"Modeling-by-Patterns" of Web Applications

Franca Garzotto[1], Paolo Paolini[1,2], Davide Bolchini[2], and Sara Valenti[1]

[1] Politecnico di Milano, Milano, Italy
{garzotto, paolini, valenti}@elet.polimi.it
[2] Università della Svizzera Italiana, Lugano, Switzerland
davide.bolchini@lu.unisi.ch

Abstract. "A pattern ... describes a *problem* which occurs over and over again in our environment, and then describes the core of the *solution* to that problem, in such a way that you can use this solution a million times over" [1]. The possible benefits of using design patterns for Web applications are clear. They help **fill the gap between requirements specification and conceptual modeling**. They support **conceptual modeling-by-reuse**, i.e. design by adapting and combining already-proven solutions to new problems. They support **conceptual modeling-in-the-very-large**, i.e. the specification of the general features of an application, ignoring the details. This paper describes relevant issues about design patterns for the Web and illustrates an initiative of ACM SIGWEB (the ACM Special Interest Group on Hypertext, Hypermedia, and the Web). The initiative aims, with the contribution of researchers and professionals of different communities, to build an on-line repository for Web design patterns.

1 Introduction

There is a growing interest in design patterns, following the original definition by architect Christopher Alexander [1]. In this paper we focus particularly on the notion of "pattern" as applied to the process of designing a Web application at the conceptual level.

Patterns are already very popular in the software engineering community, where they are used to record the design experience of expert programmers, making that experience reusable by other, less experienced, designers [2, 6, 8, 9, 19, 21, 27]. Only recently have investigations studied the applicability and utility of design patterns to the field of hypermedia in general, and to the Web in particular [3, 7, 17, 22, 25, 26].

It can be interesting to analyze how design patterns modify the traditional approach to conceptual modeling. Conceptual modeling allows describing the general features of an application, abstracting from implementation details [5]. Conceptual modeling methodologies in general, however, offer little guidance on how to match modeling solutions to problems. Design patterns, instead, explicitly match problems with solutions, therefore (at least in principle) they are easier to use and/or more effective. Design patterns, in addition, support

conceptual modeling by-reuse: i.e. the reuse of design solutions for several similar problems [13, 22, 25]. Finally, design patterns, support *conceptual modeling in-the-very-large*. They allow designers to specify very general properties of an application schema, and, therefore, they allow a more "abstract" design with respect to traditional conceptual modeling.

Despite the potential and popularity of design patterns, only a few effectively usable design patterns have been proposed so far for the Web. This lack of available design patterns is a serious drawback, since design patterns are mainly useful if they are widely shared and circulated across several communities. Just to make an obvious example, *navigation* design patterns (originating within the hypertext community), cannot be completely independent from *interface* design patterns (originating within the visual design community). Both types of design patterns are more effective if they are shared across the respective originating communities.

The main benefits we expect from widespread adoption of design patterns for the Web include the following.

- *Quality of design.* Assume that a large collection of "good" and tested design patterns is available. We expect by reusing patterns that an inexperienced designer would produce a better design with respect to modeling from scratch.
- *Time and cost of design.* For similar reasons we expect that an inexperienced designer, using a library of patterns, would be able to complete her design with less time and less cost than would otherwise be required.
- *Time and cost of implementation.* We expect that implementation strategies for well-known design patterns are widely available. We may therefore expect that the implementation of an application designed by composing design patterns will be easier than implementing an application designed from scratch. Additionally, in the future we may expect that toolboxes for implementation environments, such as JWeb [4], Autoweb [24], and Araneus [20], will provide automatic support for a number of different design patterns.

The rest of this paper is organized as follows. In Section 2 we describe a way to define design patterns and discuss conceptual issues involved with the definition of a pattern. In Section 3 we introduce examples of design patterns. In Section 4 we state our conclusions and describe the ACM-SIGWEB initiative for the development of a design-pattern repository.

