A Comparison Study of Web Development Methods

André Luís dos Santos Domingues[1,2], Sandro Lopes Bianchini[1], Reginaldo Ré[1], and Fabiano Cutigi Ferrari[1]

[1] Universidade de São Paulo,
Av. Trabalhador São-carlense. 400, 13560-970 São Carlos, SP, Brazil
{alsd,sandrolb,rr,ferrari}@icmc.usp.br
[2] Centro Universitário Barão de Mauá,
R. Ramos de Azevedo. 423, 14090-180 Ribeirão Preto, SP, Brazil

Abstract. The increasing expansion and complexity of Web applications poses new challenges for software developers. As a result, several development methods for this class of application have been proposed in the last years. However, the majority of these methods only support the project stage satisfactorily. We can easily notice some problems related to the form of functional and information aspects are handled in conjunction. Besides, development methods for Web applications give very few attention for the conception, planning, testing and client evaluation stages. This paper presents such problems, presenting a comparison study of Web applications methods based on their main features and supporting mechanisms. Considering the fact that most of the methods have limitations, we present a case study based on a Web portal in order to describe advantages and disadvantages of some selected methods.

Key words: Web development methods, comparison, case study

1 Introduction

The increasing expansion of the Internet can be noticed through the great number of Web applications developed in the last years. This category of application is being used to support a great variety of segments, such as commerce, research and medical activities. In most of them, we can identity particularities which usually have direct influence in the software development process. In addition, some factors as high coupling, continuous evolution and orientation to content directly affect the way the Web applications are built [1].

Web applications may be classified in many ways. For example, Powell et. al [2] and Araújo [3] classify them as informational pages (with or without data forms), database centered applications (with data access by queries and/or by the use of profiles information) and business transactions supporting applications. Some other classifications can be identified in the literature, although they present slight variations. The focus of this paper is on Web applications that support business transactions. This category, in general, corresponds to large applications that can be distributed in several servers. Besides, they can be composed by several software components which perform specific rules to support business accomplishment.

XXXIV Conferencia Latinoamericana de Informática (CLEI 2008), pp. 529-538, 2008.

We can find many initiatives for the definition of development methods specific for Web applications, hereafter called simply "*Web methods*". Examples of these are: HDM [4] RMM [5], OOHDM [6], HMBS/M [7], UWE [8], WebML [9], OO-H [10], W2000 [11], WAE [12], SWM [13] and OOWS [14].

According to Pressman [1], a Web method must comprise the following stages: (i) formulation; (ii) planning; (iii) analysis; (iv) design; (v) pages generation and test; and (vi) costumer evaluation. As a complement, special attention is given to the project phase, which is composed by the design of the content and by the design and production of application's architecture, navigation and interfaces.

However, none of the referred methods include all these stages. Moreover, from the point of view of Verification & Validation (V&V), the few methods that somehow support these activities actually only describe the need for such activities [15]. None of them present mechanisms to support it. Yet, according to systematic review led by Conte et. al [16], there is also the lack of evidences about unification and correspondent representation of the mechanisms used for these methods. On the other hand, all of them present support to the project activity, which also is highlighted by Pressman [1].

We consider that this variety of Web methods certainly poses hindrances for developers to define which one is the best choice according to specific needs or domains. This paper addresses these problems, presenting a comparison study of Web methods based on their main features and supporting mechanisms. For some of them, we highlight the problems earlier referred in this section. Considering the fact that most of methods present limitations, we also present a case study that uses a Web portal to describe the benefits and drawbacks each of the selected methods may bring. We believe that such evaluation may serve as a good starting point to help developers in the difficult task of choosing the adequate method for specific development contexts[3].

The remainder of this paper is organized as follows: Section 2 presents related work. Section 3 introduces some Web methods and Section 4 presents a comparison study among these methods under different perspectives. Next, Section 5 presents the case study where some selected methods were applied to the development of a Web portal. Section 5 also brings a discussion including advantages and disadvantages according to the characteristics of the selected methods. Finally, Section 6 presents our conclusion and future work.

2 Related Work

There are relatively few examples in the literature of studies that investigate Web methods [8, 18–21]. Most research in Web engineering has concentrated on the proposal of methods and tools to serve as a basis for process improvement and higher product quality.

