

A System of Patterns for Web Navigation

Mohammed Abul Khayes Akanda and Daniel M. German

Department of Computer Science, University of Victoria, Canada
maka@alumni.uvic.ca, dmgerman@uvic.ca

Abstract. In this paper we propose a system of design patterns for Web navigation. We have collected patterns already published in the literature, selected ten of them, refined them and identified the relationships among them. The selected patterns are rewritten in the Gang of Four (GoF) notation. They are implemented and integrated together leading to a framework intended to be used as the central part in developing data intensive Web applications.

1 Introduction

Compared to software engineering –where there are several published books with design patterns– hypermedia design patterns are in an early stage: they are scattered in the literature and they have barely been scrutinized. Paolini and Garzotto stated [13] “we have not seen many real ‘booklets’ of design patterns, ready to be used”. The number of hypermedia design patterns keeps growing and it is becoming more difficult to keep track of them, to know whether there is a pattern that solves a determined problem, and, if there exists, to find it. In [9], we argued that it was necessary to:

- Unify the patterns. Some patterns provide insight that can enhance similar patterns. Some patterns provide different views of the same problem and they should be unified into a single super pattern including all the insights provided by the patterns individually.
- Patterns should be rewritten using a common pattern language and vocabulary. The hypermedia designers in general have not decided on a common vocabulary and this is reflected in the pattern descriptions.
- Patterns don’t exist in isolation. There are many interdependencies between the patterns. A list of patterns doesn’t reflect those relationships, and doesn’t provide any guideline for their implementation. It also doesn’t show how the patterns may evolve. Patterns should be organized into pattern systems. Buschmann et al. [4] defined a pattern system in the scope of software architecture. Their definition can be easily adapted to hypermedia application: a pattern system is a collection of patterns, together with guidelines for their implementation, combination and practical use in hypermedia development.

In this paper, we describe a system of patterns for Web navigation. Different authors are classifying the published patterns from different perspectives. But so far, there was no formalized system of patterns for hypermedia design patterns. Section 2 describes how we have created a system of patterns from the navigational hypermedia design patterns leading to develop a framework, implementing the patterns included in this

system, intended to be used for the creation of data intensive Web applications. Section 3 describes the benefits of creating system of patterns and rewriting the patterns in Gamma notation.

2 A System of Patterns for Web Navigation

Our objective was to create a system of patterns for information navigation in hypermedia applications. According to Buschmann et al. [4] a system of patterns should satisfy the following requirements: it should (1) comprise a sufficient base of patterns, (2) describe all the patterns included in the system in a uniform notation, (3) expose the various relationships among the patterns, (4) organize its constituent patterns, (5) support the construction of hypermedia systems, (6) support its own evolution.

In particular, we wanted to concentrate in design patterns that would help address the following navigational problems: (1) making information easy to explore, (2) helping users to find the desired information quickly, (3) providing multiple navigation paths for different users, and (4) assisting users in knowing the current position in navigation.

We have found 96 hypermedia design patterns published in the literature and organized them in [3]. From these, we selected the following most frequently used navigational design patterns in web information systems or in data intensive web applications, based on behavioral relationships among the pattern which is described in Section 2.1: *Active Reference*[1, 8, 12, 14], *Guided Tour*[1], *Landmark*[15], *News*[1, 5, 11], *Set-Based Navigation*[1, 15], *Shopping Basket*[5], *Simple Search Interface*[1], *Selectable Search Space*[1]. In addition, we discovered two new patterns: *Visited Objects* [3], *Hierarchical Navigation* [3].

