Toward a Comprehension View of Web Engineering

Semia Sonia Selmi, Naoufel Kraiem, and Henda Ben Ghezala

ENSI, National School of Computer Science, 2010 Manouba, Tunisia
Semiasonia.selmi@riadi.rnu.tn
Naoufel.kraiem@ensi.rnu.tn
Henda.BG@cck.rnu.tn

Abstract. The paper is an attempt to explore some of the issues underlying Web applications development through the use of disciplined approaches. We first present the proposed Web engineering framework which suggests considering web engineering along four different views. Each view is capturing a particular relevant aspect of Web engineering. Motivations for developing the framework are three fold: (a) to help understand and clarify the Web engineering domain, (b) to guide in classifying and comparing both web applications and approaches and (c) to help researchers to identify new research axes. Next, we briefly present evaluation of 7 different Web-based approaches according to the Web engineering framework.

1 Introduction

The technological evolution of the last decade has made the World Wide Web the ideal platform for the development of Web-based hypermedia applications and the primary support for their delivery. Indeed, an enormous number of applications have been developed and their widespread acceptance points to the effectiveness of Web design approaches. However, current applications often fail since their development is often on an ad-hoc basis, without the support of appropriate methodologies able to manage the high complexity of information.

Obviously, we have little understanding about how web applications should be developed. For example, there is no consensus on which approach to be used for development. There is also little evidence about their effectiveness and even less idea about how they are.

Consequently, considerable attention has been given to Web engineering, a new discipline proposed to provide a systematic and disciplined approach for developing, documenting and maintaining Web/hypermedia applications.

Web engineering is a rather a new research area, so, studying and understanding deeply this discipline need a web engineering framework. We propose a framework in which we consider web engineering through four different view-points each one capturing a particular aspect of this discipline.

Motivations for developing the framework are three fold: (a) to help understand and clarify the web engineering domain, (b) to guide in classifying and comparing both Web applications and approaches and (c) to help researchers identify new research axes. The latter is an important issue since the whole Web engineering field is relying on the two fundamental concepts namely *Web applications* and *Web-based hypermedia design approaches*.

D. Lowe and M. Gaedke (Eds.): ICWE 2005, LNCS 3579, pp. 19–29, 2005.

This paper is an attempt to explore some of the issues underlying Web applications development and to propose a framework. The remainder of this paper is organized as follow. Section 2 is an overview of the framework which is further detailed in section 3. Section 4 reviews 7 current web development approaches evaluated according to the framework. Finally, conclusions are drawn in section 5.

2 Framework Overview

The development of a Web application is a multi-faceted activity, involving not only technical but also organisational, managerial and even social and artistic issues [11]. Web application development refers to a set of activities applied in order to develop a web application of high quality having awaited characteristics, and to carry out this development efficiently and coherently. Obviously, goals of Web application development introduce two basic concepts namely Web application as a product and the used Web approach as a process.

The proposed Web engineering framework is based on both concepts and implies considering them along four different views, each view captures a particular relevant aspect of Web engineering.

As shown in Fig. 1, framework is composed of:

- N*ature* view deals with the classification of both web applications and web-based hypermedia methods applied for the design and the development of web applications.
- *Form* view includes representations of methods at different levels of detail.
- *Purpose* view deals with intentional aspects. It concerns goals which we attempt to reach in the web engineering field.
- *Development* C*ycle* view deals with the web applications development process and their enactment.

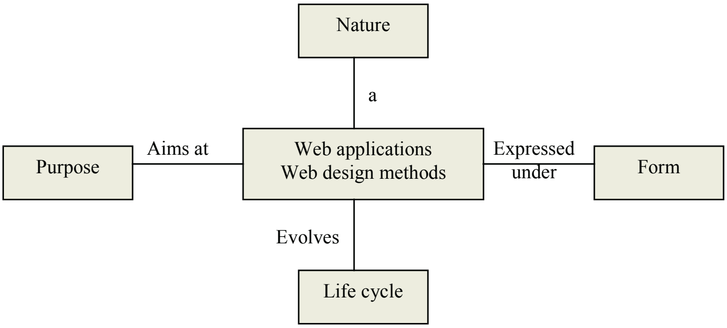

Fig. 1. Web Engineering Framework

We have adopted a faceted classification approach similar to the one proposed by [27] in Requirements Engineering. Each view is associated with a set of facets which are considered as viewpoints or dimensions suitable to characterize and classify approaches and/or applications according to this view.

