Research Journal of Applied Sciences, Engineering and Technology 3(10): 1171-1176, 2011
ISSN: 2040-7467

Submitted: July 20, 2011 Accepted: September 25, 2011 Published: October 20, 2011

Research on Web Navigations

Qian Zhongsheng
School of Information Technology, Jiangxi University of Finance and Economics,
Nanchang, 330013, China

Abstract: Web applications employ various new languages, technologies, and programming models to implement applications with very high quality requirements. In building Web applications, three types of models must be considered, which are conceptual model, navigation model and presentation model. Navigation problems are significant among them that must be confronted. This work illuminates three different classifications of Web navigations as well as their feature, analyzes and compares various popular Web navigation models. Moreover, it also presents and suggests some guidelines for Web navigation design based on Web development design pattern experiences.

Key words: Navigation model, Web application, web navigation

INTRODUCTION

Among the Web activities, the most popular one should be navigating. Web applications make users more convenient and efficient, but the problem of leading them to "getting lost in hyperspace" arises for a mass of information in a hypertext database. A person ability to find and navigate effectively to new information and to new Web applications is extremely important, and this has driven many researchers to understand both how people navigate within the Web, and how Web applications and browsers should be designed.

Navigation plays a significant role in helping users to find their way in the complex cyberworld, viz. to know where they are, how to return to a previously visited page, what information is available and how to access information of interest without losing their way when wandering in some isolated subspace. Before considering these problems, the basic concept of navigation should be analyzed carefully and understood clearly.

This study reviews the current Web navigations and their modeling approaches. It is not intended to cover every range of Web navigations, but rather to complement them. The primary purpose of this study is to identify the research progress of Web navigations and to provide some suggestions for further research.

Classifications and feature of web navigations: Navigation is a central activity widely used in Web applications. Its original meaning is bound to the hypertext idea, expressing the action of jumping from page to page through a hyperlink, viz. a path from a source page to a target one, which involves interaction with browser buttons, is simply regarded as navigation. The advance of Web technology has afforded us a wider vision of the navigation concept, which is not only the action of jumping from page to page, but the idea of moving through an information space (Reina and Torres, 2002). Schwabe and Rossi (1998), navigation is defined as the sensation the user has when s/he navigates through an object space from the application domain. These objects are not the conceptual ones, but are customized according to the user profile using the view mechanism.

Intra and inter-page, dynamic-content, frame-based, multi-windowed and browser-specific navigations: In a Web application, Web pages and links replaces information fragments and relationships in the hypertext model. Users usually navigate from a source page to a target one (yet, they are allowed to be the same page and hence, linking to different sections of the same page is possible) by clicking on a hyperlink. Moreover, various client-side scripts, such as Javascript and VBScript, provide other ways to control traversal between different Web pages without clicking on a hyperlink or button. Dynamic HTML, ActiveX components are also widely employed to change the content and hyperlinks of the currently displaying Web pages. Concurrent display of Web pages is feasible by using frames and windows. Synchronization between the concurrently displayed Web pages can be achieved with functions of the client-side scripts and controlled programs. In addition, Web browsers can provide additional navigation functions that are out of control of the Web pages, such as scrolling, back, forward, refresh buttons, history lists and bookmarks (or favorites).

Therefore, according to different quarries that result in navigations mentioned above, Web navigations are divided into five categories: Intra and inter-page navigation (often by clicking on the hyperlink), dynamic-content navigation (usually seen in search engines), frame-based navigation, multi-windowed navigation (for example, popping windows) and browser-specific navigation (i.e., back, forward or refresh buttons of a

Web browser are pressed). A detailed analysis on different kinds of Web navigations is presented by Karl *et al.* (2000).

Frames inside a browser window make concurrent view of Web pages possible, with each containing a separate Web page for viewing. And all the frames in a window are created and destroyed at the same time. Frames can ease the problem of redundant traversals in hub-and-spoke browsing (People visit a central page named hub, then follow the links to a new page named spoke and back again) by allowing multiple independent frames on a single screen, each with its own distinct URL. Unfortunately, navigations within sub-frames are not recorded on the history list; hence, pages cannot be revisited using the back button of a Web browser.