2.1 Relationships Among the Patterns

The patterns in this system were chosen based on behavioral relationships (i.e., how they interact with one another when they are used in an application together) such that when they were implemented and integrated together, would lead to a framework to be the central part of data intensive web application. Figure 1 depicts the relationships among the patterns. The diagram shows the flow of information from one pattern to another. For example, *Set based navigation* gets the search results to be explored from *Selectable search space* and *Simple search interface*. *Set-Based Navigation* plays a central part in this system, as many other patterns rely on it to display specific information to the user. This figure shows that *Simple search interface* is an extended version of *Selectable search space*, *Hierarchical navigation* is an extended version of the *Set based navigation*. We can get all the behaviors of *Selectable Search Space* using *Simple search interface*. The same is true for *Hierarchical navigation* and *Set based navigation*. In fact, still they all are strong enough to stand by themselves. *Shopping basket* is a special case of *Visited objects*. The set of all the elements of *Shopping basket* will be the subset of the set of all the elements of *Visited objects* when they both are instantiated in an application. In fact, it is not possible to get all the behaviors of *Shopping basket* using *Visited objects* only.

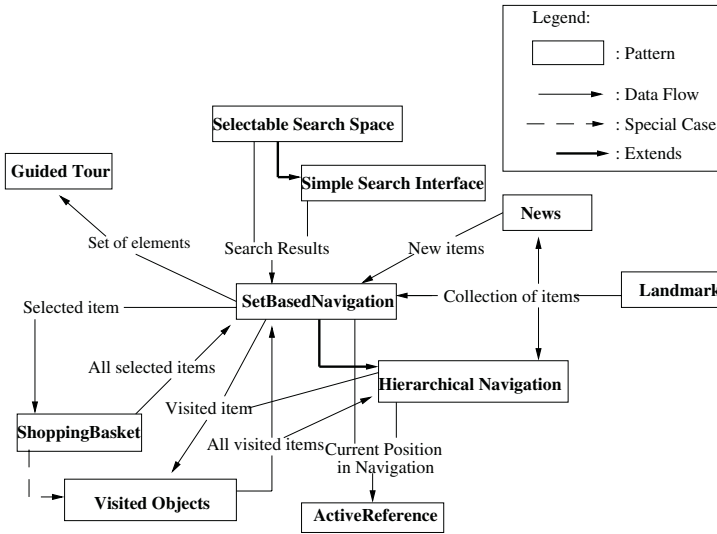


Fig. 1. Relationships between the patterns

2.2 Patterns in GoF Notation

The next step was to describe the patterns in a uniform notation and we decided to use the Gang-of-Four (GoF) one [6, 7]. One important aspect that we want to include in our descriptions is the diagrams that provide insight on how the pattern should be implemented. In the GoF notation, patterns are described using the UML notation. We felt that UML was not sufficient to describe navigational patterns, due to its inability to depict navigation. We decided to use UML-based Web engineering notation (UWE) [10]. Two diagrams of UWE have been particularly useful: *Navigation class* and *Presentation class*. The modeling element, *Navigation class* is used here to represent an information space to be explored and *Presentation class* is used to represent the actual pages or part of the pages created by different patterns to be shown in browser. Here we are including the description of the pattern *Hierarchical Navigation*. The rest of the patterns can be found in [3].

Hierarchical Navigation

- **Name:** Hierarchical Navigation
- **Intent:** Organize the information by creating sets of related items recursively, create a hierarchical structure like tree of those sets and allow inter- and intra-set navigation.
- **Motivation:** Collections of objects are commonly displayed in a Web application. They need to be displayed in an easy-to-navigate way. The objects of collections can be explored like exploring the files in windows explorer or Konqueror.
- **Applicability:** This pattern will be applicable in exploring different collections of items in different Web Information Systems.
- **Structure:** Figure 2 shows the structure of the pattern.