2 Defining Design Patterns

The definition schema for a pattern includes the following:

- The pattern *name*, used to unambiguously identify the pattern itself.
- The description of the design *problem* addressed by the pattern. It describes the context or scenario in which the pattern can be applied, and the relevant issues.

- The description of the *solution* proposed by the pattern, possibly with a number of *variants*. Each variant may describe a slightly different solution, possibly suitable for a variant of the problem addressed by the pattern. The use of variants helps keep down the number of distinct patterns.
- A list of *related patterns* (optional). Relationships among patterns can be of different natures. Sometimes a problem (and the corresponding solution) addressed by a pattern can be regarded as a specialization, generalization, or combination of problems and solutions addressed by other patterns. One pattern might address a problem addressed by another pattern from a different perspective (e.g. the same problem investigated as a navigation or an interface issue). Several other relationships are also possible.
- The *discussion* of any aspect which can be useful for better understanding the pattern, also including possible restrictions of applicability and conditions for effective use.
- *Examples* that show actual uses of the pattern. Some designers claim that for each "real" pattern there should be at least three examples of usage by someone who is not the proposer of the pattern itself. The idea behind this apparently strange requirement is that a pattern is such only if it is proved to be usable (i.e. is actually used) by several people.

In the remainder of this paper we use this simple schema to describe design patterns.

A design *model* has a different aim than a design *pattern*. The purpose of a conceptual model is to provide a vocabulary of terms and concepts that can be used to describe problems and/or solutions of design. It is not the purpose of a model to address specific problems, and even less to propose solutions for them. Drawing an analogy with linguistics, a conceptual model is analogous to a language, while design patterns are analogous to rhetorical figures, which are predefined templates of language usages, suited particularly to specific problems.

Therefore, design models provide the vocabulary to describe design patterns, but should not influence (much less determine) the very essence of the patterns. The *concept* behind a design pattern should be independent from any model,[1] and, in principle, we should be able to describe any pattern using the primitives of any design model.

In this paper we use the Web design model HDM, the Hypermedia Design Model [10–12], but other models (such as OOHDM [28] or RMM [18]) could also be used. We now discuss some general issues concerning Web design patterns.

2.1 Design Scope.

Design activity for any application, whether involving patterns or not, may concern different aspects [5]. The first aspect, relevant to Web applications, is related to *structure*. Structural design can be defined as the activity of specifying types of information items and how they are organized.

[1] Continuing our analogy, a rhetorical figure is independent from the language used to implement it.

The second aspect is related to *navigation* across the different structures. Navigation design can be defined as the activity of specifying connection paths across information items and dynamic rules governing these paths.

The third relevant aspect is *presentation/interaction*. Presentation/interaction design is the activity of organizing multimedia communication (visual + audio) and the interaction of the user. We could also distinguish between *conceptual presentation/interaction*, dealing with the general aspects (e.g. clustering of information items, hierarchy of items), and *concrete presentation/interaction*, dealing with specific aspects (e.g. graphics, colors, shapes, icons, fonts).

If ideally each design pattern should be concerned with only one of the above aspects, we should also admit that, sometimes, the usefulness (or the strength, or the beauty, ...) of a pattern stems from considering several aspects simultaneously and from their relative interplay.

2.2 Level of Abstraction

A design problem can be dealt with at different levels of abstraction. For a museum information point, for example, we could say that the problem is "to highlight the current main exhibitions for the casual visitor of the site." Alternatively, at a different level of abstraction, we could say that the problem is "to build an effective way to describe a small set of paintings in an engaging manner for the general public." Yet at another level we could say that the problem is to "create an audio-visual tour with automatic transition from one page to the next." The third way of specifying the problem could be considered a refinement of the second, which in turn could be considered a refinement of the first.

Design patterns can be useful or needed at different levels of abstraction, correspondingly also to the different types of designers. Interrelationships among patterns at different levels should be also identified and described.

