Webology, Volume 12, Number 1, June, 2015

Characterizing relatedness of web and requirements engineering

Akshi Kumar
Dept. of Computer Science & Engineering, Delhi Technological University, Delhi, India.
E-mail: akshi.kumar@gmail.com

M.P.S. Bhatia
Division of Computer Engineering, Netaji Subhas Institute of Technology, Delhi, India.
E-mail: mps.bhatia@nsit.ac.in

Rohit Beniwal
Division of Computer Engineering, Netaji Subhas Institute of Technology, Delhi, India.
E-mail: rohitbeniwal@yahoo.co.in

Received October 15, 2014; Accepted June 15, 2015

Abstract

Web and Requirements Engineering have been well- recognized as two individual active areas of research in the past. Convergence between these two notable areas has been a point-of-discussion in recent years and offers new avenues of research. This paper explores this alliance from two perspectives; firstly, where Requirement Engineering can be viewed as a process for Web application development as it primarily concerns with adapting the Requirement Engineering process to the Web applications which are special in characteristics as compared to traditional software applications and secondly, where Web can be viewed as a supporting technology for improving the requirements engineering process and enabling new capabilities. The basics of this two-way relationship are examined along with state-of-art and potential opportunities to equip a better understanding.

Keywords

Web application; Requirements engineering; Web 2.0; Web 3.0

Introduction

Requirements Engineering is an intricate, multifaceted and time-consuming process. Its objective is to essentially assure the quality of resulting software. More recently, as our dependence and reliance on Web has increased; there has been a rising interest in the development of methodologies including processes, models and techniques to build web applications. An expected alliance of the two notable and active areas of research, requirements engineering and Web has been much advocated.

We characterize that the relation between Web and Requirements Engineering (RE) can be interpreted in two ways. Firstly, RE can be viewed as a process for Web application development where it primarily looks into adapting of RE process to the Web applications which are special in characteristics as compared to traditional software applications. Secondly, Web can be viewed as a supporting technology for improving the requirements engineering process and enabling new capabilities. Thus, this relation can be examined from two different perspectives, each opening new avenues of research.

The intensive use of Web Applications has produced, among others, a rising interest in the development of methodological approaches providing a suitable support for the construction of Web applications (Escalona & Koch, 2004). Web Applications (Web Apps) necessitate a more extensive and efficient Requirements Engineering (RE) process as they are typical as possibility compared to conventional software. Key discerning characteristics which compel distinctive and special management are the heterogeneity of stakeholders involved, the diversity of the requirements, dynamic technology and environment (user needs), lack of communication with end-users and the high likelihood of risk.

Knowledge management in requirements engineering has persistently been a concern for better software development and maintenance. There is always some gap in understanding about what the business partners and stakeholders want, how software designers and managers design the modules and how software developers implement the design (Guha, 2011). With time, this gap is broadened due to the heterogeneous involvement of parties and constantly varying requirements of the software. Thus, the problem of requirement management manifolds with vague, inconsistent, incomplete, ambiguous and dynamic nature of requirements, and the much anticipated trend of Web is a savior. The web technologies serve as supporting technology that has found many valuable applications in the field of requirements engineering. Especially the advent of Web 2.0 and Web 3.0, the Social and the Semantic Web (Aghaei et al., 2012) respectively, has assisted in surmounting few challenges and limitations of requirements management.

Requirements Engineering

A *requirement* is a necessary, quantifiable, and verifiable capability, function, property, characteristic, or behavior that a product must exhibit to solve a real-world problem, or a constraint that it must satisfy or be satisfied during the development of a product (Ferreira & Loucopoulos, 2001). Requirements describe what the system will do without describing how it will do. Once these business needs also known as requirements are gathered, they are analyzed for their validity and studied for feasibility. *Requirements Engineering (RE)* is defined as the principles, methods, and tools for eliciting, describing, validating, and managing project goals and needs. It is an iterative and co-operative process with the objective to analyze the problem, to document the results in a variety of formats and evaluate the precision of the results produced (Ferreira & Loucopoulos, 2001). The iterative process of requirements engineering consists of three main activities, namely, requirements elicitation; requirements specification; requirements validation (Lowe, 1999). Some classic techniques to elicit, specify and validate requirements have been listed in Table 1 below.

