IBIMA Publishing
Communications of the IBIMA
http://www.ibimapublishing.com/journals/CIBIMA/cibima.html
Vol. 2010 (2010), Article ID 306853, 20 pages
DOI: 10.5171/2010.306853

A Model for Characterizing the Salient Directions of Evolution in the Engineering of Web Applications

Pankaj Kamthan[1] and Nazlie Shahmir[2]

[1] Concordia University, Montreal, Quebec, Canada
[2]Shaw Cablesystems G.P., Calgary, Alberta, Canada

Abstract:

The development and sustenance of Web Applications is viewed from an engineering perspective. The evolution of the developmental environment of Web Applications has been multi-directional. This paper provides a conceptual characterization of these directions, relationships between these directions, and their implications towards academia and industry. The consequences of a commitment to these directions, along with the role of international standards, are considered. The challenges faced by Semantic Web Applications and Social Web Applications are briefly outlined.

Keywords: Conceptual Modeling, Experiential Knowledge, Quality, Stakeholder, Web engineering

Introduction

The Internet, particularly the Web, has played an increasingly vital role towards communication, information, and entertainment. This evidently has had an impact on how Web Applications are perceived, developed, and managed.

It is known (Kruchten, 2004) that conventional engineering practices cannot be simply mapped to software engineering without first understanding the nature of software, and the same applies to Web Engineering (Ginige & Murugesan, 2001). This paper builds upon previous work (Kamthan & Shahmir, 2010) and, from the perspective of Web Engineering, identifies and elaborates the characteristics that make the developmental environment of Web Applications unique, and analyzes some of their theoretical and practical implications. It also identifies the practical challenges that academic institutions and industrial organizations can face in doing so.

The rest of the paper is organized as follows. First, the background and related work is presented. This is followed by an exploration of a list of characteristics that uniquely posit the nature of the developmental environment of Web Applications along with the implications of standards wherever applicable. Next, the implications of these characteristics in academia and industry are considered, and challenges and directions for future research are outlined. Finally, concluding remarks are given.

Contact author: Pankaj Kamthan, email: kamthan@cse.concordia.ca

Background and Related Work

This section includes preliminaries that are relevant for the rest of the paper. It also discusses previous work on characterizing Web Applications.

Basics of Web Engineering

The Web came into prominence in the early 1990s and, as indicated by a number of indices (http://www.worldwidewebsize.com/), continues to grow. However, the need for its management in general and an engineering approach towards the development of Web Applications in particular was only realized in the late 1990s (Powell, Jones, & Cutts, 1998). It is then that the terms such as *Web Site Engineering* and *Web Engineering* were coined. Web Engineering is a discipline concerned with the establishment and use of sound scientific, engineering, and management principles, and disciplined and systematic approaches, to the successful development, deployment, and maintenance of 'high-quality' Web Applications.

Table 1 illustrates a number of other disciplines from computer engineering, science, and technology that Web Engineering derives from and depends upon for its existence. A detailed discussion of these cognate disciplines is beyond the scope of this paper. Each of these disciplines contributes to one of more of the basic elements of Web Engineering, namely project, people, process, and product. There are certain disciplines such as quality engineering that can be further divided into sub-disciplines like reliability engineering, security engineering, usability engineering, and so on.

Table 1. A list of cognate disciplines of Web Engineering.

Web Engineering	Is-Related-To	Document Engineering
		Distributed Systems Engineering
		Human-Computer Interaction
		Hypermedia Engineering
		Information Systems Engineering
		Model Engineering
		Multimedia Engineering
		Network Engineering
		Quality Engineering
		Software Engineering

For the sake of this paper, a *Web Site* is defined as a collection of resources that reside in a distributed computing environment enabled by the technological infrastructure of the Internet and a *Web Application* is defined as a Web Site that behaves like an information-intensive interactive software system specific to a domain and typically requires a non-trivial infrastructure for development. The aforementioned infrastructure may include a disciplined and systematic development process, a team with high-level of knowledge and skills, deployment of additional software on the client- and/or server-side, and a schedule comprising of several weeks or months from inception to completion. A Web Application is an exemplar of distributed software systems.

Stakeholders of Web Applications

A *stakeholder* is a person who has interest in a Web Application for some purpose. The stakeholders for a software system in general and Web Application in particular can be classified on the basis of different views of the business, social, and technical environment in which the system resides. Fig. 1 shows a partial classification of stakeholders of a Web Application organized in a hierarchical manner.

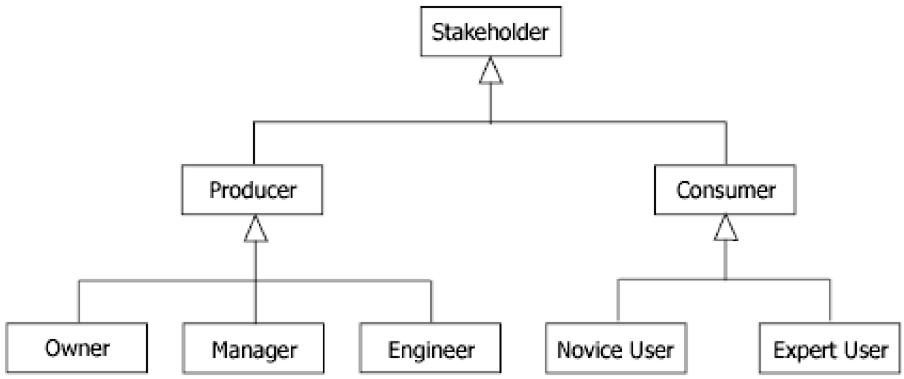

Fig. 1. A partial view of a taxonomy of stakeholders of a Web Application.

In this paper, a business view is considered, and the stakeholders of a Web Application are broadly classified into *producers* and *consumers*. The producers are responsible for server-side concerns of a Web Application; the consumers are receivers on the client-side of a Web Application. For example, an engineer is a kind of producer of a Web Application, and a user is a kind of consumer of a Web Application.

The classification can be granularized further. For example, a user could be decomposed into novice and expert user, or into beginner and advanced user.

It is possible to devise more sophisticated stakeholder classification schemes (Pacheco & Garcia, 2008) based on other criteria. For example, stakeholders could be classified as based on the degree of influence on the development of a Web Application or the interaction with a Web Application. It is also possible to devise a multifaceted classification scheme for the stakeholders of a Web Application. However, doing so is beyond the scope of this paper.

Related Work on Characterizing the Development of Web Applications

This section presents a chronological account of previous work related to this paper. A model for characterization of Web Applications has been presented (Lowe, 2002). However, details of individual characteristics are not given and the discussion is relatively dated.

It has been noted that Web Applications vary in a number of ways from traditional software including uncertainty of the domain, often shorter time-to-market, and rapid changes in technologies (Lowe, 2003; Ziemer & Stålhane, 2004). This is relevant to the requirements engineering of Web Applications. However, the arguments are often based on perception rather than social and technical reality.