A metric is attached to each facet which is measured by a set of relevant attributes. Both web applications and methods are positioned in the framework by affecting values to the attributes of each facet. Attribute values are defined within a domain which may be a predefined type (Integer, Boolean, etc), an enumerated type (ENUM{x, y, z}), or a structured type (SET or TUPLE).

The multi-facet and multi-view approach adopted makes it possible to look at web engineering in a comprehensive way. Facets provide an in-depth description of each aspect of Web engineering whereas aspects give a view of Web engineering in all its diversity.

3 The Proposed Framework

3.1 The Nature View

The Nature view is characterized by two facets namely the *application nature* facet and the *method nature* facet.

Application Nature Facet

Many classifications of web applications are referenced in literature. For instance, classifications proposed in [14] [13] and [1] are generally based on functionality criteria and are, consequently, considered as specific classifications.

A more general classification is proposed by both [7] and [9]. According to [7], a distinction can be made between a *Web application* and a *Web site*. The web application uses a web site as the front end to a more back office application.

[9] proposes a similar distinction by identifying the *kiosk type* and the *application type*. A kiosk web site mainly provides information and allows users to navigate through that information. Whereas, an application web site is an information system where users process data, communicate and collaborate with other users.

As classification in [9] is largely referenced in literature such as in [17] [31] and [9], we keep this classification and define the following application nature attribute having the same name of the facet:

Application nature: ENUM {kiosk type, application type}

Method Nature Facet

Some researchers have attempted to classify web design approaches such [32] in which approaches are classified into 4 categories: *Resource-oriented* approaches*, Site-oriented* approaches*, Design-oriented* approaches*, Model-based* approaches.

[9] proposed another classification of web design approaches based on three categories: *Data-driven* approaches, *User-centered* approaches*, User-driven* approaches.

A web design approach can be classified in only one category according to the second classification. As the latter is the most referenced in literature, we define the following attribute having same name with the facet:

Method nature: ENUM {data-driven, user-driven, user-centered}

3.2 Form View

The form view is composed of the three following facets: Models facet, Notation facet and Abstraction facet.

Models Facet

The model facet is concerned with the content of a web design approach.

With the rapid increase of web applications complexity it becomes increasingly important for web design approaches to provide different modelling artefacts that support various viewpoints. When designing complex web applications, designers can look at the application from different but inter-related perspectives. They can break down applications into manageable pieces. Indeed, approaches consider design process in terms of process phases and their deliverables, often models.

The following models[1]: *Conceptual model*, *Navigation model* and *Presentation model* are commonly delivered. Besides, and due to the evolution of the web and web applications, other phases are recognized delivering the following design models: *Requirements analysis model*, *User model*, *Adaptation model* and *Business process mode and Business model* (required especially for e-commerce applications).

Thus, in order to capture aspects considered during web applications design, the Models facet introduces the following attribute with the same name:

Models: SET (ENUM {Requirements analysis model, Conceptual model, Navigation model, presentation model, User model, Adaptation model, Business process model, Business model})

Notation Facet

In order to support the representation of application features during development lifecycle, notations with different levels of formality and abstraction are used.

To express structural features, the best-known conceptual data models, like E-R Model and various objects models [29] are mostly used. Various approaches belonging to the hypermedia field are proposed to enrich traditional conceptual models with new concepts.

For modelling the navigation, most approaches employ notations and techniques proposed for the more general problem of human-computer interaction specification [10] and extend data models with navigation primitives.

For presentation modelling, most of methods, except methods based entirely on UML, use principally proprietary formalisms and notations combined sometimes with standard notations.

The domain of the attribute Notation captures the notation used in a method.

Notation: ENUM {Standard, Proper, Mixture}.