Lee et. al [18] describe an analysis where some attributes of a Web method were measured. The attributes used were: key modeling technique, phases, documentation, source of navigation, approach to identify users' view and semantic richness. However, most of the results described by the authors are subjective, which may have influenced the validity of their results.

[3] In a previous paper [17] we have presented preliminary results of this research.

Koch [8] compares the phases covered by some Web methods. As we can see later in this paper (Sect. 4.1), such kind of comparison may hide some important aspects. The depth in which a method describes a phase can vary significantly. For example, some Web methods only propose a set of textual guidelines for some development phases while others provide supporting tools for the same phases.

Montero et. al [20] present a framework of requirements to survey design methods for hypermedia systems. Such requirements are taken from Software Engineering and Hypermedia Engineering fields. Finally, these requirements are applied to a set of outstanding design methods in order to identify their advantages and disadvantages.

Gu et. al [19] explore the requirements of Web methods to represent the characteristics of Web applications and where the existing methods fail to meet these requirements. Existing limitations tend to be most significant in several key areas, for example, the lack of support for effective modeling of the link between information and functional aspects. Gu et. al's evaluation can be considered as less subjective and imprecise than Lee et. al's, Montero et. al's and Koch's work, although only a few Web methods were presented and some extensions were not taken into account. We extend Koch's and Gu et. al's works in order to include some more recently proposed methods. In addition, we present a case study that provided us with some interesting feedback regarding the methods' abilities.

Baresi and Morasca [21] carried on three empirical studies focusing the effort needed on designing Web applications. The empirical studies were based on the use of W2000, but the hypotheses and results may apply to other methods.

3 Web Method Attributes

Methods to support the development of Web applications must be established in order to enable the management of the software life cycle, including development and maintenance. They should combine traditional techniques and principles of the Software Engineering with specific aspects of the Web. However, the latter are considered neither by traditional software development techniques (e.g. the OMT) nor by traditional modeling languages (e.g. the UML).

Generally, the several methods extend the classic ones with the addition of some model to specify the navigational characteristics of a Web application. A similar procedure is also used for the extension of some development techniques or modeling languages to the context of the Web applications. These models define navigational views of a system, becoming related to their specific groups of users. Usually, graphs are used in order to represent navigational description, which specify the data views of an application and the defined functionality in structural models. Graph nodes represent system views and they can be linked with navigational links (the edges).

In the last years, several methods have tried to extend some known techniques, such as ObjectOriented (OO), Entity-Relationship (E-R) and Structured Analysis (SA) for the Web application context [13]. Moreover, some characteristics such as the just-in-time or the short time requested for its development can suggest the use of other methods, such as the agile ones, since among their principles, one can find the fast delivery of versions of a running product.

Considering the extensions based on known techniques, and based on previous works [8, 15, 18], Table 1 presents the following attributes of some methods found in

the literature, each one in the corresponding table column: (i) the analyzed method; (ii) the employed modeling technique; (iii) the chosen notation for the models; and (iv) the supporting tool for the method.

Table 1. Development methods

Method	Tech.	Notation	Tool
OOHDM	OO	UML and own	OOHDM-Web
UWE	OO	UML and own	ArgoUWE
WebML	E-R and OO	E-R, UML and own	WebRatio
OO-H	OO	UML and own	CASE Tool
WAE	OO	UML and own	n/a
OOWS	OO	UML	OlivaNOVA
HDM	E-R	E-R	n/a
RMM	E-R	E-R and own	n/a
SWM	SA	DFD	ASCENT
W2000	E-R and OO	HDM, UML and own	n/a
HMBS/M	OO and states	Fusion, Statecharts and own	HySCharts and WebScharts

While analysing these methods, we can observe that although the technique and notations vary from method to method, the sequence of activities which should be performed is usually similar. Initially, the domain of the application is analyzed and modeled. In the sequence, the focus is on the structure and the navigation of the application. Finally, the graphical interface is designed.

4 Comparison

In general, any evaluation involving development methods, either Web or traditional applications, is not trivial. The goals of different development methods can relatively differ. While some methods have in mind to support many aspects of the development process, others only detail few parts of it.

