



Developing Hypermedia Applications with Methods and Patterns

Gustavo Rossi, Fernando Daniel Lyardet

*LIFIA - Facultad de Informática, UNLP Web: <http://www-lifia.info.unlp.edu.ar/>
Calle 50 y 115 Primer Piso, La Plata,
Buenos Aires, (1900), Argentina
Tel/Fax: +54 221 4228252
Email: gustavo@sol.info.unlp.edu.ar, fer@sol.info.unlp.edu.ar
Web: <http://www.lifia.info.unlp.edu.ar/~gustavo>, <http://www.lifia.info.unlp.edu.ar/~fer>*

Daniel Schwabe

*PUC-Rio
Dept. de Informatica Web: <http://www-nt.inf.puc-rio.br/>
R. M. de S. Vicente 225, Rio de Janeiro, RJ 22453-900, Brazil
Tel: +55-21-512 8045/274 4449. Fax: +55-21-259 2232
Email: schwabe@inf.puc-rio.br
Web: <http://www-di.inf.puc-rio.br/~schwabe/>*

Abstract: In this paper we argue that hypermedia applications can and should be built using software engineering practices. We briefly motivate our point by surveying the state of the art of the most representative family of hypermedia applications: web-based applications. We next discuss why design methods improve the development process and briefly describe the OOHDM approach. Next, we analyze the rationale for recording design experience and present hypermedia patterns; some outstanding patterns are briefly mentioned. Finally, the need for a hypermedia design repository is discussed together with our work in this direction

General Terms: Design, Hypermedia, Patterns, Web Engineering

Additional Key Words and Phrases: Web Information Systems

Motivation

The increasing popularity of the World Wide Web as a host to different kinds of applications and the rapid evolution of Web-based software from fancy and amateur Web-sites to critical corporate systems have pushed hypermedia techniques further than ever before. This new generation of applications that incorporate hypermedia functionality into complex transactional systems should not suffer the effects of the well-known software crisis [Pressman 1998]. They evolve continuously; new releases are needed with tight schedules as they can influence the core of the company's business (as for example in e-commerce applications). This brief analysis leads us to the first important conclusion:

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept, ACM Inc., fax +1 (212) 869-0481, or permissions@acm.org

1. Complex hypermedia applications (like Web applications) *must* be designed using software engineering techniques. However, hypermedia (and Web) applications possess specific characteristics that make them different from conventional software. First, since building their navigation architecture is critical, this design activity (navigation design) should be explicitly addressed. Besides, hypermedia applications being highly interactive, user interface aspects should also be recorded using some design primitives. Unfortunately, state of the art software design notations and methods, such as the Unified Modeling Language and their associated process approaches [UML] fail to consider these particular characteristics of hypermedia applications. This leads us to conclude that:
2. There is a need for *new* methods that address the previously mentioned design dimensions. Those methods should be easy to integrate with existing ones, to conform to existing standards.

Finally, since this market imposes short development cycles, time has become a critical aspect for deploying effective applications, and therefore design reuse must be emphasized to avoid development from scratch every time. Design experience is now an essential aspect for improving development times, which leads to the last fact of this motivation:

3. Design experience should be recorded using a formalism that makes it easy to convey and share by designers.

In the rest of this paper, we discuss our approach for developing hypermedia applications. It combines the use of an object-oriented method (OOHDM) with a growing set of hypermedia patterns that encapsulates our (and others') design experience with this kind of application.

The Role of Design Methods

Many design methods have been devised for the hypermedia domain [Garzotto 1993], [Isakowitz 1995], [Schwabe 1998]. They provide abstract, environment-independent constructs that help to describe the problem domain in such a way that hypermedia components (nodes, links, and indexes) can be then implemented using the tools provided by a run-time setting.

What is common to all these methods is the view that hypermedia development is as complex as general-purpose software development. OOHDM [Schwabe 1998] for example divides the design space in four activities: conceptual design, navigation design, abstract interface design and implementation. In each activity a design model is built or improved. OOHDM provides well-known object-oriented constructs such as classes, objects, relationships, use scenarios, etc. to specify the different models.