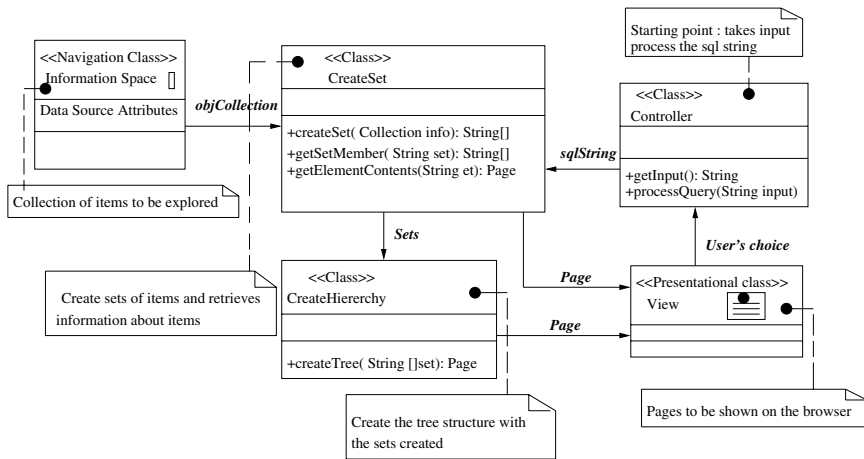


Fig. 2. The structure of Hierarchical Navigation

– **Participants:** The participants of this pattern are:

1. *Information space*: It represents the collection of items to be explored like a collection of books, movies, CDs etc.
2. *CreateSet class*: It creates the sets of related items. It may provide the detail information of an item of a particular set too.
3. *CreateHierarchy class*: It creates a hierarchical structure using the sets created like the tree structure created using the files and/or directories in file explorer like Windows explorer or Konqueror.
4. *Controller*: It processes the SQL string based on which sets will be created.
5. *View*: It is the user interface to show the pages.

– **Collaborations:**

1. *Information space*: This will provide the collection of items to be explored to *CreateSet* class.
2. *CreateSet class*: It receives the collection of items from the *Information space* and provides the sets created to *CreateHierarchy* class. It also sends pages containing the element(s) of a set to *View*.
3. *CreateHierarchy class*: It receives the sets from *CreateSet* class and provides pages containing the hierarchy of sets to *View*.
4. *Controller*: It takes user input using the *View* and provides the SQL string to *CreateSet* class.
5. *View*: It receives the pages from *CreateSet* class and *CreateHierarchy* class.

– **Known Uses:** Figure 3 shows the instance of the pattern in www.atPictures.com.

– **Related pattern:** This pattern is an extended version of Set based navigation.

2.3 Implementation

We have implemented all the patterns included in this system in the form of Java components (using the Tomcat Apache Web server) and we have developed a framework

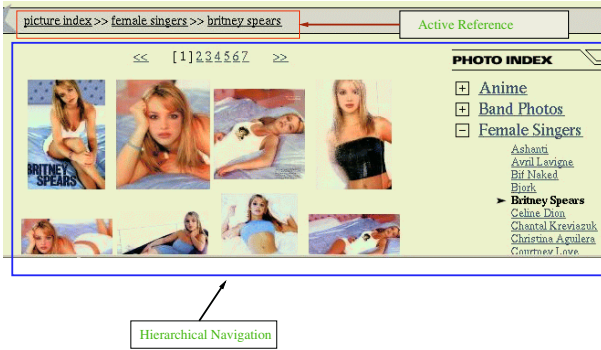


Fig. 3. Instance of Hierarchical navigation in www.atpictures.com

integrating all the developed components intended to be used to navigate an information space in Web.

A framework is a set of cooperating classes, some of which may be abstract, that make up of a reusable design for a specific class of software and usually incorporates several design patterns [16]. We have developed a navigational framework incorporating all the selected navigational design patterns. The individual components of the developed framework are independent. It is possible to use the components individually instead of the whole framework. It is also possible to replace one component by a different implementation as long as it maintains the same interface. This framework is extensible by incorporating more patterns as long as they maintain a behavioral relationship with the already incorporated patterns. The core concept of this framework of our implementation is that each pattern will be a component and we have described the implementation details in [2, 3].

3 Conclusions

A system of patterns shows how the patterns are connected together which will help in choosing a suitable pattern for a particular problem at hand. It provides guidelines for implementation, and facilitates in comparing the patterns, as all the patterns included in the system are written in a uniform notation. It shows practical uses of the patterns, and supports its own evolution. So, a system of patterns is arguably stronger than just a catalog of patterns.