In this manner, following we present two comparative studies. We based our comparison on: (i) the stages of the development process; and (ii) information and functional aspects of Web applications.

4.1 Stages of Development Process

Table 2 extends the works of Domingues [15] and Koch [8]. The table considers the stages of the development process proposed for Pressman [1], which are: (i) formulation; (ii) planning; (iii) analysis; (iv) design (architectural, navigational and interface); (v) pages generation; (vi) testing; and (vii) customer evaluation. The following notation is used in this table: C, if the method fully fulfils the development stage; P, if the the stage is partially fulfilled; and blank when the method does not deal with the activity.

At Table 2 we can observe that most of the methods concentrate efforts on analysis and design activities, giving also some special attention to the generation phase. However, it is important to notice that an evaluation like this can omit some important aspects. The details that a method offers to a specific activity and the information

Table 2. Development methods process stages

	Formulation	Planning	Analysis	Design			Generation	Testing	Evaluation
				Arch.	Navig.	Interf.			
HDM				C	C				
RMM				C	C	C	P		
OOHDM			P	C	C	C	P		
HMBS				C	C	C	C	P	
UWE			C	C	C	C			
WebML		C	C	C	C	C	C	P	
OO-H			C	C	C	C	P		
W2000			C	C	C	C			
WAE		C	C	C	C	C	P	P	
SWM	P	P	P	P	P	P	P	P	
OOWS			C	C	C	C	C		

gathered from it can significantly vary. While some methods only consider a set of guidelines in textual way, others provide tools with support to the development stages. As an example, we can consider the OOHDM method, that is composed by graphical representations for the architectural, navigational and interface design stages. These steps are supported by the OOHDM-Web tool. On the other hand, the method only describes some concerns regarding the pages generation stage.

The Table 2 also shows that most of Web methods provide few if no support to V&V activities, in particular testing. For instance, the WebML method focuses only on acceptance tests of non-functional requirements (performance, availability, scalability, security, maintainability). As a complement, regarding revision and inspection activities, no method provide any kind of support to these activities.

4.2 Functional and Information Aspects

For the comparison presented in the Figure 1, in extension to the work of Gu et. al [19], we initially considered the same two aspects considered by them: (i) information; and (ii) functional. Besides the support that a method should offer from the information and functional points of view, another critical issue that should be considered is the ability to make a consistent and a cohesive connection between information and functionalities.

From each point of view, the following groups of abilities were considered:

Group 1 - Information aspects: (i) modeling presentation level concepts; (ii) modeling navigational structure and behavior; (iii) modeling user interactions with information; (iv) modeling user roles and user groups; and (v) modeling content.

Group 2 - Functional aspects: (i) modeling integration and connectivity; (ii) supporting pattern modeling; (iii) representing concepts in a technology neutral fashion; and (iv) modeling sophisticated system functionality.

Group 3 - Interconnection of information and functional aspects: (i) modeling business domain concepts; (ii) linking business model with technical architecture; (iii) linking information architecture with functional architecture; (iv) maintaining system integrity; (v) supporting understanding and communication; (vi) being process independent; and (vii) supporting Web system life cycle management.

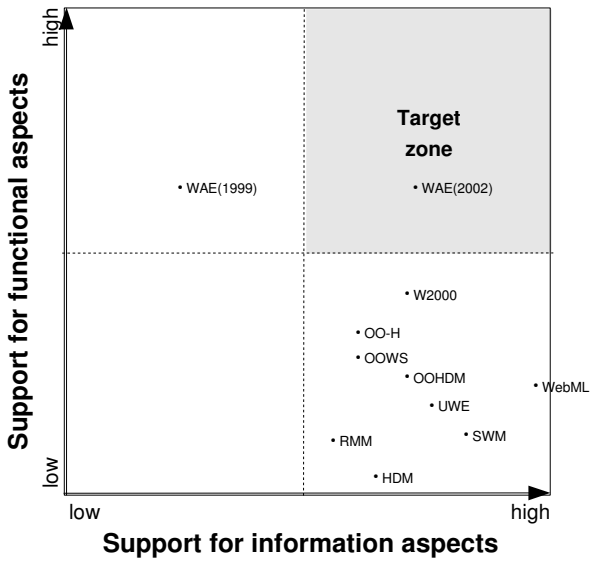

Fig. 1. Development method evaluation:information and functional aspects

In this analysis we included the OO-H, OOWS, SWM and WAE (2002) methods, since they give prominence to information or functional aspects according to Figure 1. Table 3 presents the requirements checklist for these methods. Notice that the same notation used in Table 2 is used here.

