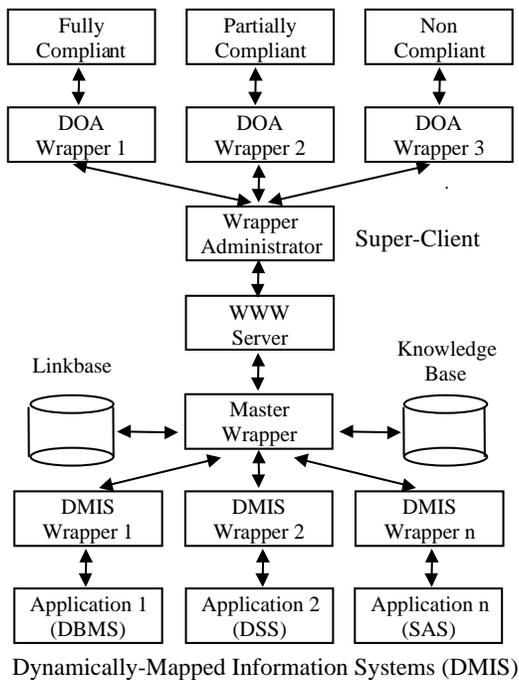
The main purpose of using the wrapper administrator is to make all sub-clients work in an integrated environment. It is inefficient and not flexible to require all partially and fully compliant applications to support networking capability. Building networking capability into the wrapper administrator can simplify the functionality of applications to be integrated.

4. The Master Wrapper coordinates schema mapping and message passing among different DMIS domains, thus aiding DMIS-to-DMIS integration. It also provides the following functionality: (1) decodes attributes (e.g., object type, object ID and

command) underlying the link anchor, (2) searches the knowledge base for commands (e.g., list fields) accessing various relationships on the selected DMIS object based on the object type, (3) maps commands to link anchors, and (4) forms a HTML document that includes mapped link anchors and sends the document to the Web server. Note that the master wrapper can be implemented using CGI scripts, server-side Java, server APIs, etc.

5. A DMIS wrapper translates and routes messages between its DMIS and the Web server. It also provides its DMIS's mapping rules to infer links and nodes from outputs of its DMIS. A DMIS wrapper must map commands sent by the Web browser to actual system commands and invoke its DMIS to execute actual DMIS commands. A comprehensive DMIS wrapper will allow us to integrate an existing DMIS with few or no changes.
6. A DMIS is an application system with which users interact to perform some task, which dynamically produces output content for display.
7. A knowledge base stores commands for accessing various relationships on DMIS objects and information that can not be accessed directly from DMISs (e.g., relationships in E-R diagrams).
8. A linkbase stores user-created annotations (e.g., comments) and ad hoc links.

Display-Oriented Applications (Sub-Clients)



We note that the Web server will serve any DMIS that has an appropriate wrapper (e.g., a CGI program). To integrate with a new DMIS, one has to build a wrapper and store information (e.g., commands) in the knowledge for it. Thus, to provide a DMIS application with hypertext support, the developer has to declare mapping rules in the master wrapper and its DMIS wrapper. The basic concept underlying our proposal is the use of *mapping rules* to automatically create the necessary interface between the WWW and the DMISs used. Note that one set of mapping rules can serve all instances of a DMIS.

Conclusion

The WWW presents a way to integrate hypermedia into information systems and third-party applications. We believe that integrating the WWW with information systems in the business world should constitute a major thrust for the WWW research. This will go a long way toward making information systems more understandable. When reengineering information systems for the WWW, dynamic relationship mapping could prove an effective way to add additional hypermedia links. This should help counteract the danger that new Web applications (especially DMISs) will have no hypertext in them [BV97]. We believe that more third-party applications will be restructured to provide hypertext functionality through the Web. This will lessen the problem that users have to discard applications they use everyday to navigate and publish in the cyberspace. We hope this paper will start people thinking about two mechanisms: (1) mapping rules for better integrating the WWW and analytical information systems and (2) wrapper administrator for better interfacing third-party application and the WWW.

Figure 1. A Conceptual Architecture for Integrating the WWW with Both DMISs and DOAs

Acknowledgments

This research is supported by the New Jersey Center for Multimedia Research, the New Jersey Commission of Science and Technology, and the New Jersey Institute of Technology under Grant 991967.

References

- [BK95] Bieber, M. and Kacmar, C. "Designing Hypertext Support for Computational Applications," *Communication of the ACM*, 38(8), 1995, 99-107.
- [BV97] Bieber, M. and Vitali, F. "Toward Support for Hypermedia on the World Wide Web," *IEEE Computer*, 30(1), 1997, 62-70.
- [C97] Chiu, C.M. *Fusing the World Wide Web and the Open Hypermedia System Technologies*. Ph.D. Dissertation, Rutgers University, Newark NJ 07102, 1997.