2.3 Development Method

Some authors claim that patterns are discovered, not invented [25, 28], implicitly suggesting that not only the validation but also the identification of a pattern is largely founded on the frequency of use of a given design.

This approach is only partially correct. First, the pure analysis of existing solutions does not detect their relationships to application problems (which are usually unknown and must be arbitrarily guessed). In addition a pure frequency analysis could lead to deviating results. In a frequency analysis of the navigation structure of 42 museum Web sites (selected from the *Best of the Web* nomination list of "Museums and the Web" in 1998 [7]), we detected a number of "bad" navigation patterns. The term "bad" in this case is related to the violation of some elementary usability criteria [14–16, 23].

We are therefore convinced that frequency of use is useful for a posteriori validation, but it can not deliver, for the time being, a reliable development method. Until the field matures, design patterns for the Web should come from

a conscious activity of expert designers trying to condense their experience, and the experience of other designers, into viable problem/solution pairs.

3 Examples of Design Patterns for the Web

In this section we give definitions of several design pattern examples. Since lack of space prevents us from explaining HDM in depth, we do not use precise modeling primitives to describe the patterns; the negative effect upon the precision of the definition is balanced by the enhanced readability.

The goal is to convince the reader that a design pattern can condense a number of useful hints and notions, providing a contribution to the quality and efficiency of design. We remind the reader that design patterns are devices for improving the design process, mainly aiming at inexperienced designers, not as a way for experienced designers to discover new very advanced features.

The examples we discuss in this section are related to *collections* [11, 12] in the terminology of HDM. A collection is a set of strongly correlated items called *members*. Collection members can be pages representing application-domain objects, or can themselves be other collections (*nested collections*). A page acts as a collection entry point: information items associated with the entry point are organized in a *collection center*, according to HDM (and this will be elaborated later). Collections may correspond to "obvious" ways of organizing the members (e.g. "all the paintings," "all the sculptures"), they may satisfy domain-oriented criteria (e.g. "works by technique"), or they may correspond to subjective criteria (e.g. "the masterpieces" or "the preferred ones"). The design of a collection involves a number of different decisions, involving structure, navigation, and presentation (see references).

Pattern Name. Index Navigation
Problem. To provide fast access to a group of objects for users who are interested in one or more of them and are able to make a choice.[2]
Solution. The core solution consists of defining links from the entry point of the collection (the *collection center* in HDM) to each member, and from each member to the entry point. To speed up navigation, a variant to this core solution consists of including in each member links to all other members of the collection, so that the user can jump directly from one member to another without returning to the collection entry point.
Related Patterns. Collection Center, Hybrid Navigation
Discussion. The main scope of this pattern is navigation. In its basic formulation, it captures one of the simplest and most frequent design solutions for navigating within collections, adopted in almost all Web applications. The variant is less frequent, although very effective to speed up the navigation process. The

[2] This requirement is less obvious that it may appear at first sight; in many situations, in fact, the reader may not be able to make a choice out of list of items. She may not know the application subject well enough; she may not be able to identify the items in the list; she may not have a specific goal in mind, etc.

main advantage of the variant is that a number of navigation steps are skipped if several members are needed. The disadvantage of the variant is mainly related to presentation:[3] displaying links to other members occupies precious layout space.
Examples. Navigation "by-index" is the main paradigm of the Web site of the Louvre museum (www.louvre.fr; May 18, 1999), where most of the pages representing art objects are organized in nested collections, according to different criteria. Figure 1 shows the entry point of the top-level collection, where users can navigate "by-index" to each specific sub-collection. Figure 2 shows the entry point (center) of the sub-collection "New Kingdom of Egyptian Antiquities," where each artwork from Egypt dated between 1550 and 1069 BC can be accessed directly. From each artwork, the only way to access another page within the same collection is to return to the center.