During conceptual design, an (object-oriented) model of the problem domain is built. This model may contain complex object interactions and behaviors as is now common in Web applications. Reusing standard components is straightforward in this activity as the underlying (object) model supports this kind of reuse. Navigation design aims at specifying those components that the user will explore in the application; nodes and links are specified using a view definition language [Schwabe 1998] that allows customizing the conceptual model to different user's profiles.

High level navigation constructs (such as navigational contexts, composite nodes, hierarchical indexes, etc) can be defined using a set of base classes that may be further specialized. OOHDM provides a set of simple design primitives for specifying both the static and dynamic aspects of the interface. These design models can be straightforwardly mapped into an implementation environment. When this environment is itself object-oriented, the translation is even more direct. Simple heuristics exist (see [Schwabe 1998] for example) for deploying OOHDM models using non object-oriented tools (HTML, Toolbook, etc). OOHDM has also proved to be a good design framework for extending legacy applications with hypermedia technology. When such applications have been designed with object-oriented techniques the extension is straightforward as the application is considered as the conceptual model in the OOHDM process model; if the application has been designed using conventional techniques

(such as relational databases), object wrappers can be easily used to adapt them to the object-oriented style [Gamma 1995].

We have used OOHDM for implementing applications in different domains and implementation settings [Portinari 1996]; we are also aware of dozens of implementations using other design methods. We can draw some clear conclusions: first, viewing the hypermedia development enterprise with an engineering view not only helps to understand and document design decisions (thus making maintenance and evolution easier), but also improves communication among different stakeholders (such as managers and final users). However, if we want to improve design reuse we need higher level abstractions both to document complex design decisions and to record successful solutions to recurrent problems.

Capturing Design Experience with Patterns

Design Patterns are a powerful vehicle for recording and conveying design experience. They have been introduced by C. Alexander [Alexander 1977] in the area of urban planning, and they are being widely used in object-oriented software design [Gamma 1995]. A design pattern describes a recurrent problem and an abstract solution to this problem in such a way that this solution can be applied to different instantiations of the same problem; patterns also describe the rationale and trade-offs when the proposed solution is used.

Hypermedia is a rich field for discovering and applying patterns. The OOHDM design framework has led us to classify hypermedia patterns as architectural, navigation and interface patterns. Architectural patterns address common problems related with the connection of databases or legacy software with navigation front-end (as commonly found in Web applications). Navigation and interface patterns [Rossi 1997] are quite similar to Alexander's patterns as they show how to build (virtual) navigable information spaces. They record the collective experience of the hypertext community using a template similar to the one in [Gamma 1995], and are grouped in a growing and increasingly dense catalogue.

Navigation patterns show how to push further the simple metaphor of nodes and links and go a step beyond the usual naive solutions to a design problem. For example, while inexperienced designers will only look for links in strong semantic relationships, the "Set-based Navigation" pattern shows when it is useful to connect nodes opportunistically (for example when they belong to the same set, by virtue of some property relevant to the task the user is trying to perform). The "Node in Context" pattern meanwhile shows how to address the situation when the same node appears in multiple sets, allowing different representations depending on the set within which the node is being accessed.

Interface patterns deal with building more than "fancy" interfaces. For example, "Information on Demand" describes how to organize interface objects when we want to avoid cognitive overhead or when a node has many information items to be perceived. Discovering new patterns is a rewarding activity as it helps to leverage our experience, increasing the level of discourse and it helps us reason on our own (and others') designs. Pattern languages for user interface and interaction design (i.e. set of patterns that address the whole design space) are being now developed [Tidwell 1998].

Patterns are not tied to a particular design method; we can use our pattern catalogue when developing hypermedia applications, even if we are not using OOHDM. Moreover, patterns are powerful even if we do not use any method at all to describe hypermedia applications. We are now improving this catalogue by interacting with designers in different areas inside the hypermedia domain. A first step in this direction can be found in [Lowe 1999].