It has been pointed out that different types of Web Applications vary along the lines of their nature, form, purpose, and development (Selmi, Kraïem, & Ghézala, 2005). However, the treatment is relatively dated.

The variations between software engineering and Web Engineering have been pointed out (Mendes & Mosley, 2006). However, the criteria focus on the underlying technologies rather than the stakeholders.

Finally, a characterization from the viewpoint of basic elements of Web Engineering, namely project, people, process, and product has been proposed (Kamthan, 2009a). It highlights the characteristics and challenges that make a Web Application unique as compared to desktop software systems. The approach taken in this paper is an alternate, although not necessarily orthogonal, to this work.

Towards Identifying and Understanding the Evolution of Web Applications Engineering

The following characteristics, based on the observations and experience of the authors over a number years, identify the pivotal directions of evolution in the developmental environment of Web Applications: [C-1] Computing Environment-Neutral, [C-2] Domain-Specific, [C-3] Human-Centered, [C-4] Information Interaction-Intensive, [C-5] Model-Driven, [C-6] Open Environment-Based, [C-7] Pattern-Oriented, and [C-8] Quality-Sensitive. [C-1] – [C-8] are not orthogonal.

Fig. 2 depicts [C-1] – [C-8], and their interrelationships of dependencies. The presence of an arrow signifies a binary relationship in the sense that the source 'relies-upon' the destination.

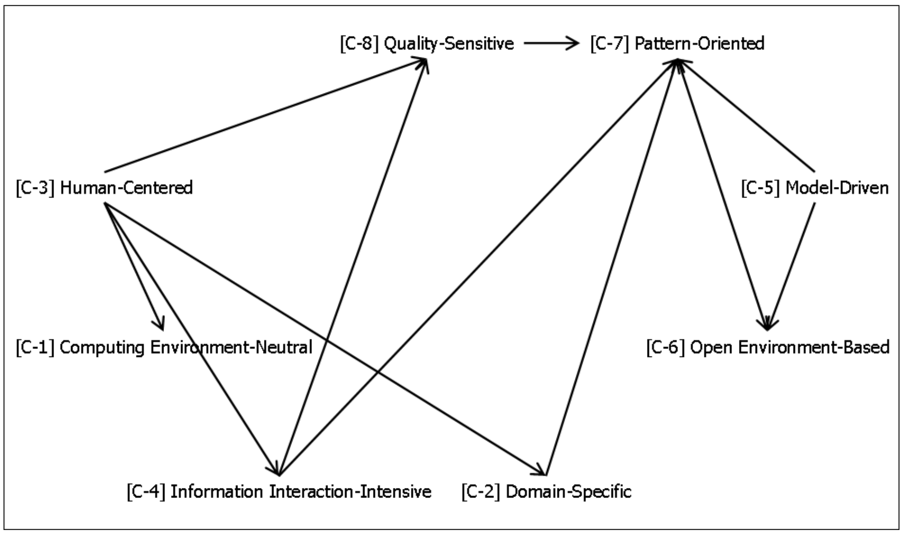

Fig. 2. The characteristics of evolution in the development of Web Applications and their interrelationships of dependencies.

Characteristics in Context

The characteristics [C-1] – [C-8] apply only to the Surface Web, not the Deep Web. It is the contention of the authors that [C-1] – [C-8] are relevant; however, there is no claim that they are sufficient. The degree of salience of [C-1] – [C-8] is not equal; however, this paper also does not compare (say, rank) these characteristics. The relationships between [C-1] – [C-8] are many-to-many and non-transitive. In the rest of the paper, the relationships between these characteristics have been highlighted explicitly wherever necessary.

It could be noted that certain additional characteristics, such as Ethically-Aware, Standards-Conforming, or Value-Added, are among the potential candidates that are desirable. However, currently these can not be unequivocally seen as perceived directions for the evolution in the development of Web Applications.

[C-1] Computing Environment-Neutral

In the last two decades, there has been a proliferation of computing devices and user agents with diverse configurations. In particular, there has been an influx of computing devices that vary considerably in their options for data input/output, screen capabilities, memory, disk space, and processing power.

It is not practical for a typical user to change the underlying device or user agent to suit the demands of different Web Applications. It is expected by the users that the same services are available on an array of devices (related to [C-3]). Therefore, it has become ever more important for Web Applications to be perceived as being neutral to the volatility in the computing environment.

In the past decade, there has been notable progress in this direction. The current initiatives such as the W3C's Ubiquitous Web Applications Activity are a step forward. The producers of Web Applications have been confronting the challenge by providing dynamic delivery of resources based on an assessment of the client-side computing environment. This assessment can, for example, include recognizing the device and personalizing according to user preferences.

Indeed, the dynamicity of the underlying technology is being increasingly suppressed in favor of the services being rendered. For example, pointing out to the user to adjust screen resolution, suggesting that a specific user agent is the 'best fit' for a given Web Application, or recommending to download a specific plug-in for a particular user agent to view a video clip or read a document, has become far less common than it used to be a decade ago. Indeed, doing so may even be considered a design anti-pattern.

Implications from Standards

The proliferation of information technologies such as the Extensible Markup Language (XML), the Extensible Stylesheet Language Transformations (XSLT), and the Cascading Style Sheets (CSS) has been useful in bringing the neutrality of computing environment to a practical realization.

Challenges

In general, a commitment to computing environment-neutrality is not free of cost. In order to achieve computing environment-neutrality to an acceptable degree, any advantages specific to, for example, hardware or software optimization, must be sacrificed. To check device variability, particularly that on mobile devices, the devices themselves and/or device simulators need to be acquired. The aim of minimizing (ideally, eliminating) dependency of a Web Application on user agents can require more time and effort on part of producers, not least due to the fact that the user agents are themselves prone to change. This variability also places extra burden on acceptance testing.

[C-2] Domain-Specific

From initial 'experiments' in new technologies, Web Applications have evolved to become an integral and, in some cases, uniquely indispensable, part of an organization's vision. In this sense, the evolution of Web Applications has been from genericity to specificity.

There are different types of software systems addressing different domains (Forward & Lethbridge, 2008), and the same is the case for Web Applications (Arrue, Vigo, & Abascal, 2008). The nature and underlying goals of an organization can constitute information of the underlying domain and this information can be reflected in a Web Application. For example, a Web Portal is domain-specific, and has been instrumental in the proliferation of domain-specific Web Applications. Indeed, being domain-specific has been central to the success of Web Applications aimed for e-business, e-education, and e-government.

There is a need to distinguish and classify different types of Web Applications addressing different domains. This is because an understanding of the domain and its subsequent realization has an impact on all aspects of the development of a Web Application. The actual information, style of expressing information, and presentation of information in a Web Application are all influenced by the properties of the domain. For example, the theme of a Web Portal selling gaming software needs to be identifiably different from a Web Application connected to an airline's information system or to a University's learning management system.