Table 1. Description of techniques of RE phases

Requirements Elicitation	Requirements Specification	Requirements Validation
○ Interviewing ○ Joint Application Development (JAD). ○ Brainstorming ○ Concept Mapping ○ Sketching and Story Boarding ○ Use Case Modeling ○ Questionnaire and Checklist	○ Natural language ○ Glossary and ontology ○ Templates ○ Scenarios ○ Use case Modeling ○ Formal Description ○ Prototypes	○ Reviews or Walkthrough ○ Audit ○ Traceability metrics ○ Prototype for validation

RE is imperative for an accomplished proposal in terms of projection of the system, improved product quality and significant reduction in probability of project failure. Prolifically published studies have demonstrated consequences of not applying RE properly. Primary concerns are inadequate software architecture, unexpected problems like budget overrun, delay in production and "that's not what I asked for" and low user acceptance. The vital role of "requirement" in successful completion of project can be substantiated through much prolifically published work. (Field, 1997) tells that "projects fail too often because the project scope was not fully appreciated and/or user needs not fully understood." Likewise, (O'Brochta, 2002) says "the big problem with assessing project success is that it is not precise".

According to the 1995 *The Standish Group Report* (1995), the reasons for failed or challenged projects which deal to requirement engineering are incomplete requirement; lack of user involvement; lack of resources and unclear objectives. IBM systems magazine, on an article in 2012 on project pitfalls (Gulla, 2012) has defined 7 major reasons of failure of the project,

namely, poor project planning and direction; insufficient communication; ineffective management; failure to align with stakeholders and constituents; ineffective involvement of executive management; lack of soft skills or the ability to adopt; poor or missing methodology and tools. As we can observe, that there is an endless list of authors as well as so many survey reports (range of past two decades, including recent) which consider that RE is a critical initial stage, but often poorly executed and handled with less relevance (Kumar & Sharma, 2015). It is worth noting that challenged projects are those which were completed, but were over cost, over time, and/or lacking all of the features and functions that were originally specified.

The Emergence of Web

The World Wide Web is a huge, widely distributed, global source for information services, hyper-link information, access and usage information and web site content and organization (Bhatia & Khalid, 2008). During the last decade, the World Wide Web has evolved as an information space. The authors in (Aghaei et al., 2012) provide an overview of the evolution of the Web. It is summarized in the following Table 2.

Table 2. Evolution of Web

Evolution of Web	Characteristics
Web 1.0 (1991-2003)	○ It is a heteronym that refers to the state of the World Wide Web, and any website design style used before the advent of the Web 2.0 phenomenon. ○ Information-oriented Web. ○ It is Read only Web. ○ Connects real people to the World Wide Web ○ A Web of cognition.
Web 2.0 (2004-till date)	○ The term Web 2.0 is primarily linked with Web Apps that assist interactive information sharing, collaboration, user-centered design, and interoperability on the World Wide Web. ○ Social Web: connects real people who use the World Wide Web. ○ It is a Read/Write Web. ○ A Web 2.0 site gives its users the free choice to interact or collaborate with each other in the dynamic environment of social media in contrast to websites where users are limited to the passive viewing of content. ○ A Web of communication.
Web 3.0 (2006-Till date)	○ It refers to the latest version of the Web. ○ There are two different perspectives for defining Web 3.0. It can either be viewed as the **Semantic Web (or the meaning of data)** or as **intelligent web**, where software agents will collate and integrate information to give "intelligent" responses to human operators. ○ A Web of co-operation.

Requirement Engineering for Web

In the development of conventional (pre-Web) applications both practitioners and process experts regard requirements engineering as an imperative phase in the development process since the most common and time-consuming errors as well as the most expensive ones to repair, are errors that result from the inadequate engineering of requirements (Kumar & Sharma, 2015). The promising area of *Web Requirements Engineering* (WRE) which has been a point-of-discussion in numerous latest research projects and papers, entails addressing the technological challenges of requirements engineering activities for web applications. A Web Application is customization software developed for a network, based on technologies and standards of the World Wide Web Consortium (W3C) that offers an explicit purpose and functionality. These are programs which are written for the internet and to be hosted via a browser (Acunetix, n.d.). Fundamentally, till date, the Web-applications have been divided into nine categories (Deshpande & Hansen, 2001) with the development timeline depicting more sophistication and high complexity of new Web Apps. The Web evolution effectively fits in all the categories of web applications that exist till date. The following Table 3 depicts the same relation.

