Hypertext and Web Engineering

Michael Bieber

CIS Department; New Jersey Center for Multimedia Research, and
National Institute of Transportation and Industrial Productivity,
New Jersey Institute of Technology
University Heights, Newark, NJ 07102 USA
bieber@njit.edu; http://megahertz.njit.edu/~bieber

ABSTRACT

We take a two-stage approach to engineering World Wide Web applications. First a Relationship-Navigation Analysis, analyzes an existing or new application specifically in terms of its intra- and inter-relationships. Second, a dynamic hypermedia engine (DHymE), automatically generates links for each relationship and metaknowledge items at run-time. Links and navigation supplement the application's primary functionality.

KEYWORDS: Hypermedia, World Wide Web, Web Engineering, Software Engineering, Relationship Management, Linking, Complex Interfaces

MOTIVATION

As the World Wide Web and its programming tools mature, we increasingly find analytical applications with content generated instead of hand-created being converted to Web interfaces. This paper concerns application design for all such systems. It also addresses the danger that developers will endow these systems with a paucity of links instead of embellishing them with the rich layer of linking and navigational opportunities which the Web could support [BV97].

Furthermore, developers of these analytical applications often struggle with the need to present complicated information in a way users can best understand it. The ability to explore a piece of information in more detail could help users resolve doubts about or simply better understand both that item, as well as the analysis or display of which it is a part. Users may wish to dig deeper around data values and symbols they see, labels on graphs or user input forms, options in pop-up lists, information users enter as input (before actually submitting it), or even on the menu commands and other controls they can invoke.

Providing links that represent application relationships that give the user access to what he or she wants is one of the main purposes of hypermedia. We take a two-stage approach to engineering applications for the World Wide Web. First the developer performs a Relationship-Navigation Analysis, analyzing an existing or new application specifically in terms of its intra- and inter-relationships. This leads him or her to better understand the application's complexity and richness. Second, a dynamic hypermedia engine (DHymE), automatically generates links for each of these relationships and metaknowledge items at run-time, as well as sophisticated navigation techniques not often found on the Web (e.g., guided tours, overviews, structural query) on top of these links. The links and navigation, as well as annotation features, supplement the application's primary functionality [Bi98].

RELATIONSHIP-NAVIGATION ANALYSIS

The Relationship-Navigation Analysis (RNA) technique has 5 steps:

(1) stakeholder analysis
(2) element analysis
(3) relationship and metaknowledge analysis
(4) navigation analysis
(5) relationship and metaknowledge implementation analysis

RNA helps the developer form a deeper comprehension of the application. This occurs primarily in steps 1-4. Which relationships to implement is decided in the last step.

While quite useful in its current form, we intend to develop the RNA technique further by producing specific guidelines for each step and by reducing the range of options that the developer must consider.

Step 1: Stakeholder Analysis

The purpose of the stakeholder analysis is to identify the application's audience. Knowing who will be interested in an application helps the developer broadly determine the entire range of important elements and relationships, and then to focus on these specifically. Especially those applications with public Web access have a much broader range of stakeholders than many imagine. Many developers, in fact, find this the most enlightening part of the RNA. The developer also should identify and understand the tasks each type of user will want to perform within the application. These will help the developer focus on specific areas during the RNA steps that follow.

HyperText 98 Pittsburgh PA USA
Copyright ACM 1998 0-89791-972--6/98/ 6...$5.00

Step 2: Element Analysis

Here the developer lists all the potential elements of interest in the application. At one level these include all types of items displayed in any on-line display (information screens, forms, documents, and any other type of display), as well as the screens, forms and documents themselves. The easiest way to start is to examine each screen (or mock-up) and identify each value and label it contains. Note that developer should identify kinds or classes of elements, not individual instances. The relationship types we discuss in step 3 all are for specific kinds of elements.