There is increasing support for domain engineering in the development of Web Applications. The Feature-Oriented

Domain Analysis (FODA) methodology is one of the approaches for domain analysis. The information of the underlying domain is critical to the user models and usage models (related to [C-3]), and the requirements specification. There are high-level design patterns such as the SITE BRANDING pattern (Van Duyne, Landay, & Hong, 2003) that suggest that the genre of a Web Application be specified at the start of development (related to [C-7]). These high-level decisions constrain the selection of low-level design patterns and, as a result, the design of the final Web Application is specific to the target domain.

Implications from Standards

It has been pointed out in the ISO 9241-151:2008 Standard that a Web Application should explicitly indicate its purpose.

Challenges

There can be undesirable side-effects of domain-specificity. The language used in a domain-specific Web Application, including terminology, may be acceptable to regular users; however, it may alienate new users if appropriate steps are not taken. These steps could include general introduction to the features of the Web Application and provision of context-sensitive help.

[C-3] Human-Centered
The need to explicitly take into account the concerns of users always had a part in the development of highly interactive systems of the past few decades. For example, it led to the ascent of *participatory design* in user interface engineering in the 1980s and the proliferation of *agile methodologies* in software engineering in the 1990s.

In the past decade, it has been recognized even more so that consideration of only technical aspects is insufficient in providing the quality of service expected from Web Applications by their consumers. The users of interactive systems such as Web Applications are not homogeneous. Indeed, users can vary in a number of ways including age, mental and physical ability, educational background and skills, culture, gender, geographical location, goals, personal preferences, and temperament. This diversity must be acknowledged, embraced, and subsequently acted upon. To increase the likelihood of acceptance and hope for success, the human factors need to be explicitly acknowledged and embraced. Indeed, the human-centeredness of Web Applications needs to be reflected *throughout* development and during operation.

The acknowledgement of the significance of the role of humans has led to a variety of changes at all levels of development of Web Applications. It also led to a number of approaches, as shown in Table 2, for developing interactive systems in general and Web Applications in particular.

Table 2. The different human-centered approaches for the development of Web Applications.

Human-Centered Approaches in Web Engineering	Activity-Centered Design
	Contextual Design
	Empathic Design
	Goal-Directed Design
	Participatory Design
	Sociable Design
	Task-Centered Design
	Usage-Centered Design
	User-Centered Design
	User Experience Design

The shift of the focus on technologies to focus on people has had an impact on how the development of Web Applications should be viewed. The process models for development of Web Applications have become more human-centric (Kamthan, 2008). For example, approaches for formulating concrete user models based on an input of aforementioned human factors and for eliciting user requirements that are based on a dialogue with real users have garnered attention.

The acceptance of a user's context, inception of personalization (related to [C-2]), and use of recommender systems are some of the prime examples of human-centeredness of Web Applications.

The New Generation of Consumers of Web Applications

The interplay between people and digital technology has reached a new frontier. A *digital native* (Palfrey & Gasser, 2008) is a person who was born at the time digital technologies were taking shape and/or has grown up with digital technologies. These digital technologies include those that underlie the current non-stationary computing devices as well as the Internet in general and the Web in particular. In contrast, there is also a growing population of digital laggards, especially in certain countries where the human life expectancy has improved.

The need for taking heterogeneity of consumers into account during the development of Web Applications has become ever more important due to the increasing number of and the widening gap between digital natives and digital laggards.

Implications from Standards

This movement towards human-centeredness had an impact on the development of international standards. For example, there is treatment of software ergonomics in the ISO/TR 9241-100:2010 Standard, for user needs and context of use in the ISO 9241-210:2010 Standard, for user goals and user tasks in the ISO 9241-151:2008 Standard, and for user-centered design in the ISO 9241-210:2010 Standard.

Challenges

There are possible side-effects of human-centeredness of Web Applications. An emphasis on the users and their environment may lead to over-attention and dependence on the data derived from ethnographic studies and user feedback, both during and after development. An initiative for personalization usually comes at the cost of privacy, and providing a priori guarantees to users of the appropriate use of the data submitted by them can be a challenge. There is also the potential for the use of the Web for *persuasion* (Weinschenk, 2009) that can adversely impact the credibility of a Web Application (Fogg, 2003).

[C-4] Information Interaction-Intensive

The 'invisibility' of a computer was posited more than two decades ago (Norman, 1998). To be able to shield the intricacies of a computer from humans has become an intended goal of software systems in the intervening years. Indeed, for some developments involving frequent, sophisticated interactions between humans and computers, such as 'smart' environments (Streitz, Kameas, & Mavrommati, 2007), it has become a reality.

By treating information as a 'first-class' concern, the Web has acted as a catalyst in this regard. The Web has placed yet another layer between a human and the computer's operating system interface, namely that of *information interfaces* (Pirolli, 2007). It has been shown in empirical studies (Weinreich et al., 2008) that, instead of merely seeking information, users now also expect to be able to interact with a Web Application to carry out certain tasks. Therefore, the study of human-information interaction (HII) in general and information interfaces in particular is imperative for understanding the nature of Web Applications. It is likely that, among the consumers of Web Applications, the digital natives are the most exposed to and accustomed to the presence of information interfaces (related to [C-3]).

Fig. 3 illustrates a tier of multiple interfaces: the information interface of an abstract Web Application, the user interface of the user agent, and the user interface of the underlying operating system.

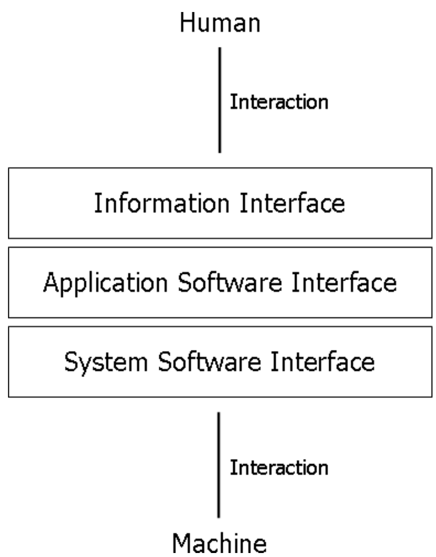

Fig. 3. The information interface in the hierarchy of human-to-machine interaction.

The presence of hypermedia in general and hypertext in particular in a document; controls in an embedded applet; menus of conventionally desktop, office publishing applications inside a user agent; and so on, are essentially manifestations of information interfaces. Fig. 4 shows an abstract information interface in relation to other, higher-level interfaces, namely that of the user agent and the operating system. The presence of information design patterns in the design of the elements of the information interface is evident.