Abstraction Facet

The Abstraction facet allows capturing abstraction levels in which methods are described. Depending on its level of abstraction, the method component will be reused as such or will be instantiated before being assembled in the method under construction. Approaches based on meta-modeling mechanism are at the meta-type level. The abstraction facet has one attribute *level* having values in the following enumerated domain:

Level: SET (ENUM {type, meta-type}).

[1] Models mentioned are deliverables of the three common design phases referenced in [11]

3.3 Purpose View

The purpose view deals with goals assigned to web application methods. This issue is associated with the *purpose* facet. As applications change and evolve continuously in time, a method should support this evolution. This aspect is captured by the facet named *Method management policy*. These two facets are described in the following.

Purpose Facet

Synthesis of studies in the Software Engineering [20], [8] and Information Systems communities [3],[22] and [25] have shown two main aims of web application design methods:

- Prescriptive: they prescribe how the process should be accomplished.
- Descriptive: they study existing processes and describe how the process is carried out.

However, some approaches mix descriptive and prescriptive strategies [22]. A web application design approach can be classified according to the role that plays in the Purpose facet:

Purpose: SET (ENUM {descriptive, prescriptive})

Method Management Policy Facet

Given the rapid changes in context and in user requirements when considering the Web, an environment where change in both technology and user requirements is a standard part of life, applications should have the ability to evolve. This evolution should be supported during their development. Design evolution should be supported with automatic propagation of the modifications from one step to another during development process. The use of structured techniques and the product of a process tracings and separation between the following aspects of design constitute a solid base to method evolution.

As in Software Engineering, reuse is also an aspect of this view. Reuse is a strategic tool for reducing the cost and improving the quality of hypermedia design and development. It consists in taking advantage of any of the efforts done for previous works to reduce the needed effort to achieve a new one [19]. Reuse can occur at any level of hypermedia development. It may concern data, software components conceptual schemas, design schemas, content and physical application pages as well as design experience. The most common form of reuse on the web is content reuse.

Thus, a web design method might be positioned within the Method Management Policy facet with the two following attributes *Evolution* and *Reuse* which allow to determine respectively even the method supports evolution and reuse or not.

Evolution: Boolean
Reuse: Boolean

3.4 The Development Cycle View

Lifecycle Coverage Facet

Several lifecycle models of a web application have been proposed in the literature such as in [30], [2], and [11]. However, typical activities involved in the construction of a web application can be partially obtained from the lifecycle models of traditional Information Systems and enriched with specific activities. The lifecycle model used as a reference in this paper was proposed in [11].

Note that defining the lifecycle coverage facet allows one to determine activities considered during the development of a web application.

Life cycle Coverage: SET (ENUM {Requirement Analysis, Conceptualisation, Prototyping & validation, Implementation, Evolution & Maintenance})

Construction Technique Facet

In the web development context, we find the construction techniques of traditional Information Systems domain, exploring the meta-modelling aspect and using languages. Meta-modelling consists in identifying common and generic characteristics to different applications and representing them by a system of generic concepts. It uses two principal techniques: instantiation and assembling. It is to note that most of web applications are developed based on developer's experience, we can that they are the result of an ad-hoc technique of construction.

Technique: ENUM {Instantiation, Assembly, language, ad-hoc}

Interaction Facet

Compared with traditional software applications, web applications tend to provide much more sophisticated interactions with users.

Dynamic description and transformations occurred when users interact with application should be supported by the web design approach. Thus, an approach needs to provide the ability to model these different interactions in a complete way so that users' interactions can be captured, designed and implemented.

We introduce a Boolean attribute *Interaction* which allow determining if method considers interaction of users with application or not do.

Interaction: Boolean

Enactment Support Facet

Besides the construction process of web applications, the development view deals also with their enactment.

CAWE category (Computer Aided Web Engineering) provides best development lifecycle coverage by applying design modelling and code generation techniques. We find all basic principles of software engineering. Benefits are comparable to those of CASE tools, we cite for instance reduction of effort and reuse. Efforts have been conducted in last years to the design and the development of prototype of hypermedia and web design tools.

Consequently, this facet has an attribute *tool support* that allows knowing the tool used.