Among the analyzed factors, our main findings regarding method limitations are: (i) inability to model complex functionalities; (ii) disconnection between functional aspects and information aspects; (iii) disconnection between business model and technical architecture; (iv) potential misuse of UML extension mechanisms; and (v) partial ability to support software life cycle management. Notice that these problems are basically the same as the ones raised by Gu et. al [19].

Considering the disconnection between information and functional aspects, we highlight that most of the methods have their origin based on the hypermedia (namely, HDM, RMM, OOHDM, W2000, UWE and WebML). They usually provide interesting and sufficiently rich characteristics to support information aspects of Web applications, specially for the navigational and presentation modeling. On the other hand, they offer a very limited capacity to support more complex functionalities during the design stage of an application. The WAE method extends the UML notation to support operations and interactions of Web applications. From this point of view, WAE is the only one that fully deal with functional and information aspects, thus reaching the target zone (see Figure 1).

5 Case Study

We conducted a case study aiming at having a better understanding of some development methods discussed in previous sections of this paper. This case study consisted

Table 3. Requirements Checklist

#	**Requirements for modelling languages**	**WAE**	**OO-H**	**OOWS**	**SWM**
1.i	Model presentation level concepts	P	C	C	C
1.ii	Model navigational structure and behaviour	C	C	C	C
1.iii	Model user interactions with information	P	P	P	
1.iv	Model user roles and groups	P	P	P	P
1.v	Model content	C			
2.i	Model integration and connectivity	C	C	C	P
2.ii	Support Pattern Modelling	P	C	C	
2.iii	Represent concepts in a technology neutral fashion	C	C	C	C
2.iv	Model sophisticated system functionality	C	P	P	P
3.i	Model business domain concepts	C	P	P	P
3.ii	Link business model with technical architecture	C	C	P	
3.ii	Link information architecture with functional architecture	P	P	P	
3.iv	Maintain system integrity	P			
3.v	Support understanding and communication	C	P	P	P
3.vi	Be process independent	C			
3.vii	Support web system life cycle management	P	P	P	P

of the modeling of a specific Web application using different methods. The application in question is a knowledge portal of testing tools, criteria and strategies for Web Applications.

In the first part of the case study, some methods were selected considering the evaluation carried out on the Section 4.2. We selected the ones that give prominence to information or functional aspects. These methods are the WebML [9], which it is the best rated regarding information aspects, the W2000 [11] and the OO-H [10], which are closer to the target zone (see Figure 1), and the WAE (2002) [12], that is the only one inside the target zone.

After the selection of these methods and the definition of requirements for knowledge portal, for each method we performed the proposed activities. Considering that some stages consist of well known activities and artifacts, these artifacts were reused during the modeling for several methods. Examples of these artifacts are the use cases and the class diagrams. The stages and the artifacts that are not common for more than one method are following commented.

In the WAE method, the use of UML diagrams such as the activities diagram and sequence diagram enable the detailed modeling of functional aspects. To support these diagrams, the UX (User eXperience) model is proposed. The UX model makes possible to treat several characteristics of information aspects. It is similar to a class diagram, however it includes new stereotypes and associations that enable the representation of system screens and the way that they can be browsed by users. Based on this, the UX model represents fine-grained details of: (i) elements which are inside the navigational context of a Web application; (ii) user views; and (iii) information present in each application page.

In the WebML method, a specific notation is used to represent information aspects. These aspects are represented in the structural, hypertext, presentation and personalization models. The structural model may be seen as a combination of class diagram and E-R notations. The hypertext model defines Web pages and their links. It is divided in navigational and composition models. The presentation model describes page layouts. The personalization model describes the users and groups they make

part of, being modeled with predefined entities and used to store specific information of an user or a group.