History lists provided by Web browsers allow people to select and revisit pages viewed previously. (Tauscher and Greenberg, 1997) identified seven browsing patterns, which are first-time visits, revisits to pages, authoring of pages, regular use of Web-based applications, hub-and-spoke, guided tour (some pages include structured links such as next page, which can be chosen to follow), depth-first search (users follow links deeply before returning to a central page). The history lists should support the many browsing patterns users exhibit. For example, the refresh button is very convenient for authoring. Stack-based history mechanisms and the *back* button support both hub-and-spoke and depth-first search patterns. Guided tours contain hyperlinks that encourage a linear pattern of navigation. Perhaps the excessive backtracking that results from depth-first navigation styles and the hub-and-spoke could be reduced by a graphical navigation map. A navigation map describes the possible paths from page to page. However, the components on the server side can and typically do modify and redirect requests, making the resulting navigation very difficult to discern. Navigation routing for Web applications describes how requests from Web pages are routed through components on the server (Han and Hofmeister, 2004).

The Back and Forward buttons do not control browsing of a temporal ordering of previously visited pages, but rather determine the currently displayed page in a *stack* of pages. At the top of the stack is the page that has been most recently loaded. At the bottom of stack is the page that was least recently loaded. Forward and Back allow the user to move up and down the stack of the history list, to recall previously visited pages.

Adaptive and non-adaptive navigations: In the simplest case of the navigation, the next page is determined by the current page and the action (e.g., clicking on a link or pressing a button) selected by the user. However, many Web applications now incorporate adaptation, where the next page also depends on the user mode, for example whether they are a customer or an administrator, or depends on what pages the user has visited previously. Adaptive Web applications can semi-automatically improve their organization and presentation by learning from visitor access patterns (Perkowitz and Etzioni, 2000). The Web application supports adaptation in all its variants and allows the system to take control over the process of navigation under special conditions, such as a period of user inactivity. Some current Web applications make adequate use of past knowledge about the individual user who is using the system or of past experience gained by the group of users s/he belongs to. Such knowledge can be used to adapt the system to the user goal.

Therefore, Web navigations are also classified into adaptive navigation and non-adaptive navigation. One type of adaptive navigation is user-mode adaptive navigation, where the navigation depends on the user mode. Another type of adaptive navigation is history-sensitive navigation, where the navigation depends on history actions or previously visited pages. In adaptive navigation, as a user navigates through the pages, the Web application may keep track of what pages s/he has accessed and put her/him into one or more known categories if possible. Links to interesting pages may then be inserted for her/him to follow. It is important that Web applications learn from a few examples and adapt quickly to changing user interests, which must be inferred implicitly from actions. Moreover, the adaptive interface should not limit the user choices by filtering information.

.

Task-directed, task-related and task-unexpected navigations: A Web application can be regarded as a structured set of objects that may be navigated and processed, in order to achieve one or more tasks. An alternative to classify Web navigations is task-oriented or task-based in terms of whether users traverse through Web applications with their tasks in mind or not and which type of the tasks. Thus, we can group Web navigations into three types: task-directed navigation, task-related navigation and task-unexpected navigation. A user goal is undoubtedly definite and extremely concrete in task-directed navigation beforehand, as often leads directed search. In task-related navigation, the seeking sources have a high likelihood of items of interest. The navigation is associated with some goal, but the goal is not directed and enough specific. In task-unexpected navigation, a user searches for the unexpected element or a haphazard connection with no goal, as is purely random. This continuum provides a nice middle ground to distinguish between navigation as an approach to accomplishing a task and open ended navigation with no particular goal in mind. Marchionini *et al.* (1989) further developed this distinction in designating closed and open tasks. Closed tasks have a specific answer and often integrate subgoals. Open tasks are much more subject oriented and less specific. Similarly, Michael and Wan

(1994) discussed the use of backtracking within a multi-windowed hypertext environment. They introduced the concept of task-based backtracking, in which a user backtracks to compare information from different sources for the same task or to operate two tasks simultaneously. A similar technique, in a Web application, would be backtracking to review previously retrieved pages.