Table 3. Web Application Categories

Application	Characteristics	Web Evolution
Document Centric	○ precursor to Web applications ○ stored on a Web server as ready-made, that is static, HTML documents and sent to the Web client in response to a request ○ contents are frequently represented redundantly on several Web pages ○ simplicity and stability ○ short response time	Web 1.0
Interactive Web Applications	○ web pages and links to other pages are generated dynamically according to user input ○ uses CGI, ASP, PHP and HTML form ○ for example, virtual exhibitions, news sites, or timetable information	Web 2.0
Transactional	○ provide more interactivity ○ performing updates on the underlying content ○ efficient and consistent handling of the increasing amount of content ○ offer the possibility of structured queries ○ for example, online banking, online shopping, and booking systems	Web 2.0
Workflow	○ handling of workflows between different companies, public authorities, and private users ○ availability of appropriate Web services to guarantee interoperability ○ for example, Business-to-Business solutions (B2B solutions) in e-commerce, e-government applications.	Web 2.0
Collaborative	○ especially for cooperation purposes in unstructured operations (groupware) ○ communication between the cooperating users ○ support shared information and workspaces ○ for example, wiki, Weblogs, scheduling systems, e-learning platforms	Web 2.0

Social	○ anonymity traditionally characterized World Wide Web ○ Moving towards *communities of interest* ○ *Examples:* Blogs, collaborative filtering systems, social bookmarking (for example, del.icio.us) ○ Integration with other forms of web applications (e.g., NetFlix)	Web 2.0
Portal oriented	○ provide a single point of access to separate, potentially heterogeneous sources of information and services ○ for example, online shopping malls, community portals	Web 2.0
Ubiquitous	○ provides customized services anytime anywhere and for any device ○ facilitating ubiquitous access ○ either personalization or location-aware services or multi-platform delivery	Web 2.0
Semantic Web	○ present information on the Web ○ not merely for humans, ○ but also in a machine readable form ○ facilitate knowledge management on the Web ○ linking and reuse of knowledge (content syndication) ○ locating new relevant knowledge ○ interoperation on the semantic level and the possibility of automating tasks (via software agents)	Web 3.0

Classical requirement engineering methods are applicable for traditional system applications but developing web applications is distinctive. The key characteristics what make RE for Web "special" are multidisciplinary teams, unavailability of stakeholders, rapidly changing requirements and constraints, unpredictable operational environment, no manual for the user interface, content management (Ginige & Murugesan, 2001; Escalona & Koch, 2004). Thus, there is a strong need of some special, typical requirement engineering techniques which could handle these issues efficiently. Adapting RE methods to Web application development needs to address and focus on some important issues, viz., types of requirements vital for the Web application mainly identifying critical requirements; the level of documentation and description of these requirements; tools suited for the particular project needs. As a step to provide an insight to the RE adaptation to Web Apps, the authors in (Kumar & Sharma, 2015) present SWOT Analysis of Web Requirement Engineering that necessitates a holistic and proactive approach to Web Requirement Engineering to bring the current chaos in web application development under control, minimize risk, and enhance its maintainability and quality. The work done by authors in (Escalona & Koch, 2004) cites the state of the art of requirements engineering in methodologies used for the development of Web applications.

Web for Requirements Engineering

Requirements Engineering has a prominent effect on the success or failure of a software project. Inaccurate, inadequate, or misunderstood requirements are the most common causes of poor quality, cost overruns and late delivery of software systems (El Emam & Madhavji, 1995). Fundamental problems of requirement engineering have been widely acknowledged across

numerous published studies. Some examples of fairly common problems with the RE process are as follows (El Emam & Madhavji, 1995; Siddiqi & Shekaran, 1996; Sommerville & Kotonya, 1998; Nikula et al., 2000; Nuseibeh & Easterbrook, 2000; Hall et al., 2002):

- Vague requirements,
- Undefined requirements process,
- Inadequate requirements traceability,
- Lack of stakeholder involvement,
- Business needs are not considered,
- Lack of requirements management,
- The requirements do not reflect the real needs of the customers,
- Requirements are ambiguous (more possible interpretations), inconsistent (contains contradictions), and/or in complete (forgotten features),
- It is expensive to make changes to requirements after they have been agreed,
- There are misunderstandings between customers and software engineers,
- Requirements growth, and
- Stakeholders communication problems.

Specifically, the operative challenges motivating researchers in the area of Requirements Engineering can be categorized as either related to the stakeholders involved or to the requirements management, apart from some miscellaneous problems. The inception of Web, especially with the advent of Web 2.0 and Web 3.0, many momentous applications in the field of requirements engineering have been realized that assist in surmounting few challenges and limitations of requirements management and stakeholder communication issues. In this paper, we have tried to depict how Web has been utilized as a supporting technology to overcome some if not all issues related to requirements engineering phase. The following Table 4 portrays this relation of Web and requirement engineering where specifically the role of Web 2.0 and Web 3.0 is examined to improve requirements engineering process.