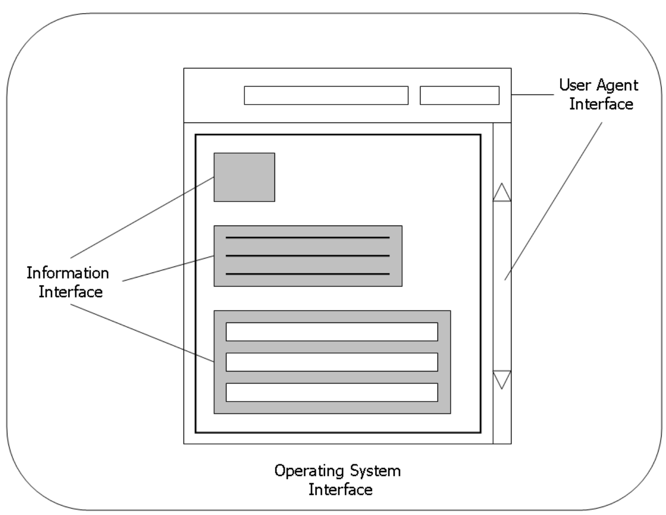

Fig. 4. An example of an abstract information interface for a Web Application and its context.

HII could be considered as a sub-field of human-Web interaction (HWI). HII relies on a number of cognate disciplines (Albers, 2008) such as cognitive psychology, social psychology, human factors, HCI, information science, and technical communication. The structural, behavioral, and creational aspects of information interface design usually make use of patterns (related to [C-7]).

The ascent of HII has had broad and profound implications, a trend that is likely to continue in the foreseeable future. HII has led to the emergence in recent years of new sub-disciplines such as the ascent of information modeling, information design, and information quality. It has also helped reduce the (at times, superficial) compartmentalization of art, science, and engineering, and brought people from these areas together to collaborate. The usefulness of conventional conceptual modeling approaches and relevance of conventional (software) quality models is being challenged.

Implications from Standards

There conventionally has not been any standard for user interfaces on the Web. The existence of the ISO 9241-151:2008 Standard has changed that; however, not being legally binding, there is no guarantee that it will be adopted and followed in the future.

Challenges

The presence of information interfaces presents new development challenges. For example, a consumer may have to deal with multiple different interfaces within the same Web Application or across different Web Applications. This could be prohibitive, particularly in the presence of a weak *information scent* (Pirolli, 2007) and/or the absence of any context-sensitive help. The challenge increases if there are features across interfaces that are same or similar in presentation but different in functionality, or same or similar in functionality but different in presentation. For example, accessibility and usability issues of the search interface provided by a Web Application are not necessarily identical to that of a general-purpose search engine or to that provided by a user agent being used to access that Web Application.

[C-5] Model-Driven

The development of Web Applications has steadily been moving towards *abstraction*. There are various approaches for achieving abstraction, one of which is conceptual modeling. There are a number of advantages of modeling including creating an environment for discussion across project team, early cost estimation, exploring and experimenting with design alternatives, identifying stakeholders, minimizing attention at the level of transient technologies, and so on.

The assortment of desirable models includes a problem domain model, user model (such as user role, user profile, and/or persona), usage model (such as a use case model and/or a task model), and macro- and micro-architectural design models. The design models can be refined further to include specific aspects of design such as structure and behavior, usually using design patterns (related to [C-7]). Table 3 lists some early conceptual models in the development of Web Applications and their potential uses.

Table 3. The potential uses of early conceptual models in the development of Web Applications.

Conceptual Model Type	Potential Use(s)
Problem Domain Model	Requirements Elicitation and Specification, Glossary
Stakeholder Model	Project Management Plan
User Model	User Requirements Elicitation and Specification, Acceptance Testing
Use Case Model	Cost Estimation, Behavioral Requirements Elicitation, Test Case Specification
Task Model	Interaction Design Description

The interest and initiatives towards modeling Web Applications has gradually increased over the past decade (Rossi et al., 2008). There are modeling approaches such as the Object-Oriented Hypermedia Design Method (OOHDM), UML-based Web Engineering (UWE), W2000, and the Web Modeling Language (WebML) that are specific to Web Applications. The availability of open modeling environments (related to [C-6]) provide choices and such as ArgoUML can help Web Application projects with budgetary constraints.

Implications from Standards

There is explicit support for conceptual modeling of information and information structures in the ISO 9241-151:2008 Standard. In recent years, various general approaches for meta-modeling have been proposed including the ISO/IEC 24744:2007 Standard and, under the umbrella of the Model-Driven Architecture (MDA) in general and the Meta-Object Facility (MOF) in particular, the Unified Modeling Language (UML). There are now a number of extensions of UML for Web Applications in form of UML profiles, each often focused on one of the possible views of the application.

Challenges

In spite of its usefulness, the adoption of conceptual modeling is not automatic. The organization's process maturity and availability of resources can affect the degree of commitment to conceptual modeling. In general, modeling is both art and science (Lieberman, 2007), and the knowledge and experience in these is rare. The presence of different modeling notations may give engineers a choice; however, lack of communicability across notations persists. There are a number of UML profiles for Web Applications (Rossi et al., 2008), often with different goals; however, none have emerged as a 'standard.' Finally, for automatic model transformations, such as in the case of MDA, adequate mapping and necessary tool support may not exist.

[C-6] Open Environment-Based

The availability of open resources, for both the producer and the consumer, has played a critical role in the success of the Web. Indeed, it is not an overstatement to assert that a widespread acceptance of the Web in absence of an open environment would not have been possible.

The architecture of the Web (Jacobs & Walsh, 2004) is based on an open environment. The specifications for Hypertext Transport Protocol (HTTP), the Internet protocol suite on which it is based, and the successors of HTTP, including but not limited to, the Hypertext Transport Protocol Secure (HTTPS) and SOAP, are open. The same applies to addressing schemes such as the Uniform Resource Identifiers (URIs) and Internationalized Resource Identifiers (IRIs). Finally, specifications of markup languages for information description such as the HyperText

Markup Language (HTML) and DocBook, scripting languages such as ECMAScript and PHP: Hypertext Preprocessor (PHP), relational database management systems (RDBMS) such as MySQL, and content management systems (CMS) that are based on PHP/MySQL such as WordPress, are also open.

The support software that is of primary concern to the producers and the consumers of Web Applications has also been open. The classical exemplars on the client-side include the NCSA Mosaic and Lynx and on the server-side include the NCSA Web server. These were followed by the introduction of the Apache Web server and then the formation of the Apache Software Foundation, followed by the ascent of the Mozilla Project. The Open Source Software (OSS) has made an indispensable contribution to the evolution of the Web and its successors, namely the Semantic Web and the Social Web. The 'family' of OSS clients, servers, scripting and programming language processors, and content management systems has flourished over the past decade. This has eased the entry barriers for small-to-medium-size enterprises (SME) to use and/or produce Web Applications.