Fig. 1. The center of a collection of collections, in the Louvre Web site (`www.louvre.fr`)

An example of the variant is found in the Web site of the National Gallery of London (`www.nationalgallery.org.uk`; May 18, 1999). Each painting in a collection can be accessed directly from the entry point page that introduces the collection (e.g. "Sainsburg Wing" in Figure 3) and from each member (e.g. "The Doge Leonardo Loredan" in Figure 4). Each page related to a painting has links to the other members, represented by thumbnail pictures, title, and author.
Pattern Name. Guided Tour Navigation
Problem. To provide "easy-to-use" access to a small group of objects, assuming

[3] Even this simple example shows that it is difficult to discuss one aspect, navigation, without discussing also other aspects, e.g. presentation.

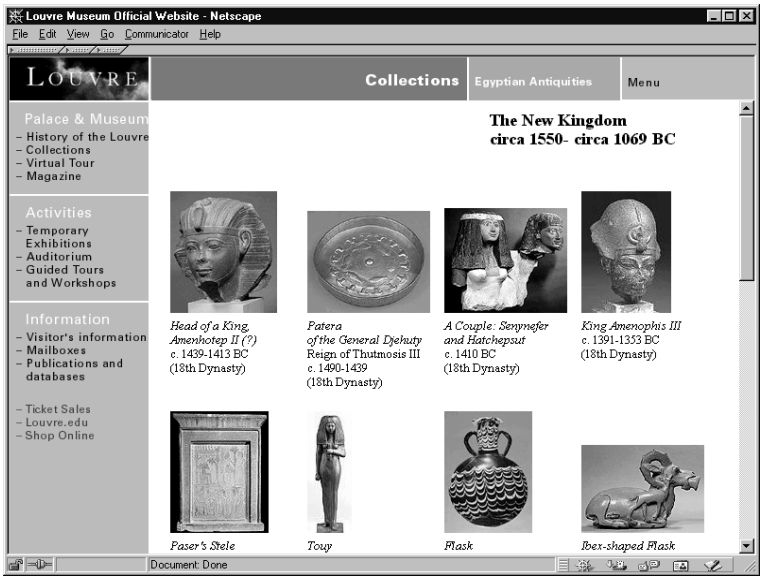

Fig. 2. The center of the collection "Egyptian artworks dated between 1550 and 1069 BC," in the Louvre Web site (`www.louvre.fr`)

Fig. 3. The center of the collection "Sainsbury Wing" in the National Gallery of London Web site (`www.nationalgallery.org.uk`)

Fig. 4. A member of the collection "Sainsbury Wing" in the National Gallery of London Web site (`www.nationalgallery.org.uk`) showing links to all other members of the collection

the user has no reason (or is unable) to select one of them.

Solution. The solution consists of identifying an order among the collection members, and creating sequential links among them. Links can be one-way or two-way (forward or backward). A variant is the circular guided tour, where the last member is linked to the first.

To start the navigation, the page corresponding to the collection center must link to the first member. In order to allow return to the collection center several variants are available: establishing a link from every member, the last member, or the first member to the collection center. The circular variant is useful in the last case, in order to avoid the need for the user to scan the whole collection backwards up to the first member. In order to improve usability [14, 15] it is advisable to support user's orientation, i.e. to include in each member some perceivable visual cues about the current navigation status. Examples of such cues are an indication of the name of the current collection, the position of the current member, and the total number of members.

Related patterns. Collection Center, Hybrid Navigation

Discussion. This is a navigational pattern. It is suitable for "naive" users (with little knowledge of the application domain) or for first-time users (who need to get acquainted with the content of the application) or "couch-potato" users (who want to get an easy ride around, rather than engaging in free navigation). The main disadvantage of a guided tour is that its members must be traversed sequentially: it may be bothersome if the user wants to rapidly access a member

of his or her choice, or if there are many members.
Examples. Guided tours are quite common, and we have found interesting examples in commercial sites of car companies. Though complex and highly structured in most cases, these applications are mainly intended to provide the user an easy, flamboyant presentation of new car models. For example, see Opel (`www.opel.com`), Volkswagen (`www.volkswagen.com`), Renault (`www.renault.com`), BMW (`www.bmw.com`), Porsche (`www.porshe.com`), Citroen (`www.citroen.com`), Mercedes (`www.mercedes.com`), Audi (`www.audi.-com`), Ferrari (`www.ferrari.com`), and Lamborghini (`www.lamborghini.com`); all inspected May 18, 1999.