Table 4. Web as supporting technology for RE process

	Problem	Solution	Weakness
Stakeholders	**Lack of stakeholder involvement** **Stakeholders communication problems** o Participants are geographically distributed o Face-to-face contact is limited.	**Web 2.0** o Web-based, community-centered applications (Hippner, 2006) o Social Software Engineering (Lohmann et al., 2009) o Many existing commercial tools that support collaborative development of requirements. For example Rational RequisitePro (Rational Software Corporation, 2003), Borland Caliber RM (Whitehead, 2007), and Telelogic DOORS (Telelogic, 2004), RavenFlow (Ravenflow, 2007), Gatherspace (Gatherspace.com, 2007) and eRequirements (eRequirements, 2007).	o Security and privacy breaches in social applications. o Lack of Trust among participants o Collaboration among stakeholders happens independent of RE tools (Lohmann et al., 2009).
Requirements Management	**Vague Requirements** o Absence of key requirements. o Work based on incorrect supposition.	**Integration of ontologies (Web 3.0) and models** o Use of models such as petri nets to represent requirement, removing vagueness to some extent (Brockmans, 2006). o Petri nets require semantic annotation for formal analysis, ontologies have been used to provide semantic annotation to petri nets (Baek Jorgensen & Bossen, 2004; Gašević et al., 2009).	**Relying on Models such as Petri nets** o Require models to represent requirements in the form of use cases.
	Inadequate requirements traceability o High human effort for requirements categorization, conflict analysis and tracing, especially with a large number of requirements (Farfeleder et al., 2011).	**Semantic links (Web 3.0)** o Helps in establishing traceability. o Concepts in ontology can be associated with requirement specification items (Gašević et al., 2009; Zhao et al., 2009).	

	Requirements are ambiguous, inconsistent and/or incomplete o Diverse stakeholders, who produce different interpretations of the same requirement. Moreover, one stakeholder can interpret the same requirement in diverse ways. o Contains Contradictions o Incompleteness (Farfeleder et al., 2011) of requirements categorization and conflict identification, in particular, when performed manually.	**Ontology Reasoning (Web 3.0)** o Ontology reasoning with closed world assumption has been used to remove inconsistency and incompleteness among requirements (Siegemund et al., 2012).	**Relying on closed world assumption** o As requirements are dynamic and changing, relying on closed world assumption never seems to be a solution
	Dynamic and Changing Requirements o Constant requirements revision in order to help to understand new client's needs and to identify how they can be satisfied.	**Ontology (Web 3.0) driven requirement engineering (Siegemund et al., 2011)** o Requires ontology evolution (ontology maintenance) in order to adapt new changes (Flouris & Plexousakis, 2005) o Includes task such as ontology alignment, mapping, merging, articulation, versioning, et cetera.	o Complicated and multifaceted task (Flouris & Plexousakis, 2005) o Human intervention required o Cost incurred
Miscellaneous	o Business needs are not considered. o The requirements do not reflect the real needs of the customers. o There are misunderstandings between customers and software engineers.	**Web 3.0: Ontology based Computing (Lee & Gandhi, 2005)** o Use ontologies to enable communication (or mappings) among multiple disciplines (for example, to the engineering team members, to marketing, to management and to customers), o Employ semantic descriptions of application domains, o Designed to capture natural language descriptions of domains of interest, o Provide more consistent representations of such domains, o Reduce ambiguity inherent in communication, o Reduce error via the structuring of knowledge, and o Provide ways to extend and specialize captured domain knowledge.	

Thus, as a supporting technology, social web technology (Web 2.0) explores collaborative development of requirements and overcomes stakeholder problems to some extent. At the same time, Semantic web (Web 3.0) can be used in capturing, modeling, developing, checking, and validating of requirements specifications. It can play a vital role in all phases of requirement engineering such as in requirement elicitation (Al Balushi et al., 2006), requirement analysis (Caralt & Kim, 2007), requirement documentation (Bhatia et al., 2015) and requirement review (Kaiya & Saeki, 2005; Zhu & Jin, 2005a; 2005b).

Conclusion

We characterized that the relation between the Web and Requirements Engineering (RE) can be interpreted in two ways. As a process for Web application development, a need for more extensive and detailed requirements engineering is identified to bring the current chaos in web application development under control, minimize risk, and enhance its maintainability and quality. At the same time, the Web can impact requirement engineering processes to develop quality software by assisting in surmounting few challenges and limitations of requirements management. As a supporting technology, the Web especially the current generations, Web 2.0 and Web 3.0, enables new capabilities and has opened up new avenues of research.