In the W2000 method, the use of UML extensions is proposed. Theses extensions are compatible with the HDM models. The use case diagrams are divided in functional and navigational use cases. The functional use cases are the same used in the UML. The navigational ones contain information about the roles and rules for each user. Navigation and information aspects are represented by the hypermedia project, which is divided in information and navigational design. The information design organizes information by using types of entity and types of semantics association. The types of entity describe observable object classes by the users. The semantic types of association are the binary relationship in a specific domain that occurs between the types of entity. The navigational design defines the nodes and navigational links. The nodes are derived from the entities, components and associations, normally mapped as Web pages or as functional blocks in a Web page. Links are the associations between the represented nodes.

The OO-H method also uses class diagram for the conceptual modelling. For the navigational modeling a top-down approach is used. In this phase an initial diagram represents all the possible paths that can be reached by a user. From this diagram, each path is detailed in a new diagram, which includes elements from the class diagram.

5.1 Discussion

The knowledge portal was modeled using the four methods. Based on that, it was possible to notice the advantages and/or disadvantages among them. It is important to notice that the selected methods use several models based on UML notation, which is a very common standard used in several kinds of software modeling.

The WAE method, that uses a UML extension with specific stereotypes and notations, has shown to be very complete and consistent, specially when considering its ability of modeling the server logic through the use of communication between server and client inside pages in a Web application. Moreover, the human-computer interaction was well represented by the use of UX diagrams, which include several elements, such as forms, boxes, frames and others important elements for Web applications.

The WebML method, as well as the WAE, also uses UML resources for the modeling of domain elements. Besides, their models provide a good representation for navigation and hypermedia elements of a Web application. However, several artifacts requires an own notation, what results in a steep learning curve.

The W2000 method divides the modeling in two stages: “in-the-large” and “in-the-small”. From one stage to another, several refinements are required. However, these is one among other forms of employment of the W2000 method. It may, for example, support incremental development. Therefore, this should not be considered as a differential of the method, since de development process may vary in other methods as well. Moreover, the extensions created from the UML are not representative when it is necessary to create the navigational and information models. Some extensions, such as the scenarios or sequence diagrams, do not add relevant information to the original diagrams.

The OO-H method uses elements which are common to Web applications such as forms, buttons and interface elements. These element allows one for a complete and

detailed navigational model. An interesting point in this model is how a link can be used. There are links that indicate services (activities performed by a class) or links that indicate an answer (tasks executed by other methods).

6 Conclusions and Future Work

As long as people and business become dependent on Web applications, these need to be more reliable and to be executed in a proper way. However, in the majority of the cases, the development approach employed for Web applications is ad-hoc, in which maintenance is continuously carried out without any control. In general, the development of Web applications does not enforce any severe, systematic approach and adequate quality assurance. In the absence of a systematic Web application development process, many problems can be identified, related to development, deployment, operation and maintenance.

Regarding the way these applications are created, a number of specific development methods have been proposed lately. In general, they aim at dealing with the current needs of this category of application. These initiatives try to move identified problems to a more controlled stage, in turn trying to improve the quality of Web applications development. Some of these methods also bring tooling support. However, most of them consider a different number of stages and activities, as well as the use of different techniques and notations.

Considering these several issues, this paper presented some comparative studies involving Web methods. We can easily notice that they give special attention to design-related stages (e.g. architecture, navigation and interface). On the other hand, they do not handle other though not less important stages of software development, such as planning, requirements analysis and V&V. In many cases, the support that a method offers for a specific stage varies significantly. Another observed characteristic is the lack of mechanisms to support the proposed methods, specially when we consider an application under both functional and information points of view. In this context, the WAE method is the only that contemplates these aspects satisfactorily.

The results initially presented in this paper have been serving as a basis for other ongoing works, such as: (i) the extension of the case study presented for other Web methods; (ii) the conduction of others case studies involving the same methods with other Web applications; (iii) the development of a knowledge portal of testing tools, criteria and strategies for Web Applications testing; (iv) the definition of a framework that comprises a set of guidelines aiming at evaluating and/or specifying Web methods; and (v) the definition or extension of a development method regarding the reported problems, specially with respect to V&V activities.