Tool support: text

Dynamic Generation Facet

Information content of web applications is stored in pages and users can request a page by its name or can access through path.

In some situations, page content is assembled at run time from information stored in the data base or a repository. It can also be generated from loaded modules e.g. CGI. Web sites using this strategy are dynamic web sites. These have the advantage to keep the content up to date and synchronised with the data of the data bases. All these aspects should be taken into account by the design method when developing applications.

According to this facet, methods are classified into those that consider dynamic content generation and those that not do. This facet is described by a Boolean attribute.

Dynamic generation: Boolean

Adaptation Facet

Many researches have been focused on adaptation forms such as [4] [6] [19]. [33] completes these adaptation dimensions by namely adaptation based on functionality and adaptation based on management of material conditions of exploitation. Summarizing, adaptive dimensions are following: adaptive content, adaptive navigation, adaptive presentation, Adaptive functionality, Management of Material conditions of exploitation.

Adaptation: SET (ENUM {content, navigation, presentation, functionality, material condition of exploitation})

4 Review of 7 Web Design Approaches According to the Framework

We propose, in this section, to illustrate the use of the web engineering framework through the evaluation of the 7 following web-based hypermedia design approaches: Relationship Management Methodology (RMM) [17], Object-Oriented Hypermedia Design Model (OOHDM) [30] [15], Hypermedia Flexible Process Model (HFPM) [22], UML-based Web Engineering (UWE) [19], the method proposed by Takahashi [31], Web Sites Design Method (WSDM) [9] and WebML [6].

We aim to get a large picture of the web engineering area and to help understand currently developed web-based hypermedia approaches.

Table 1, Table 2 and Table 3 provide instantiation of the 7 approaches according to the four views of the proposed web engineering framework namely: Nature, Form, Purpose and Development Cycle.

Several conclusions can be obtained from we can get both strength points and limitations of evaluated approaches.

Each of these proposed approaches has its strength points. We can notice, for instance, that except WSDM all approaches are proposed for the development of complex web applications and cover the whole life cycle of web applications. They support reuse and are able to evolve when application environment change. This is made possible since approaches consider the different design steps separately.

Although proposed approaches have strength, some limitations can be identified such as:

The Inability to Model Business Processes: Most existing modeling approaches do not address the modeling of functionality or business processes. Except WSDM,

Table 1. Instantiation of 7 web-based hypermedia approaches according to Nature view and Form view of the Web Engineering Framework

	Nature View		**Form View**		
	Application Nature	**Method Nature**	**Models**	**Notation**	**Abstraction**
RMM	applications	Data-driven	Conceptual M. Navigational M. Presentation M.	Mix	Type
OOHDM	applications	User-driven	Req. Analysis M. Conceptual M. Navigational M. Presentation M.	Mix	Type
Takahashi M.	applications	User-driven	Req. Analysis M. Conceptual M.	Mix	Type
HFPM	applications	User-driven	Req. Analysis M. Conceptual M. Navigational M. Presentation M.	Mix	Type
UWE	kiosques	User-centred	Req. Analysis M. Conceptual M. Navigational M. Presentation M. User Model Adaptation M.	Standard	Type
WSDM	kiosques	User-centred	Req. Analysis M. Conceptual M. Navigational M. User M. Business process M.	Mix	Type
WebML	applications	Data-driven	Conceptual M. Navigational M. Presentation M. User Model Adaptation M.	Mix	Type

Table 2. Instantiation of 7 web-based hypermedia approaches according to Purpose view of the Web Engineering Framework

Purpose View			
	Purpose	**Method management Policy**	
		Evolution	**Reuse**
RMM	Presc.	Yes	Yes
OOHDM	Presc.	Yes	Yes
Takahashi M.	Presc.	Yes	Yes
HFPM	Presc. + Desc.	Yes	Yes
UWE	Presc.	Yes	Yes
WSDM	Presc.	Yes	Yes
WebML	Presc.	Yes	Yes

approaches focus has been on the organization aspects, navigation and presentation modalities. Some others approaches are user-centered and then consider moreover user viewpoint. However, none has addressed Business processes viewpoint which has been consistently overlooked.