One important contribution of this work is the structure diagram of each of the patterns, which represents the patterns graphically. None of the previous versions of these patterns contains a structure diagram of the patterns. We believe that the patterns described herein using this notation are more explanatory, easier to understand, and to use in practical applications.

The depicted relationships among the patterns shows how the patterns of this system are connected together or how they will interact with one another when they are being instantiated in application. This will help the users in choosing the required pattern(s) from the patterns included in this system for an e-commerce application or an Web Information Systems.

References

1. ACM-SIGWEB and University of Italian Switzerland. Hypermedia Design Pattern Repository. Available at <http://www.designpattern.lu.unisi.ch/HypermediaHomePage.htm>, January 2003.
2. M. A. K. Akanda and D. M. German. A Component Oriented Framework for the Implementation of Navigational Design Patterns. In *International Conference on Web Engineering (ICWE'03)*, pages 449–450, 2003.
3. Mohammed Abul Khayes Akanda. A system of patterns for web navigation. Master's thesis, University of Victoria, Canada, December 2003.
4. F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. *Pattern-Oriented Software Architecture- A System of Patterns*. John Wiley Sons Ltd., New York, 1996.
5. G. Rossi and D. Schwabe and F. Lyardet. Improving Web information Systems with Navigational Patterns. In *Proceedings of the 8th International World Wide Web Conference*, Available at www-di.inf.puc-rio.br/schwabe/papers/www8.pdf, May 1999. W3C, Elsevier.
6. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley, Reading, 1996.
7. Erich Gamma, Richard Helm, Ralph E. Johnson, and John M. Vlissides. Design Patterns: Abstraction and Reuse of Object-Oriented Design. In *ECOOP1993*, pages 406–431, 1993.
8. A. Garrido, G. Rossi, and D. Schwabe. Pattern Systems for Hypermedia. In *Proceedings of The 4th Pattern Languages of Programming Conference*, Available at <http://st-www.cs.uiuc.edu/users/hanmer/PLoP-97/Workshops.html>, 1997. University of Washington.
9. D. M. German and D. D. Cowan. Towards a unified catalog of hypermedia design patterns. In *Proceedings of the 33th Hawaii International Conference on System Sciences*, Available at http://www.turingmachine.org/dmg/research/papers/dmg_hicss2000.pdf, Jan. 2000.
10. R. Hennicker and N. Koch. Systematic Design of Web Applications with UML. In *Unified Modeling Language: Systems Analysis, Design and Development Issues, 2001*, K. Siau and T. Halpin, Idea Group Publishing, Available at <http://www.pst.informatik.uni-muenchen.de/personen/kochn/publications.html>, 2001.
11. F. Lyardet, G. Rossi, and D. Schwabe. Patterns for Dynamic Websites. In *Proceedings of The 4th Pattern Languages of Programming Conference*, Available at http://jerry.cs.uiuc.edu/plop/plop98/final_submissions/P56.pdf, 1998.
12. F. Lyardet, G. Rossi, and D. Schwabe. Using Design Patterns in Educational Multimedia Applications. In *Proceedings of EDMedia'98*, Available at http://www.oohdm.telemidia.puc-rio.br/site_oohdm/oohdm.html, 1998.
13. P. Paolini and F. Garzotto. Design Patterns for the WWW hypermedia: problems and proposals. In *Hypermedia Development: Design Patterns in Hypermedia*, 1999.
14. G. Rossi, D. Schwabe, and A. Garrido. Design Reuse in Hypermedia Applications Development. In *Proceedings of the Eighth ACM Conference on Hypertext*, Hypertext Design, pages 57–66, 1997.
15. G. Rossi, D. Schwabe, and F. Lyardet. Improving Web Information Systems with Navigational Patterns. In *Proceedings of the Eighth International World-Wide Web Conference*, Available at www-di.inf.puc-rio.br/schwabe/papers/www8.pdf, 1999.
16. C. Szyperski. *Component Software*. Pearson Education Ltd., Great Britain, 1999.