Most of these sites have a section called "Virtual Showroom" (or a similar name) that is organized as a guided tour. From a page promoting the style and the character of a car model, the user can access different views of a car (either pictures showing the interior or the exterior of the car or photos of the car on the road). The navigation structure of a virtual showroom is illustrated in Figure 5, and corresponds to a circular guided tour, as described above.

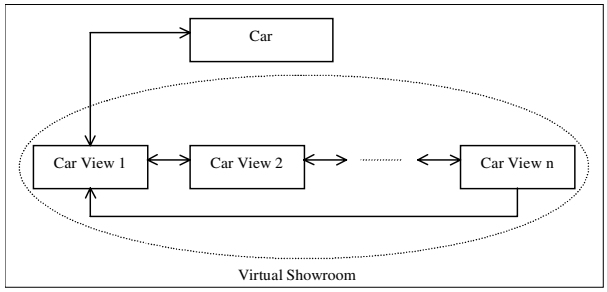

Fig. 5. The navigation structure of a Virtual Show Room

Pattern Name. Hybrid Collection
Problem. To provide easy-to-use access to a small group of objects, allowing both a complete scan and searching for a specific member.
Solution. Combine solutions of Index Navigation and Guided Tour Navigation.
Related Patterns. Index Navigation, Guided Tour Navigation, and Collection Center
Discussion. This pattern provides a compromise to satisfy the requirements of different types of users, or of different situations of use, within the same design. Users can choose different navigation styles according to their (evolving) needs. First-time users, for example, may prefer to traverse the collection sequentially in the guided-tour style. For subsequent visits they may adopt the index style, looking for specific members already seen.
Examples. This hybrid pattern is adopted, among other sites, at the National Gallery of Washington (`www.nga.gov`; May 18, 1999). This site presents a vast

selection of paintings exhibited in the museum and is one of the best organized museum applications on the Web. Access to artwork is available through searching or via tours. The typical navigation structure of a tour is depicted in Figure 6. Each collection center provides a short introduction to its tour and shows a list of all the paintings included in the tour. Navigation can therefore proceed by selecting any of the listed works. From each work the user can either proceed to navigate to the center (index page) or directly choose another work. "Next" for the last member of the collection is linked to the center (index) page.

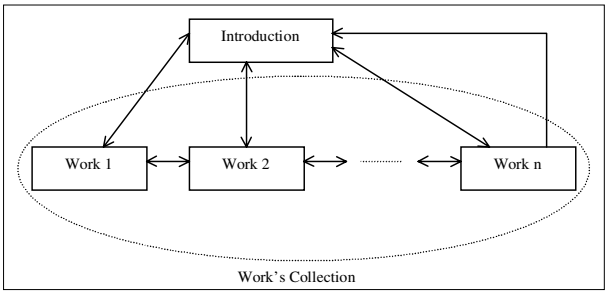

Fig. 6. The navigation structure of a tour in the National Gallery of Washington

Pattern Name. Collection Center
Problem. To make a collection well understood and usable.
Related Patterns. Index Navigation, Guided Tour Navigation, and Hybrid Navigation
Solution. A collection center can provide several types of additional information to improve the usability and effectiveness of an application. For simple collections it may be sufficient to provide the collection title and expressive anchors (place-holders) for its members. Link anchors have the dual role of describing the collection content and also providing a navigation mechanism. For complex collections it may be useful instead to design an additional page specifically devoted to explaining the purpose, rationale, and background of the collection. In general a collection center may include the following elements:

- An overview of the collection theme and purpose.
- An explicit description of its scenarios of use.
- Information common to all members (e.g. if the collection is about the works of a given artist, the center may include a general introduction to the artist's work).
- Anchors (placeholders) for links that allow access to collection members. Anchors should identify their destination by means of appropriate icons, textual labels, or combinations of both.
- Anchors may also convey expressive representations of destination objects. Such representations could provide a preview of collection members by means of thumbnail-sized pictures or short descriptions.