Table 3. Instantiation of 7 web-based hypermedia approaches according to Development cycle view of the Web Engineering Framework

Development Cycle View						
			Enactment Support			
	Life cycle Coverage	Technique	Tool support	Dynamic Generation	Interaction	Adaptation
RMM	Conceptualization Design Prototyping & validation Implementation	/	RMCase	No	No	No
OOHDM	Req. analysis Conceptualization Design Implementation	/	OOHDM-Web	Yes	Yes	Yes (relatively)
Takahashi M.	Req. analysis Conceptualization Design Implementation Maintenance	/	WebArchitect PilotBoat	No	No	No
HFPM	Req. analysis Conceptualization Design Implementation Maintenance	/	/	No	No	No
UWE	Req. analysis Conceptualization Design Implementation Maintenance	/	/	No	Yes	Yes
WSDM	Req. analysis Conceptualization Design Implementation	/	/	No	No	Yes (relatively)
WebML	Req. analysis Conceptualization Design	language	WebRatio Site development studio 24	No	Yes	Yes

The Inability to Support Various Abstraction Levels: Modeling approaches need to provide modeling artifacts at different abstraction levels. The need for various abstraction levels is reflected in the importance to be guided from high to low abstraction level.

The Not Use of Standard Notation: Except UWE, existing modelling approaches address main design phases through the use of different notations in most cases proprietary. However, it desirable to adopt known and standard notations. This facilitates both communication between involved people in design and maintenance phase. The latter plays an increasingly important role in comparison with conventional software systems.

Inability to Model Interaction Aspects: Not all existing modelling techniques support interaction aspect during web design. However, modelling techniques need to provide ability to model users' interactions with web applications. They should specify transformations occurred when user interacts with application, objects behaviour after external events and dynamic descriptions.

Inability to Support Dynamic Generation Content: Major web applications are becoming complex in term of information and then need to be dynamic, that is content is assembled at run time from information stored. Information sources to which a web application can connect may include databases, file servers, document repositories, etc. Consequently, modelling approaches need to support dynamic generation of content.

5 Conclusion and Further Work

More recently, the web engineering community advocates and emphasizes the use of disciplined approaches for the development of web applications.

Our study has shown that both web design approaches and web applications can not be treated adequately with simple predicate-based classification techniques. However, here is a need for a four-dimensional framework to well describe web engineering discipline. Through the notion of dimensions and facets, we are able to successfully capture the global view and the more detailed view of web engineering respectively.

References

1. Amitay, A., Carmel, D., Darlow, A., Lempel, A., Soffer, A.: The Connectivity Sonar: Detecting Site Functionality by Structural Patterns. J. Digit. Information. 4 (2003) Issue 3
2. Atzeni, P., Mecca, G., Merialdo, P.: To Weave the Web. 23rd Conference on Very Large Data Bases (VLDB'97), pp. 206-215, Athens, Greece, August 26-29
3. Brinkkemper, S.: Formalisation of information systems modeling. Ph. D. Thesis (1990), University of Nijmegen, Thesis Publishers
4. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. Int. J. of User Modeling and User-Adapted Interaction (1996). Kluwer Academic Publishers 6, 2-3, 87-129
5. Chen, P.P.: The entity-relationship model: toward a unified view of data. ACM TODS (1976) 1, 1, 9-36
6. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): a modeling language for designing Web sites. 9th International World Wide Web Conference (Www9), Amsterdam, Netherlands, May 15-19, 2000
7. Conallen, J.: Modeling web Application Architectures with UML. Communications of the ACM. 42 (1999) N°.10, pp.63-70
8. Curtis, B., Kellner, M. I., Over, J.: Process Modeling. Communications of the ACM. 35 (1992) N° 9, pp 75-90
9. De Troyer, O.M.F., Leune, C.J.: WSDM: a User Centered Design Method for Web Sites. WWW7 Conference (1998), Brisbane
10. Fraternali, P., Paolini, P.: A conceptual model and a tool environment for developing more scalable and dynamic Web applications. EDBT'98. 421-435, Valencia, Spain
11. Fraternali, P.: Tools and Approaches for Developing Data-Intensive web Applications: a Survey. ACM Computing Surveys (1999)
12. Garzotto, F., Paolini, P., Schwabe, D.: HDM. A model-based approach to hypertext application design. ACM Transactions on Information Systems 11(1993), N°1 1-26
13. Ginige, A.: Web Engineering: An Introduction. IEEE Multimedia (2001)