OSS for the Web and the Web for OSS

The relationship between the Web and OSS is symbiotic (Kamthan, 2007). Indeed, Web has become the de facto platform for the development of OSS. There are a number of OSS foundries such as SourceForge (http://sourceforge.net/) that have been set up as Web Applications, and have been instrumental in dissemination of OSS; there are OSS tools for high-fidelity prototyping, which have proven useful for user interface design; there are OSS tools for test automation, which have contributed to quality evaluation; and so on.

Implications from Standards

The technological infrastructure underlying the Web including HTTP, HTTPS, SOAP, URI, HTML, ECMAScript, XML are standards or are considered as standards.

Challenges

In spite of the prevalence of OSS, their quality is still an open issue. The stability of OSS is not guaranteed: there are open source projects that have become dormant or have been discontinued after relatively short period of time. A commitment to such OSS can therefore impact the velocity of development of a Web Application. As open source projects are voluntary efforts, timely customer support can be an issue, especially for new initiatives.

[C-7] Pattern-Oriented

The reliance on the knowledge garnered from past experience is crucial for any development. A *pattern* is defined as an empirically proven solution to a recurring problem that occurs in a particular context (Buschmann, Henney, & Schmidt, 2007). If made available properly, patterns constitute one form of conceptually reusable experiential knowledge for the commons.

In the past couple of decades or so, patterns have been discovered and applied in a variety of domains, including those related to Web Applications. The discipline of Web Engineering was initially influenced by the work in hypermedia engineering, interaction design, and information systems engineering, followed by software engineering. The same trend seems to have been followed in the initial 'discovery' of patterns for Web Applications.

In recent years, a systematic approach based on patterns, for orienting the development of (Mobile) Web Applications aiming for 'high-quality', has been realized (Kamthan, 2008). In particular, it is expected that conceptual models of development during the analysis and design phases are based on patterns. The deployment of tools can become crucial in a 'large-scale' use of patterns (related to [C-6]). Table 4 illustrates the role of patterns in the development of Web Applications based on some Web Engineering process.

Table 4. The relationships between patterns and models in a process for the development of Web Applications.

Web Engineering Process	Is-Sensitive-To	Web Application Quality Model
	Is-Visible-In	Conceptual Models
Web Application Quality Model	Is-Supported-By	Patterns
	Depends-On	Web Application Stakeholder Model
Conceptual Models	May-Use	Patterns
	Lead-To	Web Application

Patterns for Web Applications and Web Applications for Patterns

The relationship between the Web and patterns is symbiotic (Kamthan, 2008). Indeed, Web has become the de facto platform for dissemination of patterns. There are a number of pattern collections on repositories such as the Hypermedia Design Patterns Repository, the Portland Pattern Repository, and the Amsterdam Collection of Patterns that have been set up as Web Applications, and continue to serve the pattern community.

Implications from Standards

There are currently no standards related to any inherent aspects of patterns. The Software Engineering Body of Knowledge (SWEBOK), as defined by the ISO/IEC TR 19759:2005 Standard, has support for patterns in its Software Design Knowledge Area. The ISO/IEC 23026:2006 Standard and the ISO 9241-151:2008 Standard also have limited support for patterns.

Challenges

In spite of the advantages emanating from a commitment to patterns, there are certain limitations. There may be insufficient development experience in new domains. Therefore, even though desirable, there may not be any patterns for such domains. For example, even though the awareness of accessibility in the development of Web Applications has increased over the past decade, there is currently a scarcity of stable and mature accessibility patterns. The acquisition of patterns can pose a challenge, particularly by those who are new or unfamiliar with the pattern collections available in print and/or electronic form. This is largely due to the absence of a single index of patterns and lack of documented experience in using patterns. The selection of patterns is also non-trivial to compare patterns that are similar, that is, provide different solutions to the same problem due to a number of reasons including the absence of a 'standard' for pattern description. It is expected that the use of patterns can aid quality; however, in general, the relationship between quality and patterns (Kamthan, 2009b) is equivocal (related to [C-8]).

[C-8] Quality-Sensitive
The pursuit for product quality is intrinsic to all engineering, and Web Engineering is no exception. The success of a software system as perceived by its stakeholders is often strongly related to its quality (Pertet & Narasimhan, 2005). Indeed, systems exhibiting 'poor' quality can be rejected by their consumers and, in turn, can incur notable losses to their producers.

A quality model is useful for creating an understanding of quality. In general, a quality model provides a decomposition of quality into a number of relevant attributes and relationships between those attributes. The quality models proposed by standards initiatives for desktop systems are deemed insufficient for the needs presented by distributed software systems and their stakeholders. For example, the ISO/IEC

9126-1:2001 Standard does not adequately address information quality. In a recent quality model for Web Applications (Kamthan, 2008), there is emphasis on a number of quality attributes including accessibility, credibility, legality, maintainability, privacy, reliability, security, and usability. This model also underscores the interplay of quality and patterns (related to [C-7]).

The movement over the years in the quality engineering of Web Applications has been from technical to social concerns. The needs of the consumers should be viewed as 'first-class' by producers (related to [C-3]). It is therefore only natural that the development must take into account the *experience*, including emotional responses (Norman, 2004), of users during interaction with a Web Application. The motivation for achieving 'high-quality' Web Application has evolved naturally. The users are spending increasingly more time on the Web, and increasingly depending on the Web for routine activities. For example, a corporate trainer may want to use a Web Application for laboratory demonstration without being interrupted by unsolicited pop-ups on unrelated topics; a senior citizen with low visual acuity would like to use the Web for banking from home but may be concerned about readability of text and entrusting others with personal information; a person with epilepsy would like to use the Web to look for travel destinations for her upcoming vacation without being confronted with animations; and so on. For that, accessibility, credibility, and usability are likely to be seen as increasingly significant measures of success of a typical Web Application.

There is concerted effort by commercial vendors like Adobe Systems, IBM, Microsoft, and Sun Microsystems to ensure that their products are sensitive to quality concerns of end-users. In particular, the support for accessibility in products such as Adobe Flash, Adobe Reader, and Java programming language has increased over the past decade.

Prevention and Cure

In recent years, the significance of *preventative* approach to quality assurance and evaluation throughout the development process of Web Applications has been acknowledged (Kamthan, 2008). This is necessary in conjunction with curative approaches to quality assurance and evaluation such as inspections and testing.

Implications from Standards

There are standards pertaining to both generic as well as specific aspects of quality. The ISO 9000:2005 Standard and the ISO/IEC 9126-1:2001 Standard apply to a general software system. There is explicit emphasis on accessibility, privacy, and usability in the ISO/IEC 23026:2006 Standard and/or the ISO 9241-151:2008 Standard. Furthermore, there is an implicit emphasis on credibility in these standards. There have been notable efforts in this area but more needs to be done.