– The arrangement of anchors within the center should also convey the collection topology, i.e. the arrangement of its members.

Discussion. This pattern addresses both structure and presentation aspects. From a structural perspective, the key issue is how to select, for each member of the collection, information elements that best describe it. The choice may depend upon several factors: the nature of the object itself, the context provided by the collection, the user profile, etc.

From a presentation perspective it must be possible to arrange the information elements of the center in an aesthetically pleasing presentation and in a usable manner. Let us consider the case where collection members are paintings. The information items associated with anchors of the collection center could be chosen from several possibilities, such as painting title, thumbnail photograph, artist name, technique, date and place of creation, or name of the museum where it is exhibited. Each of these information items could be very relevant or obvious, according to the context, user profile, or other factors. If the collection is "paintings of X," the data about the painter are clearly irrelevant; if, however, the collection is "technique Y," references to the technique are irrelevant. Miniaturized pictures might be relevant for moderately experienced users but irrelevant for art experts.

Examples. The Louvre (`www.louvre.fr`) and the National Gallery of London (`www.nationalgallery.org.uk`), show examples of design solutions illustrated in the pattern. In both sites, the entry point to a collection of artwork is represented in a separate page, where each member is described by a thumbnail picture, the artwork title, and period (in the Louvre) or the painter name (in the National Gallery). Examples of collection centers in these sites were shown in Figures 1, 2, and 3.

While the collection center in the Louvre includes the collection title and the list of members, in the National Gallery site the center also provides a painting image (alluding to the theme of the collection) and a short textual introduction to the collection subject (see Figure 3). The introduction describes the selection criterion for the collected paintings, their main subjects, and the most important painters represented.

4 Conclusions

The design of complex Web applications can greatly benefit from the adoption of design patterns. These benefits include:

– Improved quality of the result.
– Reduction of the effort, time, and cost to develop.
– Improved reliability.

We wish to discuss again the interrelationship between conceptual design, as traditionally intended, and usage of design patterns.

Conceptual models retain their usefulness, but with a radical change of role: instead of providing conceptual primitives that a designer uses to "think of" an application, they provide the basic lexicon and syntax that can be used to define design patterns. For this reason the discussion about merits and demerits of different modeling primitives becomes less relevant. The concept expressed by a design pattern is independent from the model used to describe it.

The greatest impact is on the design process. The designer is induced to think in terms of application requirements (problems) and solutions (patterns), rather than in terms of pure modeling. Most of the benefits of such a change have already been illustrated in this paper. We would like to mention here an additional benefit: the brevity of the documentation required. Describing a complex guided tour from scratch may require up to 2–3 pages of documentation; describing it by using HDM may require half a page of documentation; describing it by referring to a well-known design pattern, requires only a few parameters, i.e. a few lines of documentation. The gains are precision, compactness, and completeness of the documentation.

A more radical point of view could be the following: with design patterns we are introducing new, high-level design primitives, therefore using design patterns is the same as defining a new higher level design model. In a sense this argument is true, since the design pattern at one level is clearly a model primitive at a higher level.[4] Nevertheless the distinction remains practically relevant: we expect from a design model a few consistent, well-organized primitives. We cannot expect the same from a large library of design patterns, where many people have contributed; we may expect there to be redundancy, inconsistency, lack of organization, etc. From a theoretical point of view (to be further investigated) we may say that we need a field of design patterns until that field becomes mature enough that a new, higher level model can incorporate the essence of the design primitives required.