14. Gnaho, C.: Définition d'un Cadre Méthodologique pour l'Ingénierie des Systèmes d'Information Web Adaptatifs. Ph. D. Thesis (2000) Paris1 University
15. Güell, N., Schwabe, D., Vilain, P.: Modeling Interactions and navigation in Web Applications. World Wide Web and Conceptual Modeling Workshop, ER'00 Conference, LNCS 1921, Springer, Salt Lake City (2000).
16. Halasz, F., Schwartz, M.: The Dexter Reference Model. Hypertext Standardization workshop 95-133, Gaithersburg, Maryland: National Institute of Standards and Technology Publication 500-178 (1990)
17. Isakowitz, T., Stohr, E.A., Balasubramanian, P.: RMM: a Methodology for Structured Hypermedia Design. Communications of the ACM, Vol. 38, N°.8, pp. 34-44, (1995)
18. Jones, D., T. lynch: A Model for the design of web-based systems that supports Adoption, Appropriation and Evolution. 1st ICSE Workshop on Web Engineering, Murugesan, S., Deshpande, Y. (eds), Los Angeles, pp 47-56.
19. N. Koch: Software Engineering for Adaptive Hypermedia Systems-Reference Model, Modelling Techniques and Development Process. Ph.D Thesis (2001), Fakultät der Mathematik und Informatik, Ludwig-Maximilians-Universität München, December .
20. Lonchamp, J.: A structured Conceptual and Terminological Framework for Software Process Engineering. International Conference on Software Process (1993)
21. J. Nanard and M. Nanard: Toward an Hypermedia Design Pattern Space. Hypertext'99 Workshop on Design Pattern in Hypermedia
22. Olsina, L.: Building a Web-based information system applying the hypermedia flexible process modeling strategy. 1st International Workshop on Hypermedia development, Hypertext'98
23. Papazoglou, M.P., Yang, J.: The Role of services and Transactions for Integrated value chains. chapter IX, Idea Group Publishing, (2002)
24. Paterno, P., Mancini, C.: Designing Web User Interfaces for Museum Applications to Support different Types of Users. International Conference about Museums and the Web, March 12-14, New Orleans, LA, 1999, pp. 75-86
25. Rolland, C., Plihon, V.: Using generic chunks to generate process models fragments. 2nd IEEE Int. Conf. on Requirements Engineering, ICRE'96
26. Rolland, C., Prakash, N.: A proposal for context-specific method engineering. IFIP WG 8.1 Conference on Method Engineering, Chapman and Hall, pp 191-208, Colorado Spring, 1996
27. Rolland, C., Ben Achour, C., Cauvet, C., Ralyté, J., Sutcliffe, A., Maiden, N.A.M., Jarke, M., Haumer, P., Pohl, K., Dubois, Heymans, P.: A proposal for a scenario classification framework. Requirements Engineering Journal, 3, 1, 1998
28. Rossi, G., Schwabe, D., Garrido, A.: Design reuse in hypermedia applications development. 8th ACM conference on Hypertext (1997), ACM press,, pp. 57-66
29. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Loresen, W.: Object-oriented modeling and design. Prentice Hall international, (1991)
30. Schwabe, D., Rossi, G., Barbosa, S.: Systematic Hypermedia Application Design with OOHDM. ACM-Hypertext96 (1996)
31. Takahashi, K., Lang, E.: Analysis and Design of Web based Information Systems. Sixth International World Wide Web Conference, (1997)
32. Vigna, G., Coda, F., Garzotto, F., Ghezzi, C.: A Generative World Wide Web object-oriented Model. Politecnico di Milano, Technical Report (1997)
33. Villanova-Olivier, M.:Adaptabilité dans les systèmes d'information sur le Web: Modélisation et mise en oeuvre de l'accès progressif. Ph. D. Thesis (2002), Grenoble, France