Challenges

In spite of the evident significance of quality assurance and evaluation, it is not free of cost. The size of the organization, process maturity, requisite training, timelines-to-market, and availability of resources are some of the factors that can affect a commitment to quality control. For example, the infrastructure required for setting up a dedicated quality engineering program, including laboratories, may not be within the budget of certain academic institutions and SME. The number of engineers for testing Web Applications has improved over the years; however, experts in heuristics evaluation of accessibility or usability are scarce and not inexpensive. There are a few free-to-use tools for evaluating, say accessibility, credibility, and usability of Web Applications; however, guidance for their appropriate use may not always be freely available, if at all. There currently are limited approaches for quality meta-modeling. It appears that currently there is no single, universal quality model that is applicable to all software systems. Furthermore, in their existing form, the quality models for conventional software systems are not sufficient for Web Applications. A Web Application can be a complex system, and a single quality model may not be sufficient. Indeed, as shown in Table 5, there is a need for multiple quality models for a Web Application, each emphasizing a specific aspect of that Web Application.

Table 5. A selected collection of different quality models in the development of Web Applications.

Type of Quality Model	Examples
Conceptual Model Quality Model	Domain Model Quality Model, Use Case Model Quality Model
Specification Quality Model	Requirements Specification Quality Model, User Story Quality Model
Information Quality Model	Text Quality Model, Graphics Quality Model, Animation Quality Model
Design Quality Model	Navigation Design Quality Model, Presentation Design Quality Model
Implementation Quality Model	Data Quality Model, Source Code Quality Model

Implications of Characterizing Web Applications Engineering

The characteristics that reflect relatively stabilizing invariants in the evolution of Web Applications need to be identified and explored for a number of reasons, including their implications on the two important sectors of society: academia and industry. These are explored next.

Implications for Academia

A characterization of Web Applications aims to provide an understanding of the state-of-the-art in the development of Web Applications. This is relevant to both educators and researchers.

- **Teaching.** A characterization of Web Applications could be useful for educators who are involved in curriculum development related to Web Engineering at their respective institutions of higher education. For example, [C-1] – [C-8] could be placed into the Web Engineering Body of Knowledge (WEBOK) (Navarro, 2009) and aligned according to a pedagogical model for Web Engineering education (Hadjerrouit, 2005). Furthermore, instead of the conventional focus on technology-oriented courses under the label of 'Developing a Web Site,' 'Using Technology X or Tool Y for Creating a Web Site,' 'Web Design,' or the like, it could open new vistas for curriculum design by increasing the level of abstraction of courses. The potential directions for courses include Conceptual Modeling of Web Applications (related to [C-5]), Open Source Web Engineering (related to [C-6]), Quality of Web Applications (related to [C-7] and [C-8]), and User-Centered Web Engineering (related to [C-3] and [C-8]), to name a few. In these courses, basic concepts of Web Engineering and services to humans become primary, and technology, albeit important, becomes secondary.

- **Research.** A characterization of Web Applications could be useful for researchers who are looking for new problems related to Web Engineering. For example, an examination of the client-side properties of Web Applications has led to the formulation of a more accurate usability model (Bruno, Tam, & Thom, 2005). There are obviously many other problems of potential research interest including, but not limited to, elicitation of relationships between [C-1] – [C-8] and the standards for Web Engineering such as the ISO/IEC 23026:2006 Standard; elicitation of metaphors for Web Applications on small, non-stationary devices (related to [C-3]); being able to provide rich services to both able and disable users in a cost-effective manner (related to [C-3] and [C-8]); formulation of metrics for accessibility (related to [C-8]); and so on.

Implications for Industry

The ascent of the Web has led to fundamental shifts in the industrial culture. The Web has become a 'first-class' medium for a number of industries. This has had a notable impact on business decision making and organizational practices, especially those industries that are committed to Business-to-Business (B2B) and Business-to-Consumer (B2C) electronic commerce (E-Commerce).

A characterization of Web Applications is relevant to technical leaders who are involved in planning their future information technology policies at their respective organizations. There is a constant drive in industry to gain a competitive edge through a number of means such as by improving existing workflows, finding avenues of reusable knowledge, eliciting new directions for personnel training, and so on. An understanding of the avenues being pursued in the engineering of Web Applications could help charter new paths for these directions.

- **Process.** In the early days of the Web, the organizations followed an ad-hoc or essentially linear approaches towards the development of 'large-scale' Web Applications. With the passage of time, there have been fundamental changes in the process environment that is adopted and deployed for the development of Web Applications. For example, taking into consideration the working habits of the people involved (related to [C-3]) and following an evolutionary approach during development are being recognized as increasingly relevant to the underlying process environment, a trend that is likely to continue for the foreseeable future. In that regard, agile methodologies have a role to play. The organizations have come to embrace methodologies such as Crystal Orange Web, Extreme Programming (XP), Lean Development, Scrum, and customizations of the Unified Process (UP) including the Rational Unified Process (RUP) and the Open Unified Process (OpenUP). In some cases, process environments being followed are adapted for the industry in general and to the organization in particular. In other cases, users are being solicited, preferably early, for evaluation of and feedback on design prototypes. If carried out appropriately, these practices have positive implications towards the quality and longevity of the product (related to [C-3] and [C-8]).

- **People.** There are a number of stakeholders of any engineering product. The ubiquity of the Web has taken the classical industrial practice of attending to customers towards new directions. There is increasing attention towards people involved in the use of Web Applications. In particular, the focus continues to shift from technological determinism to the needs and tasks of users. For example, user models such as personas (Pruitt & Adlin, 2006), usage models such as use case models (Neill & Laplante, 2003), and approaches for user requirements such as user stories (Cohn, 2004) are being increasingly deployed in industry (related to [C-3]).

Directions for Future Research

The Web upon conception, and for a number of years thereafter, has remained centered on the one-to-many model of communication of information. It has been realized that the Web must evolve if it is to reach its full potential envisioned at its inception.

Towards Characterizing the Development of Next Generation of Web Applications

There have been two notable and complementary directions of the evolution of the Web. The *Semantic Web* has emerged as an extension of the Web that adds technological infrastructure for better knowledge representation, interpretation, and reasoning (Hendler, Lassila, & Berners-Lee, 2001). The *Social Web*, or as it is more commonly referred to by the pseudonym Web 2.0 (O'Reilly, 2005), is the perceived evolution of the Web in a direction that is driven by 'collective intelligence,' realized by information technology, and characterized by user participation, openness, and network effects.

The Semantic Web and the Social Web, as shown in Fig. 5, reflect machine-oriented and human-oriented extension of the Web, respectively. The development of Semantic Web Applications is concerned with representation of information, while the Social Web Applications is concerned with the presentation of information, respectively.