The most relevant problem for a Web designer today, assuming s/he is convinced of the advantages of using design patterns, is to have a suitable supply of them at hand. Designers may generate patterns from their own experience, but probably there will not be many such patterns, especially if the designer is not very experienced. The second possibility is to get patterns from friends, but this may be time consuming and ineffective. Such patterns may be poorly organized, too specific, or too general to be useful; they may deal with the wrong area of interest (e.g. they emphasize presentation issues when the designer needs a navigational pattern); they may be defined at the wrong level of abstraction.

Having recognized these problems, ACM SIGWEB (the ACM Special Interest Group on Hypertext, Hypermedia, and the Web) has launched an initiative to create an "On-Line WWW Design Pattern Repository" [17]. The repository is coordinated by a small steering committee, and operationally managed by Politecnico di Milano (for design and development) and USI – Università della

[4] We could even say, for example, that the "for" operation in C++ is a pattern for combining assembly language instructions, or that a "many-to-many relationship" in the ER Model is nothing more than a pattern for the Relational Model.

Svizzera Italiana in Lugano–Switzerland (for designing and managing operations).

The basic idea is that the repository will be open to all contributions from all different communities, and it will cover all the possible areas of interest and all the interesting levels of abstraction. The repository will not enforce any predefined view of what a design pattern should, or should not, be. It will enforce a bare minimum of standardization of the description format and will require evidence of the usage of each submitted pattern: i.e. at least three designers, other than the person submitting the pattern, should have used it.[5] For practical reasons also tentative patterns will be accepted, i.e. patterns whose definition is supported by just one application. A system of keywords and classifications (by scope, application area, level of abstraction, etc.) will help designers to quickly locate useful patterns.

Before Summer 1999 the repository will be open for submissions;[6] before the end of 1999 the repository will be open for access to designers.

References

1. C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King and S. Angel. *A Pattern Language.* Oxford University Press, New York, 1977.
2. B. Appleton. *Patterns and Software: Essential Concepts and Terminology.* Available at `www.enteract.com/~bradapp/docs/patterns-intro.html`, 1997.
3. Bernstein M. "Patterns of Hypertext", In *Proc. of the ACM International Conference on Hypertext '98,* ACM Press, 1998, pp. 21–29.
4. M. A. Bochicchio, P. Paolini, "An HDM Interpreter for On-Line Tutorials," In *Proceedings of MultiMedia Modeling 1998* (MMM'98), pp.184–190, Ed. N. Magnenat-Thalmann and D. Thalman, IEEE Computer Society, Los Alamitos, California, USA, 1998.
5. Brodie M.L., "On the Development of Data Models". In Brodie M.L., Mylopoulos J., and Schmidt J., (eds.) *On Conceptual Modeling.* Springer Verlag, 1984.
6. M.P. Cline, "Using Design Patterns to Develop Reusable Object-Oriented Communication Software", *Communication of ACM,* 38 (10), October 1995, pp. 65–74.
7. Discenza A., Garzotto F., "Design Patterns for Museum Web Sites". In *Proceedings MW'99 — 3rd International Conference on Museums and the Web*, New Orleans , USA, March 1999, pp. 144–153.
8. M. Fowler, *Analysis Patterns. Reusable Object Models*, Addison-Wesley, 1997.
9. E. Gamma, R. Helm, R. Johnson and J. Vlissides. *Design Patterns. Elements of Reusable Object-Oriented Software.* Addison-Wesley, 1996.
10. F. Garzotto, P. Paolini and D. Schwabe. "HDM - A Model Based Approach to Hypermedia Application Design". *ACM Transaction On Information Systems*, 11 (1), 1993, pp. 1–26.

[5] We remind the reader that a design pattern should not be an "intuition" but should synthesize some consolidated experience.

[6] The interested reader, possibly willing to submit a pattern, is invited to contact `davide.bolchini@lu.unisi.ch` or `sara.valenti@hoc.elet.polimi.it` for further information.