Concluding Remarks and Further Work

In this paper we have briefly addressed the problem of designing hypermedia applications with a software engineering view. It is now recognized that hypermedia design is critical in new areas such as electronic commerce, digital libraries, etc. However design methods are not by themselves a silver bullet. As hypermedia and interaction design patterns catalogues grow, they can be used synergistically with design methods to improve the development process by providing a convenient way to reuse designs and/or components. However, the interaction among patterns and methods is far from trivial. We need to study how to integrate patterns as primitives of new methods; we also need to define a systematic strategy for pattern mining as well as formal, unambiguous notations to record them.

While the area of design methods, and related web engineering initiatives (such as WCML [Gellersen 1997]), has been active in recent years, there are still several problem areas that have been largely untouched so far, that need to be addressed. For example, collaborative and participatory design, design of electronic commerce applications, application frameworks for hypermedia, etc. Another area of research and development are the computer-aided environments for supporting the entire life cycle of hypermedia application design and implementation. Ideally, such environments should be as independent as possible of the run-time environment and should provide tools for "translating" abstract designs into a running hypermedia applications. We are now working in the construction of a set of design templates that encapsulate hypermedia patterns and that can be used in the context of design environments.

References

- [**Alexander 1977**] Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language*, Oxford University Press, New York, 1977.
- [**Gamma 1995**] Erich Gamma, Ralph Johnson, Richard Helm, John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.
- [**Garzotto 1993**] Franco Garzotto, Paulo Paolini, and Daniel Schwabe. "HDM - A Model-Based Approach to Hypertext Application Design" in *ACM Transactions on Information Systems (TOIS)* 11(1), 1-26, 1993.
- [**Gellersen 1997**] Hans-Werner Gellersen, Robert Wicke, Martin Gaedke. "WebCompostion: An Object-Oriented Support System for the Web Engineering Lifecycle" in *Computer Networks and ISDN Systems* 29, Special Issue on the Sixth International World Wide Web Conference, Santa Clara, CA, 1429-1437, 1997.
- [**Isakowitz 1995**] Tom-s Isakowitz, Edward Stohr, and P. Balasubramanian. "RMM: A Methodology for Structuring Hypermedia Design" in *Communications of the ACM (CACM)*, 38(8), 34-44, August 1995.
- [**Lowe 1999**] David Lowe, Jocelyne Nanard, Daniel Schwabe, and Gustavo Rossi (co-chairs). "Second International Workshop on Hypermedia Development. Hypermedia Patterns" in *ACM Hypertext '99*, Darmstadt, Germany, [Online: <http://ise.ee.uts.edu.au/hypdev/>], February 1999.
- [**Portinari 1999**] Portinari: <http://www.portinari.org.br/>, Embratel: <http://www.embratel.com.br>, Sebrae: <http://www.sebrae.com.br>
- [**Pressman 1998**] Roger S. Pressman, R. "Can Internet-based Applications Be Engineered?" in *IEEE Software*, September/October IEEE Press, 104-110, 1998.
- [**Rossi 1997**] Gustavo Rossi, Daniel Schwabe, and Alejandra Garrido. "Design Reuse in Hypermedia Applications Development" in *Proceedings of ACM Hypertext'97*, Southampton, UK, 57-66, September 1997.

[Schwabe 1998] Daniel Schwabe and Gustavo Rossi. "An Object-Oriented Method for Developing Web-Based Applications" in Theory and Practice of Object and Systems (TAPOS) 4(4), 207-225, October 1998.

[Tidwell 1998] Jennifer Tidwell. "Interaction design patterns" in Patterns Languages of Programs (PLoP '98), Allerton Conference Center, Urbana Champaign, IL, [Online: http://www.mit.edu/~jtidwell/interaction_patterns.html], 1998.

[UML 1999] The Unified Modeling Language: <http://www.uml.rational.com/uml>