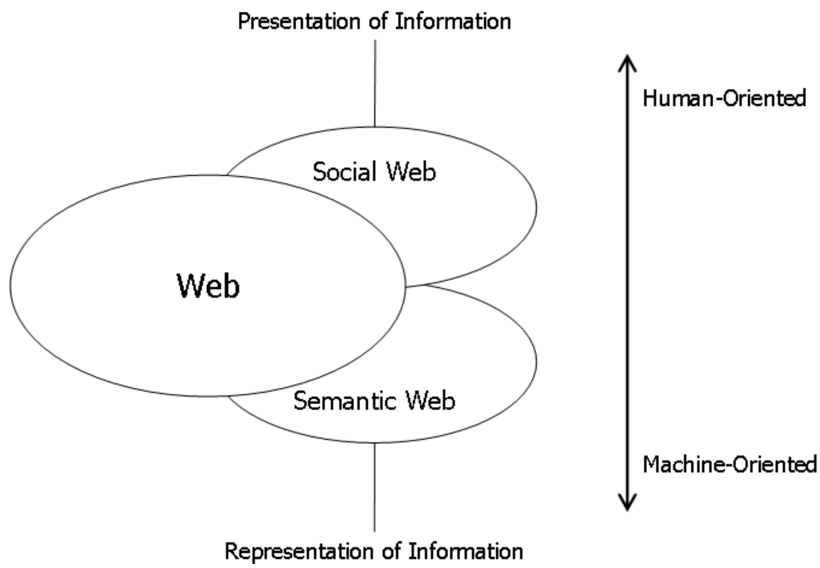

Fig. 5. The two salient directions of the extensions of the Web.

The attention on the Semantic Web has reinforced the significance of the separation of the representation from the presentation of information. This, in turn, has positive implications towards accessibility, usability, and maintainability. For example, multiple user-supplied style sheets can be associated with a single source of representation, and multiple target presentations can be generated from a single source of representation. A better representation of knowledge opens new possibilities for 'intelligent' user interfaces (IUIs), especially those for adaptation and personalization. The representation of knowledge in form of ontologies hold the promise of more precise and relevant searching as demonstrated by experimental search engines like CORESE, SWSE, and Swoogle.

The Social Web celebrates the human involvement in the evolution of the Web. If the Web leveled the playing field between large and small businesses, the Social Web, to certain extent, levels the playing field between producers and consumers. It has spawned a new era of Social Web Applications, and opened new vistas for collaboration among globally distributed participants. Indeed, Social Web applications like Del.icio.us, Facebook, Flickr, Scribd, Wikipedia, and YouTube, are but a few examples of the phenomenon where a consumer becomes a co-producer, or a *prosumer*, in a social network.

It is relatively early to characterize this new generation of applications, and constitutes an area of future research interest. In doing so, the characteristics [C-1] – [C-8] are deemed necessary but not sufficient. For the sake of argument, consider the case of Social Web Applications (Bell, 2009). It is likely that Social Web Applications need to move from genericity to specificity (related to [C-2]). For example, there are generic social networking applications like Facebook, but also means for creating community-specific social networks using systems such as Ning. The need for an appropriate information interface design is crucial for Social Web Applications (related to [C-4]). For example, rich information interface design of Social Web Applications such as the DocuWiki editor need to be mindful of its higher-level context, namely that of the user agent and the operating system. Furthermore, to be inclusive, the Social Web Applications, especially those labeled as Rich Internet Applications (RIA) and implemented in Asynchronous JavaScript and XML (AJAX), need to commit to *universal design* (Chisholm & May, 2008), and therefore to accessibility and usability (related to [C-8]). However, given a group of participants in a Social Web Application, there is also a need to consider the sociological impact that entails from the computer-mediated communication among participants. Therefore, the types of relationships among stakeholders of Social Web Applications, such as those shown in Fig. 6, become crucial. In doing so, quality attributes that were previously dormant or nonexistent, especially those related to affect (Pang et al., 2010), come into play. For example, the quality attributes related to human emotion such as politeness (Whitworth, 2009) are imperative for both the success of social computing and sustainability of Social Web Applications but is not completely addressed by [C-8].

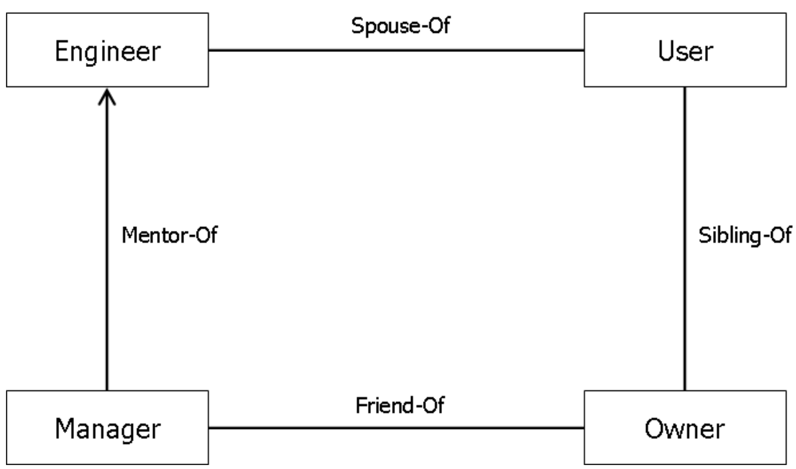

Fig. 6. A range of relationships between stakeholders of a Social Web Application.

Conclusion

The Web continues to grow at an alarming pace, and the dependence of the society on it continues to increase. The velocity (speed and direction) of evolution of the Web (Murugesan, 2010) and the sustainability of the Web are pressing imperatives towards which the society, as a whole, has a collective responsibility.

A commitment to the characteristics pertaining to the evolution of Web Applications presented in this paper comes with technical and social challenges for both the producers and for the consumers of these applications. In the past decade, there have been many advances towards enabling the technological infrastructure of the Web in lieu of addressing the technical challenges. However, there is much to be done in addressing the social challenges, especially as they pertain to the new generation of Web Applications. To do that, may require a change in the organization culture.

In conclusion, for a coherent evolution of the discipline of Web Engineering, its unique nature needs to be acknowledged and studied systematically. To address that, there is a need to consider Web Engineering from different, and hopefully time-invariant, viewpoints. The characterization presented in this paper is one direction in furthering the understanding of the nature of Web Engineering.

References

Albers, M. J. (2008). *Human-Information Interaction*. The Twenty Sixth Annual ACM International Conference on Design of Communication (SIGDOC 2008), Lisbon, Portugal, September 22-24, 2008.

Arrue, M., Vigo, M., & Abascal, J. (2008). *Supporting the Development of Accessible Web Applications*. Journal of Universal Computer Science, 14(16), 2699-2719.

Bell, G. (2009). *Building Social Web Applications: Establishing Community at the Heart of Your Site*. O'Reilly Media.