Web navigation is a complex activity and there are many other classifications. Some Web applications, for example, redirect the user to a certain page, whereas the user does not press the corresponding button or type the proper URL explicitly. Thus, Web navigations may be categorized into active navigation (e.g., the user presses a button) and passive navigation (e.g., the user is redirected to some page by the system). We do not make further research on this. In addition, different Web navigations can also be interwoven. Adaptive navigation, for example, is usually dynamic-content navigation and non-adaptive navigation is often intra and inter-page navigation.

VARIOUS WEB NAVIGATION MODELS

Many modeling techniques for Web applications include a navigation model. HDM (Hypermedia Design Method) (Garzotto *et al*., 1993) is appropriate for describing the structure of the application domain, and a few basic primitives (elements) are used to define navigation: nodes that are views of the conceptual classes, links that are views of the relationships from the conceptual schema and access structures such as indexes, guided tours and indexed guided tours. The primitives are corresponding to the UML elements Class and Association. Apart from these primitives, OOHDM (Object-Oriented Hypermedia Design Method) (Schwabe and Rossi, 1998) defines the navigational context, a new primitive to structure the navigational space. The navigational context is a set of nodes, links, context classes and other navigational contexts that are used to organize the navigation space in consistent sets that can be traversed following a particular order.

There is none standard notation for a navigation model. Some of the discussion occurs in the context of Web application modeling. These approaches focus on navigation models that contain conceptual objects and show how the conceptual objects connect. Examples are the “navigation space model” and navigation structure model employed by (Koch and Kraus, 2002), the navigation schema in Autoweb system (Fraternali and Paolini, 2000), and the navigation diagram in RMM (Relationship Management Methodology) (Isakowitz *et al.*, 1995). The most basic navigation model is a descriptive model that is used to represent the navigation, such as ASNM (Architecturally Significant Navigation Map). A navigation map contains request pages, response pages, arrows connecting them, and a label on the arrow to explain why the user would arrive at the response page. Another modeling technique in this category is the navigation model of UWE (UML-based Web Engineering) (Koch *et al.*, 2000), which improves OOHDM and uses an extended UML notation. The UWE methodology allows for the specification of conceptual, navigation, and presentation models for Web applications following the clear separation of the three structural concerns: content, hypertext, and layout. WebML (Web Modeling Language) (Ceri *et al.*, 2000) is a conceptual model for the design of Web applications. The modeling language offers a set of visual primitives for defining structural schemas that represent the organization and navigation of hypertext interfaces on top of the application data. For specifying the organization of data, the well known Entity-Relationship model is adopted. All visual primitives are accompanied by an XML-based, textual representation, which allows specifying additional detailed properties, not conveniently expressible in the visual notation, and provides the starting point for the automatic generation of the application code. Navigation models such as HMBS (Hypermedia Model Based on Statechart) (Oliverira *et al.*, 2001) and the navigation model by (Leung *et al.*, 2000) use Statecharts notation. The states in those models are Web pages and frames while the transitions are the navigation links. Both of the models are used mainly for representing navigation and providing user different levels of views by using hierarchical states. Other statechart-based models (featuring different ad-hoc extensions) are also frequently used for navigation design for hypertext (Paulo *et al.*, 1998). Differently, (Stotts and Furuta, 1989) proposed the navigation using Petri-nets.

When a model is formal, it can also be used to generate design or implementation components, and can be verified for properties such as broken links or length of navigation path. In (Yang and Shi, 2002), the authors presented WNM (Web Navigation Model) and built a navigation tree to analyze the navigational characteristics including, e.g., whether or not a page is an isolated one (such as the node unreachable or not in the navigation tree), whether or not a page is not linked by any other page (such as a dead-locked node) and so on. Casteleyn *et al*. (2003) presented a navigation model to support adaptive behavior such as navigation promotion and demotion during runtime (Casteleyn *et al.*, 2003). Han and Hofmeister (2006) proposed an approach that uses Statecharts to formally model adaptive navigation and showed how important properties of a navigation model are verified using existing model-checking tools. They also employed Z to define the navigation routing model (Baumeister *et al.*, 2005), and provided tools to extract and analyze the model. Baumeister *et al*. (2005) used aspect-oriented modeling techniques to model adaptive Web applications (Kaasten *et al.*, 2002). Links are