References

1. Pressman, R.S.: Software Engineering – A Practitioner's Approach. 6.^{a} edn. McGraw-Hill (2005)
2. Powell, T., Jones, D., Cutts, D.: Web Site Engineering: Beyond Web Page Design. Prentice Hall (1998)
3. Araújo, A.C.M.: Framework de análise e projeto baseado no RUP para o desenvolvimento de aplicações web. Master's thesis, Cin/UFPE, Recife, PE (2001)

4. Garzotto, F., Paolini, P., Schwabe, D.: HDM - a model-based approach to hypertext application design. ACM Transactions on Information Systems **11**(1) (1993) 1–26
5. Isakowitz, T., Stohr, E.A., Balasubramanian, P.: RMM: a methodology for structured hypermedia design. Communications of the ACM **18**(8) (August 1995) 34–44
6. Rossi, G.: Um Método Orientado a Objetos para o Projeto de Aplicação Hipermídia. PhD thesis, PUC-Rio, Rio de Janeiro, RJ (1996)
7. Carvalho, M.R., Oliveira, M.C.F., Masiero, P.C.: HMBS/M – an object oriented method for hypermedia design. In: V Simpósio Brasileiro de Multimídia e Sistemas Hipermídia (SBMIDIA'99), Goiânia, GO (June 1999) 43–62
8. Koch, N.: Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process. PhD thesis, Ludwig-Maximilians University, Munich - Germany (December 2000)
9. Ceri, S., Fraternali, P., Bongio, A.: Web modeling language (WebML): a modeling language for designing web sites. In: 9th International World Wide Web Conference, Amsterdam (May 2000) 1–22
10. Gómez, J., Cachero, C., Pastor, O.: Conceptual modeling of device-independent web applications. IEEE MultiMedia **8**(2) (2001) 26–39
11. Baresi, L., Garzotto, F., Paolini, P.: Extending uml for modeling web applications. In: 34th Annual Hawaii International Conference on System Sciences. (January 2001) 1–10
12. Conallen, J.: Buildind Web Applications with UML. 2nd. edn. Addison-Wesley (2002)
13. Griffiths, G., Hebbron, B.D., Lockyer, M.A., Oates, B.J.: A simple method & tool for web engineering. In: 14th international conference on Software engineering and knowledge engineering (SEKE'02), New York, NY, USA, ACM Press (2002) 755–762
14. Fons, J., Pelechano, V., Pastor, O., Albert, M., Valderas, P.: Extending an OO method to develop web applications. In: The Twelfth International World Wide Web Conference. (2003)
15. Domingues, A.L.S.: Aplicações *Web*: Definição e Análise de Recursos de Teste e Validação. PhD thesis, ICMC/USP, São Carlos/SP - Brasil (2005) Unfinished.
16. Conte, T., Mendes, E., Travassos, G.H.: Processo de desenvolvimento para aplicações web: Uma revisão sistemática. In: XI Simpósio Brasilieiro de Sistemas Multimídia e Web, Poços de Caldas, MG (2005)
17. Domingues, A.L.S., Bianchini, S.L., Costa, M.L.S., Ferrari, F.C., Maldonado, J.C.: Web application development methods: A comparison. In: Proceedings of the 13th Brazilian Symposium on Multimedia and the Web (Webmedia 2007). (2007) 1–4
18. Lee, H., Lee, C., Yoo, C.: A scenario-based object-oriented methodology for developing hypermedia information systems. In: Hawaii International Conference on Systems Sciences (HICSS'98). (1998) 47–56
19. Gu, A., Henderson-Sellers, B., Lowe, D.: Web modelling languages: the gap between requirements and current exemplars. In: The Eighth Australian World Wide Web Conference (AusWeb'02), Queensland, Australia (July 2002)
20. Montero, S., Díaz, P., Aedo, I.: Requirements for hypermedia development methods: A survey of outstanding methods. In: The Fourteenth International Conference on Advanced Information Systems Engineering (CAISE 2002). (2002) 747–751
21. Baresi, L., Morasca, S.: Three empirical studies on estimating the design effort of web applications. ACM Transactions on Software Engineering and Methodology **16**(4) (2007) 1–40