Bruno, V., Tam, A., & Thom, J. (2005). *Characteristics of Web Applications that Affect Usability: A Review*. The Nineteenth Conference of the Computer-Human Interaction Special Interest Group (CHISIG) of Australia on Computer-Human Interaction, Canberra, Australia, November 21-25, 2005.

Buschmann, F., Henney, K., & Schmidt, D. C. (2007). *Pattern-Oriented Software Architecture, Volume 5: On Patterns and Pattern Language*s. John Wiley and Sons.

Chisholm, W., & May, M. (2008). *Universal Design for Web Applications: Web Applications That Reach Everyone*. O'Reilly Media.

Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Addison-Wesley.

Fogg, B. J. (2003). *Persuasive Technology: Using Computers to Change What We Think and Do*. Morgan Kaufmann.

Forward, A., & Lethbridge, T. C. (2008). *A Taxonomy of Software Types*. School of Information Technology and Engineering. Technical Report: TR-2008-06. University of Ottawa, Ottawa, Canada.

Ginige, A., & Murugesan, S. (2001). *Web Engineering: An Introduction*. IEEE Multimedia, 8(1), 14-18.

Hadjerrouit, S. (2005). *Designing a Pedagogical Model for Web Engineering Education: An Evolutionary Perspective*. Journal of Information Technology Education, 4, 115-140.

Hendler, J., Lassila, O., & Berners-Lee, T. (2001). *The Semantic Web*. Scientific American, 284(5), 34-43.

Jacobs, I., & Walsh, N. (2004). *Architecture of the World Wide Web, Volume One*. World Wide Web Consortium (W3C) Recommendation.

Kamthan, P. (2007). *On the Prospects and Concerns of Integrating Open Source Software Environment in Software Engineering Education*. Journal of Information Technology Education, 6, 45-64.

Kamthan, P. (2008). *A Situational Methodology for Addressing the Pragmatic Quality of Web Applications by Integration of Patterns*. Journal of Web Engineering, 7(1), 70-92.

Kamthan, P. (2009a). *A Model for Characterizing Web Engineering*. In: Encyclopedia of Information Science and Technology, Second Edition. M. Khosrow-Pour (Ed.). IGI Global, 2631-2637.

Kamthan, P. (2009b). *On the Symbiosis between Quality and Patterns*. The Third International Workshop on Software Patterns and Quality (SPAQu 2009), Orlando, USA, October 25, 2009.

Kruchten, P. (2004). *Putting the "Engineering" into "Software Engineering"*. The Fifteenth Australian Software Engineering Conference (ASWEC 2004), Melbourne, Australia, April 13-16, 2004.

Kamthan, P., & Shahmir, N. (2010). *On the Directions of Evolution in the Engineering of Web Applications*. The Fourteenth IBIMA conference on Global Business Transformation through Innovation and Knowledge Management, Istanbul, Turkey, June 23-24, 2010.

Lieberman, B. A. (2007). *The Art of Software Modeling*. Auerbach Publications.

Lowe, D. (2002). *Characterisation of Web Projects*. The Eighth Australian World Wide Web Conference (AusWeb 2002), Sunshine Coast, Australia, July 6-10, 2002.

Lowe, D. (2003). *Web System Requirements: An Overview*. Requirements Engineering, 8(2), 102-113.

Mendes, E., & Mosley, N. (2006). *Web Engineering*. Springer-Verlag.

Murugesan, S. (2010). *Web X.0: A Road Map*. In: Handbook of Research on Web 2.0, 3.0 and X.0: Technologies, Business, and Social Applications. S. Murugesan (Ed.), IGI Global, 1-11.

Navarro, A. (2009). *A SWEBOK-based Viewpoint of the Web Engineering Discipline*. Journal of Universal Computer Science, 15(17), 3169-3200.

Neill, C. J., & Laplante, P. A. (2003). *Requirements Engineering: The State of the Practice*. IEEE Software, 20(6), 40-45.

Norman, D. A. (1998). *The Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution*. The MIT Press.

Norman, D. A. (2004). *Emotional Design: Why We Love (Or Hate) Everyday Things*. Basic Books.

O'Reilly, T. (2005). *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. O'Reilly Network, September 30, 2005.

Pacheco, C., & Garcia, I. (2008). *Stakeholder Identification Methods in Software Requirements: Empirical Findings Derived from a Systematic Review*. The Third International Conference on Software Engineering Advances (ICSEA 2008), Sliema, Malta, October 26-31, 2008.

Palfrey, J., & Gasser, U. (2008). *Born Digital: Understanding the First Generation of Digital Natives*. Basic Books.

Pang, M., Suh, W., Hong, J., Kim, J., & Lee, H. (2010). *A New Web Site Quality Assessment Model for the Web 2.0 Era*. In: Handbook of Research on Web 2.0, 3.0 and X.0: Technologies, Business, and Social Applications. S. Murugesan (Ed.), IGI Global, 387-410.

Pertet, S. M., & Narasimhan, P. (2005). *Causes of Failure in Web Applications*. PDL Technical Report PDL-CMU-05-109. Carnegie Mellon University, USA.

Pirolli, P. (2007). *Information Foraging Theory: Adaptive Interaction with Information*. Oxford University Press.

Powell, T. A., Jones, D. L., & Cutts, D. C. (1998). *Web Site Engineering*. Prentice-Hall.

Pruitt, J., & Adlin, T. (2006). *The Persona Lifecycle: Keeping People in Mind Throughout Product Design*. Morgan Kauffman.

Rossi, G., Pastor, O., Schwabe, D., & Olsina, L. (2008). *Web Engineering: Modelling and Implementing Web Applications*. Springer-Verlag.

Selmi, S. S., Kraïem, N., & Ghézala, H. H. B. (2005). *Toward a Comprehension*

View of Web Engineering. The Fifth International Conference on Web Engineering (ICWE 2005), Sydney, Australia, July 27-29, 2005.

Streitz, N., Kameas, A., & Mavrommati, I. (2007). *The Disappearing Computer: Interaction Design, System Infrastructures and Applications for Smart Environment*s. Springer-Verlag.

Van Duyne, D. K., Landay, J., & Hong, J. I. (2003). *The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience*. Addison-Wesley.

Weinreich, H., Obendorf, H., Eelco, M., & Herder, M. (2008). *Not Quite the Average: An Empirical Study of Web Use*. ACM Transactions on the Web, 2(1), 5:1-5:31.

Weinschenk, S. M. (2009). *Neuro Web Design: What Makes Them Click?* New Riders.

Whitworth, B. (2005). *Polite Computing*. Behaviour & Information Technology, 24(5), 353-363.

Whitworth, B. (2009). *Politeness as a Social Software Requirement*. International Journal of Virtual Communities and Social Networking, 1(2), 65-84.

Ziemer, S., & Stålhane, T. (2004). *The Use of Trade-offs in the Development of Web Applications*. International Workshop on Web Quality (WQ 2004). Munich, Germany. July 27, 2004.