References

Acunetix. (n.d.). Web applications: What are they? What of them? Retrieved March 10, 2015, from http://www.acunetix.com/websitesecurity/web-applications/

Aghaei, S., Nematbakhsh, M. A., &Farsani, H. K. (2012). Evolution of the World Wide Web: From WEB 1.0 tO WEB 4.0. *International Journal of Web & Semantic Technology*, 3(1), 1-10.

Al Balushi, T.H., Sampaio, P. R. F., Dabhi, D., & Loucopoulos, P. (2006, July). Performing requirements elicitation activities supported by quality ontologies. In *SEKE* (pp. 343-348).

Baek Jorgensen, J., & Bossen, C. (2004). Executable use cases: Requirements for a pervasive health care system. *Software, IEEE*, 21(2), 34-41.

Bhatia, M. P. S., & Kumar, A. (2008). Information retrieval and machine learning: Supporting technologies for web mining research and practice. *Webology*, *5*(2), article 55. Retrieved March 10, 2015, from http://www.webology.org/2008/v5n2/a55.html

Bhatia, M., Kumar, A., & Beniwal, R. (2015, March). Ontology based Framework for automatic software's documentation. In *Computing for Sustainable Global Development, 2015 2nd International Conference on* (pp. 725-728). IEEE.

Brockmans, S., Ehrig, M., Koschmider, A., Oberweis, A., & Studer, R. (2006, May). Semantic alignment of business processes. In *ICEIS* (3) (pp. 191-196).

Caralt, J. C. C., & Kim, J. W. (2007, January). Ontology driven requirements query. In *40th Annual Hawaii International Conference on System Sciences, HICSS 2007*, 3-6 Jan. 2007, pp. 197c.

Deshpande, Y., & Hansen, S. (2001). Web engineering: Creating a discipline among disciplines. *Multimedia, IEEE*, 8(2), 82-87.

El Emam, K., & Madhavji, N. H. (1995, March). A field study of requirements engineering practices in information systems development. In *Requirements Engineering, 1995, Proceedings of the Second IEEE International Symposium on* (pp. 68-80).

eRequirements. (2007). eRequirements product tour. Retrieved March 10, 2015, from http://www.erequirements.com/app?service=page/ProductTour

Escalona, M. J., & Koch, N. (2004). Requirements engineering for web applications-a comparative study. *Journal of Web Engineering*, 2(3), 193-212.

Farfeleder, S., Moser, T., Krall, A., Stalhane, T., Omoronyia, I., & Zojer, H. (2011). Ontology-driven guidance for requirements elicitation. In *The Semantic Web: Research and Applications* (pp. 212-226). Springer, Berlin.

Ferreira, M. J., &Loucopoulos, P. (2001). Organization of analysis patterns for effective reuse. In *ICEIS (2)* (pp. 766-773).

Field, T. (1997). When bad things happen to good projects: The warning signs of failure are detectable long before a project falls apart. *CIO-FRAMINGHAM MA*, 11, 54-56.

Flouris, G., & Plexousakis, D. (2005). Handling ontology change: Survey and proposal for a future research direction. *Institute of Computer Science, Forth. Greece, Technical Report TR-362 FORTH-ICS*. Retrieved March 10, 2015, from http://www.ics.forth.gr/isl/publications/paperlink/fgeo_TR362.pdf

Gašević, D., Kaviani, N., & Milanović, M. (2009).Ontologies and software engineering. In *Handbook on Ontologies* (pp. 593-615). Berlin: Springer.

GATHERSPACE.COM. (2007). Gatherspace: Requirements management and use case software. Retrieved March 10, 2015, from http://www.gatherspace.com/static/product2.html

Ginige, A., & Murugesan, S. (2001). Guest editors' introduction: The essence of web engineering-managing the diversity and complexity of web application development. *IEEE Multimedia*, *8*(2), 22-25.

Guha, R. (2011). Software engineering on semantic web and cloud computing platform. Retrieved March 10, 2015, from http://people.cs.pitt.edu/~chang/231/y11/papers/cloudSE.pdf

Gulla, J. (February, 2012). Seven reasons IT projects fail. Retrieved March 10, 2015, from http://www.ibmsystemsmag.com/power/Systems-Management/Workload-Management/project_pitfalls/

Hall, T., Beecham, S., & Rainer, A. (2002). Requirements problems in twelve software companies: an empirical analysis. *IEE Proceedings-Software*, 149(5), 153-160.