Step 3: Relationship Analysis

Relationship analysis concerns inter-relationships, intra-relationships and metaknowledge. The developer should consider each element of interest identified in the prior step in terms of each of the following general kinds of relationships, for each group of stakeholders. Certain relationships will be useful to only certain stakeholders, and the hypermedia engine will filter these. Relationships can lead to information inside and outside the application. Developers should exercise creativity. Only in step 5 should they consider which to implement.

RNA currently helps developers identify the following types of relationships and metaknowledge within an application: schema, process, operation, structural, descriptive, parametric, statistical, collaborative and ordering relationships. [Bi98] gives more details for each. [BV97] shows how several of these general relationship types can supplement an on-line sales invoice. [Bi98] shows how they can supplement a mathematical modeling decision support system.

Step 4: Navigation Analysis

Here the developer decides which hypermedia navigation features (e.g., trails, overviews, structural query) best serve stakeholders' needs.

Step 5: Relationship and Navigation Implementation Analysis

Here the developer decides which relationships and navigation to implement, considering the costs and benefits of implementing and displaying each. The developer then writes a mapping rule (using our specified format) for each relationship to be implemented. Mapping rules specify the commands or algorithms for finding the endpoint of each relationship.

DHYME (DYNAMIC HYPERMEDIA ENGINE)

The DHymE hypermedia engine executes separately from the target application. We write a wrapper program for each application to integrate it into our engine architecture. Applications or their wrappers then connect to DHymE through a Web proxy server. DHymE intercepts all messages passing between the application and the user interface, and uses the mapping rules specified above to map each appropriate element of the message to a hypermedia node or anchor. Our Web browser wrapper integrates these anchors into the document being displayed and passes it through the proxy server to the user's Web browser. When the user selects an anchor, the browser wrapper passes it to DHymE, which returns a list of possible links (one for each appropriate relationship as determined by the mapping rules). If the user selects a normal application command (mapped to an operation link), DHymE passes the command on to the application for processing. If the user selects a hypermedia engine link (e.g., to create an annotation), DHymE processes it entirely. If the user selects a supplemental schema, process, operation, structural, descriptive, information or occurrence relationship, DHymE infers the appropriate application commands, meta-application operations (e.g., at the operating systems level or schema level) or hypermedia engine operations that will produce the desired information. If the user selects a user-created annotation, DHymE retrieves it. Thus DHymE automatically provides all hypermedia linking (as well as navigation) to applications, which remain hypermedia-unaware and in fact often entirely unchanged. We currently are integrating several applications with DHymE, automatically giving each a Web interface or supplementing its existing Web interface: a personnel requisition tracking system, a relational database management system, a mathematical model management system, a transportation spreadsheet analysis system and a multi-criteria decision support analysis tool. [Bi98] describes these ideas and an older, non-Web prototype of DHymE in more detail.

CONCLUSION

We hope that our most enduring contribution is successfully convincing developers of Web applications (both new and transported from other computer environments) to take full advantage of linking in their applications. Time and time again designers have told us that RNA has shown them links they never imagined in their own applications. Identifying these is the necessary first step towards implementation. Implemented thoughtfully, the Web's linking and navigation facilities can go a long way to reducing the complexity of applications for users. RNA gives developers a tool for identifying opportunities for supplemental linking within applications. The DHymE hypermedia engine automatically generates these links, with little or no change to the application.

ACKNOWLEDGMENTS

Funding for this research came from the NASA JOVE program, the NJ Center for Multimedia Research, the National Center for Transportation and Industrial Productivity at NJIT and NJ Department of Transportation.

REFERENCES

[Bi98] BIEBER, M., Supplementing Applications with Hypermedia, under submission, 1998.

[BV97] BIEBER, M. and VITALI, F. (1997). Toward support for hypermedia on the World Wide Web. IEEE Computer 30(1), 1997, 62-70.

[BVA97] BIEBER, M., VITALI, F., ASHMAN, H., BALASUBRAMANIAN V., and OINAS-KUKKONEN, H. Fourth generation hypermedia: some missing links for the World Wide Web. International Journal of Human Computer Studies 47, 1997, 31-65.