Table 1: The comparison of multiple web navigation modeling approaches

Item model	Tool	Primitive/Element	What to describe	Dynamic page	Mapping
HDM	ER Dexter	Entity, entity type, component, unit, link, index, guided tour, etc	Structure	no	no
OOHDM	OMT,UML Extension	Node, link, navigation context, guided tour, etc.	Structure, partial behavior	no	no
RMM	ER, RMDM	Entity, entity type, link, grouping, guided tour, etc.	Structure	no	no
UWE	UML	Class, association, query, navigation context, index, menu, guided tour, etc.	Structure, partial behavior	no	partially
WebML	ER,XML	Content unit, link	Structure	no	partially
HMBS	Statecharts	State, transition, event	Partial structure, behavior	yes	no

associated with aspects that read the user status and change the navigation accordingly.

As we know from above, the modeling approaches, such as HDM, OOHDM, RMM, UWE, WebML and HMBS present graphical representation for navigation modeling of Web applications formally or semi-formally. The comparison of various primary navigation modeling approaches in five aspects are shown in Table 1. In the table, column five entitled "dynamic page" indicates whether or not the navigation model supports for modeling dynamic pages and column six entitled "mapping" means if they support mapping among their own sub-models (yes for supporting, "no" for not supporting and "partially" for supporting partially). For lack of space, we do not provide more detailed discussion

Guidelines for web navigation design: Broken links, isolated pages, long paths and confused navigation are frequently reported on usability test and are symptomatic of the difficulty to design efficient navigation for large Web applications. When designing navigation there are two main kinds of links: Structural links and page-specific links. Structural links give general navigation over the Web application and usually are the same in different context, such as menus and navigation bars. Page-specific links are closely related to the content. Page-specific navigation aids (e.g., index) are found to reduce search time provided they were placed in a fixed position near the top of the page. Navigation controls placed within the Web application itself have been investigated. Local navigation controls such as previous, next and last, are generally found to relief both novice and experienced users. An interpretation of the rapid page navigation is that Web applications should be designed to shorten the navigation paths to popular designation pages. Good page design, with appropriate use of navigational shortcuts, should minimize the number of transient pages used.

In Web applications, the designer should not be forced to redefine each time the basic "link following" operation. It should be a built-in mechanism, complementary to method invocation. Moreover, designers need to be cognizant of the classification of expected users as serendipitous or general visitor. Customization is a complex task for several aspects must be considered: some customization choices depend upon the user profile and role; others depend upon the overall state and evolution of the Web application.

Navigation of Web applications is also affected by the design of the browser interface. Bookmarks (or favorites, hot lists) are heavily employed and have been shown to be a useful tool for hypertext navigation. Bookmarks are used to collect information from the Web, and speed up access to information. An indirect benefit of bookmarks is the high speed since caching of Web pages referenced in the bookmark can be done to speed up the loading process. History lists help users to return to more temporally distant pages. The history can be shown in a separate window using a tree-based structure with thumbnail iconic representation for each visited page. Thumbnails of Web pages prove to be a reasonably accurate way for people to identify both the Web application and the exact page (Kaasten *et al.*, 2002). If users only have a 60% chance of recognizing the exact content from a title or URL in the history list, they may not be motivated to invest the extra work it takes to operate the list (opening, scrolling, and closing) in order to track down a page. These results suggest that thumbnails are a slightly better representation than titles or URLs.

Browser designers should consider the mapping from their model of system behavior to users' models of the system. Improve each browser's system image to better reveal the underlying stack-based navigation model (Cockburn and Jones, 1996). An iconic representation of the history list stack, for instance, could dynamically reveal the stacking behavior: the growth of the stack as pages are loaded, the descent into the stack when pages are recalled, and the shrinking of the stack when loading causes pages to pop off the stack. Such an iconic representation would provide an additional hook for user synthesis of a correct mental model, without a demand for large amounts of screen real-estate.

Most approaches, however, usually focus on the designer's point of view about the content and the navigation. The design process typically starts from an informal description of the content or a flat hierarchy, and then the design is "implemented" in a try-and-error cycle heavily supported by tools like editors and driven by

guidelines (Lynch and Hortson, 1999). User's perspective is thus only introduced informally and usually in an implicit way through testing and interviews. When designing navigation, we have to pay attention to users mental model of the application as well as to provide efficient navigation for the most important user's tasks. As well, the designers can reorganize the hyperlink structure of a Web application based on the analysis of their user's intentions. Currently, there are many systems and agents such as WebWatcher (Joachims *et al.*, 1997), WebMate (Chen and Sycara, 1998), which are trying to predict the user's navigation intention from the user's previous navigation paths.