Hippner, H. (2006). Bedeutung, anwendungen und einsatzpotenziale von social software. *HMD–Praxis der Wirtschaftsinformatik*, 43(252 S 6).

Kaiya, H., & Saeki, M. (2005, September). Ontology based requirements analysis: Lightweight semantic processing approach. In *Quality Software, 2005. (QSIC 2005). Fifth International Conference on* (pp. 223-230). IEEE.

Kumar, A., & Sharma, A. (2015, April). Swot analysis of requirements engineering for web applications. *International Journal of Advance Research in Science and Engineering (IJARSE)*, 4(4), 34-43.

Lee, S. W., & Gandhi, R. A. (2005, December). Ontology-based active requirements engineering framework. In *Software Engineering Conference, 2005. APSEC'05. 12th Asia-Pacific* (pp. 8).

Lohmann, S., Dietzold, S., Heim, P., & Heino, N. (2009). A web platform for social requirements engineering. In *Software Engineering (Workshops)* (Vol. 150, pp. 309-315).

Lowe, D. (1999). Hypermedia and the Web: An engineering approach. John Wiley & Sons, Inc.

Nikula, U., Sajaniemi, J., & Kälviäinen, H. (2000). Management view on current requirements engineering practices in small and medium enterprises. *Queensland University of Technology*. Retrieved March 10, 2015, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.6132

Nuseibeh, B., & Easterbrook, S. (2000, May). Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering* (pp. 35-46).

O'Brochta, M. (2002, October). Project success - What are the criteria and whose opinion counts. In *PMI Annual Seminars & Symposium*.

Rational Software Corporation. (2003, June). Rational Requisite ProUser's Guide. Retrieved March 10, 2015, from ftp://ftp.software.ibm.com/software/rational/docs/v2003/win_solutions/rational_requisitepro/reqpro_user.pdf

Ravenflow. (2007). RAVEN: Requirements Authoring and Validation Environment. Retrieved March 10, 2015, from http://www.ravenflow.com/products/index.php

The Standish Group. (1994). *The Standish Group Report: CHAOS*. Retrieved March 10, 2015, from http://www.projectsmart.co.uk/docs/chaos-report.pdf

Siddiqi, J., & Shekaran, M. C. (1996). Requirements engineering: The emerging wisdom. *IEEE Software*, *13*(2), 15-19.

Siegemund, K., Thomas, E. J., Zhao, Y., Pan, J., & Assmann, U. (2011, October). Towards ontology-driven requirements engineering. In *Workshop Semantic Web Enabled Software Engineering at 10th International Semantic Web Conference (ISWC)*, Bonn.

Siegemund, K., Zhao, Y., Pan, J. Z., & Assmann, U. (2012). Measure software requirement specifications by ontology reasoning. In *8th International Workshop on Semantic Web Enabled Software Engineering (SWESE'2012)* Nara, Japan.

Sommerville, I., & Kotonya, G. (1998). *Requirements engineering: processes and techniques*. John Wiley & Sons, Inc.

Telelogic. (2004, February). *Getting Started with DOORS (DOORS 7.1)*. Retrieved March 10, 2015, from http://endymion.ugent.be/~stijn/doors/doors_getting_started.pdf

Whitehead, J. (2007). Collaboration in software engineering: A roadmap. *Proceedings of FOSE 07: Future of Software Engineering*, 2007, 214-225.

Zhao, Y., Dong, J., & Peng, T. (2009). Ontology classification for semantic-web-based software engineering. *IEEE Transactions on Services Computing*, 2(4), 303-317.

Zhu, X., & Jin, Z. (2005). Ontology-based inconsistency management of software requirements specifications. In *SOFSEM 2005: Theory and Practice of Computer Science* (pp. 340-349). Springer Berlin Heidelberg.

Zhu, X., & Jin, Z. (2005, June). Inconsistency measurement of software requirements specifications: An ontology-based approach. In *Engineering of Complex Computer Systems, 2005. ICECCS 2005. Proceedings. 10th IEEE International Conference on* (pp. 402-410). IEEE.

Bibliographic information of this paper for citing:

Kumar, Akshi, Bhatia, M.P.S., & Beniwal, Rohit (2015). "Characterizing relatedness of web and requirements engineering." *Webology*, 12(1), Article 136. Available at: http://www.webology.org/2015/v12n1/a136.pdf