11. Garzotto F., L. Mainetti, P. Paolini "Adding Multimedia Collections to the Dexter Model". In *Proc. ACM ECHT'94*, Edinburgh (UK), September 1994, pp. 70–80.
12. Garzotto F., Mainetti L., Paolini P. "Hypermedia Design, Analysis, and Evaluation Issues", *Communications of the ACM*, 38 (8), August 1995, pp. 74–87.
13. F. Garzotto, L. Mainetti and P. Paolini. "Information Reuse in Hypermedia Applications". In *Proc. of the ACM International Conference on Hypertext '96*, ACM Press, 1996, pp. 43–54.
14. Garzotto F., Matera M. "A Systematic Method for Hypermedia Usability Inspection". In *The New Review of Hypermedia and Multimedia*, 3 (1), January 1997, pp. 39–65.
15. F. Garzotto, M. Matera, P. Paolini "To Use or not to Use? Evaluating Usability of Museum Web Sites". In *Proc. MW'98 – 2nd International Conference on Museums and the Web*, Washington DC, May 1998 — available at `www.archimuse.com/mw98/`.
16. Garzotto F., Matera M., Paolini P. "A Framework for Hypermedia Design and Usability Evaluation." In *Proc. of IFIP-DEUMS'98 - International Working Conference on Designing Effective and Usable Multimedia Systems*, Stuttgart, Germany, September 1998, pp. 14–28.
17. F. Garzotto, P. Paolini. "Design Patterns for WWW Hypermedia: Problems and Proposals." In *Electronic Proceedings of the ACM HT99 Workshop on Hypermedia Development: Design Patterns in Hypermedia* — available at `ise.ee.uts.edu.au/-hypdev/ht99w/`.
18. T. Isakowitz , E. Stohr, P. Balasubramanian "RMM: A Methodology for Structured Hypermedia Design", *Communications of the ACM*, 38 (8), August 1995.
19. C. Larman, *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design*, Prentice Hall PTR, 1998.
20. G. Mecca, P.Atzeni, A.Masci, P.Merialdo, G.Sindoni: "The Araneus Web-Base Management System". In *Proc. SIGMOD'98*, 1998, pp. 544–546.
21. G. Meszaros, J.Doble, "A Pattern Language for Pattern Writing", available at `www.mit.edu/~jtidwell/onteraction_patterns.html`.
22. M. Nanard, J. Nanard and P. Kahn. "Pushing Reuse in Hypermedia Design: Golden Rules, Design Patterns and Constructive Templates". In *Proc. of the ACM International Conference on Hypertext '98*, ACM Press, 1998, pp. 11–20.
23. Nielsen J. *Usability Engineering*, Academic Press, New York, 1993.
24. P. Paolini, P. Fraternali "A Conceptual Model and a Tool Environment for Developing More Scalable, Dynamic, and Customizable Web Applications." In *Proc. of EDBT'98 Conference*, Valencia, Spain, 1998, pp. 421–435.
25. G. Rossi, D. Schwabe and A. Garrido. "Design Reuse in Hypermedia Applications Development". In *Proc. of the ACM International Conference on Hypertext '97*, ACM Press, 1997, pp. 57–66.
26. G. Rossi, D. Schwabe and F. Lyardet, "Improving Web Information Systems with Design Patterns". In *Proc. of the 8th International World Wide Web Conference*, Toronto (CA), May 1999, Elsevier Science, 1999, pp. 589–600.
27. D.C. Schmidt, R. E. Johnson and M. Fayad. "Software Patterns". *Communications of the ACM, Special Issue on Patterns and Pattern Languages*, Vol. 39, No. 10, October 1996, pp. 37–39.
28. D. Schwabe and G. Rossi. "An Object Oriented Approach to Web-Based Application Design". *Theory and Practice of Object Systems*, 4 (4), J. Wiley, 1998.