Naturally, some directions for navigation design are suggested as follows:

- The focus on the designer's viewpoint must be changed into the user-centric one, as is the most significant concern.
- The navigation in an adaptive Web application can be dynamically changed according to the user's status, the user's visiting path, and other contextual information. Adaptive navigation including *user-mode adaptive* navigation and *history-sensitive navigation* is extremely popular.
- Since users often visit a certain number of pages per server, as implies that "must be browsed" pages should be accessible within two to three jumps of the initial "homepage".
- When concurrent view of Web pages is indispensable, frames can be regarded, for each frame contains a separate Web page for viewing and all the frames in a window are created and destroyed at the same time.
- An additional issue is customization. Web applications may address a potentially huge variety of different users with different navigational requirements. When more than one class of users is expected, a separate document can be constructed for each class, thus providing tailored and customized user profiles.
- Graphical visualizations often give people an intuitive view. Representation in graphics such as thumbnail and overview maps helps the user to maintain a sense of context within an information space and to recognize the pages quickly. Moreover, graphical representations can provide a rich interface for the user to take navigational actions.
- In designing Web browsers, (Greenberg *et al.*, 2002) advocated the replacement of current stack-based *back* button with recency-based one only if other design considerations warrant them. They felt that good design opportunities do exist, especially for a recency-based *back* to be integrated with a recency-based history list to produce a single model of how pages can be revisited.

CONCLUSION AND FUTURE WORK

Web Navigations is an important research issue in the Web application software architecture research area. We must carefully design the navigational architecture of Web applications and understand the user tasks while s/he is navigating the hyperspace to decide which navigation facilities should be included.

For better building the navigational architecture, we must clearly understand the basic concept of Web navigation and analyze existent navigation models beforehand. In this work, we expound on three different classifications of Web navigations as well as their feature, analyze navigation modeling methods from various view points and then compare them. Moreover, we also present and suggest some guidelines for Web navigation design based on Web development design pattern experiences. Many efforts are made towards researching Web navigations and some effective but not salient results are also achieved. However, there are a variety of inadequacies, such as the deficiency of modeling structure and behavior together as well as the inability to support dynamic pages and excellent adaptivity in existent navigation models. In the future, we will conduct an insight into these features to build a flexible navigation model with respect to the Web navigation design guidelines proposed.

ACKNOWLEDGMENT

This work was financially supported by the Jiangxi Provincial Natural Science Foundation of China under Grant No. 2010GQS0048 and the School Foundation of Jiangxi University of Finance & Economics of China under Grant No. 05072015.

REFERENCES

Baumeister, H., A. Knapp, N. Koch and G. Zhang, 2005. Modeling adaptivity with aspects. Proceeding of the 5th International Conference on Web Engineering, Springer-Verlag, pp: 406-416.

Casteleyn, S., O. De Troyer and S. Brockmans, 2003. Design time support for adaptive behavior in web sites. Proceeding of the 18th ACM Symposium on Applied Computing, Melbourne, USA, pp: 231-236.

Ceri, S., P. Fraternali and A. Bongio, 2000. Web modeling language (Web ML): A modeling language for designing web sites. Proceeding of the 9th Int World Wide Web Conference, Amsterdam, Netherlands, pp: 890-894.

Chen, L. and L. Sycara, 1998. WebMate: A personal agent for browsing and searching. Proceedings of the 2nd International Conference on Autonomous Agents, pp: 132-139.

Cockburn, A. and S. Jones, 1996. Which Way Now? Analysing and easing inadequacies in www navigation. Intl. J. Hum. Comp. Stud., 45(1): 105-129.

Fraternali, P. and P. Paolini, 2000. Model-driven development of web applications: The auto web system. ACM Trans. Inf. Syst., 28(4): 323-382.

Garzotto, F., P. Paolini and D. Schwabe, 1993. HDM: A model based approach to hypermedia application design. ACM Trans. Info. Syst., 11(1): 1-26.

Greenberg, S., G. Ho and S. Kaasten, 2002. Contrasting Stack-Based and Recency-Based Back Buttons on Web Browsers, Report 2000-666-18, Department of Computer Science, University of Calgary, Alberta, Canada.

Han., M. and C. Hofmeister, 2004. Modeling Navigation Routing in J2EE Web Applications. Lehigh University Technical Report.

Han, M. and C. Hofmeister, 2006. Modeling and verification of adaptive navigation in web applications. Proceeding of the 6th International Conference on Web Engineering, pp: 329-336.

Isakowitz, T., E.A. Stohr and P. Balasubramanian, 1995. RMM: A methodology for structured hypermedia design. Comm. ACM, 38(8): 11-15.

Joachims, T., D. Freitag and T. Mitchell, 1997. Web watch: A tour guild for the world wide web. Proceeding of the 15th International Joint Conference on Artificial Intelligence, pp: 770-775.

Karl, R.P.H.L., C.K.H. Lucas, S.M. Yiu and W.M.T. Ricky, 2000. Modeling Web Navigation by Statechart. HKU, CSIS, Technical Report TR-2000-01, Retrieved from: http://cs.hku.hk//publications//techrep.

Kaasten, S., S. Greenberg and C. Edwards, 2002. How people recognize previously seen web pages from titles, URLs and thumbnails. Proceeding of Human Computer Interaction, pp: 247-265.

Koch, N., H. Baumeister, R. Hennicker and L. Mandel, 2000. Extending UML to model navigation and presentation in web applications. Proceeding of Modeling Web Applications in UML Workshop, York, England, pp: 1114-1117.

Koch, N. and A. Kraus, 2002. The Expressive power of UML-based web engineering. Proceeding of the 2nd Int. Workshop on Web-Oriented Software Technology, Malaga, Spain, June, pp: 105-120.

Leung, K.R.P.H., L.C.K. Hui, S.M. Yiu and W.M.T. Ricky, 2000. Modeling web navigation by state chart. Proceeding of the 24th Annual International Computer Software and Applications Conference (COMPSAC), pp: 41-47.

Lynch, P.J. and S. Hortson, 1999. Web Style Guide: Basic Design Principles for Creating Web Sites, Yale University Press, pp: 164-176.

Marchionini, G., 1989. Information-seeking strategies of novices using a full-text electronic encyclopedia. J. Am. Soc. Inform. Sci., 40(1): 54-66.

Michael, B. and J.L. Wan, 1994. Backtracking in a Multiple-window Hypertext Environment. ACM European Conference on Hypermedia Technology.

Oliverira, M.C.F. M.A.S. Turine and P.C. Masiero, 2001. A Statechart-based Model for Hypermedia Applications. ACM Trans. Inf. Syst., 19(1): 28-52.

Paulo, F.B., M.A. Turine, M.C.F. Oliveira and C.M. Paulo, 1998. XHMBS: A formal model to support hypermedia specification, Proceeding of the 9th ACM Conference on Hypertext and Hypermedia. Pittsburgh, USA, pp: 161-170.

Perkowitz, M. and O. Etzioni, 2000. Towards adaptive web sites: Conceptual framework and case study. Art. Int., 118(1-2): 245-275.

Reina, A.M. and J. Torres, 2002. Analyzing the Navigational Aspect. Second Workshop on Aspect-Oriented Software Development. Bonn, Alemania, pp: 111-118.

Schwabe, D. and G. Rossi, 1998. An object oriented approach to web-based applications design. TAPOS-Theory Practice Object System.

Stotts, D.P. and R. Furuta, 1989. Petri-net-based hypertext: Document structure with browsing semantics. ACM Trans. Inf. Syst., 7(1): 3-29.

Tauscher, L. and S. Greenberg, 1997. Revisitation Patterns in World Wide Web Navigation. ACM SIGCHI'97 Proc. of the Conference on Human Factors in Computing Systems, Atlanta, Georgia, USA, 22-27, ACM Press, pp: 325-329.

Yang, W.D. and B.L. Shi, 2002. Web applications navigation modeling based on statecharts and analysis of navigation properties. J. Comp. Res. Dev., 39(8): 992-997.