A Process Model for Developing Semantic Web Systems

Amjad Farooq and M. Junaid Arshad

Department of Computer Science and Engineering, University of Engineering and Technology,
Lahore (54890), Pakistan
amjadfarooq@uet.edu.pk junaidarshad@uet.edu.pk

Abstract: Before the Web era various software development methodologies have been proposed for the development of software applications for different domains. The main objectives of those methodologies were to meet user's requirements, find out means to suggest a systematic software development and reduce the maintenance cost of the developed software. On the emergence of the Web and to develop the web-based software systems, some existing methodologies have been extended. Also, new approaches (or informal methodologies) are introduced for the development of web-based systems because the development process for these systems is not considered as an extension of the classical software engineering, although both development processes for web-based systems and non web-based systems have the same basic objective which is software development. Of course, the development of the web-based systems needs a new kind of development methodologies which should meet and capture their unique and different requirements. Currently available software development methodologies are inappropriate and unsuitable to use for the development of web-based software systems, especially for the third generation web, called Semantic Web. In this paper, we present a brief review of the existing software development methodologies for the development of web-based systems. Some informal software development methodologies (or approaches) for the semantic web are also reviewed. Then, based on this analytical review, we propose a model for the development of semantic web systems. This model can be used as a benchmark to propose formal methodologies for the development of the semantic web systems. [New York Science Journal 2010;3(9):34-39]. (ISSN: 1554-0200).

Key words: Semantic Web, Ontology, Process Model, Software Engineering, Web development.

1. Introduction

Most of the early software development methodologies were proposed using the function-oriented approach. Their main objective was the systematic software development that could meet user requirements at a reasonable development and maintenance cost. With the passage of time software reusability, interoperability and integration problems raised, and they became the main motivation of several object-oriented methodologies, and many such methodologies are proposed and reported in the literature. But these methodologies are unsuitable to use for the development of software for the third generation web because there is a need of machine understandable semantics of web contents in this type of software systems.

Since most of the web engineering methodologies don't support this need, i.e., machine understandable semantics of web contents.

As the nature of semantic web systems differs from the nature of non-semantic web-based systems (Yuhui et al., 2001), therefore, the need of methodologies of semantic web engineering is essential and urgent (Tim et al., 2001; Moura and Schwabe, 2004). A few efforts have been done in the recent years by annotation approaches such as Creating Relational Annotation-based Metadata for the Semantic Web CREAM to achieve this objective (Handschuh et al., 2001), Simple HTML Ontology Extensions (SHOE) (Heflin et al., 1999), Semantic Markup, Ontology and RDF Editor (SMORE) (Booch, 1991) and the methodologies such as Semantic Hypermedia Design Method (SHDM) (Frasincar, 2002), Hera (Frasincar, 2002), OntoWeaver (Brickly et al., 2004), OntoWebber (Yuhui et al., 2001; Yuhui et al., 2002), SEmantic Web PortALs (SEAL) (Maedche et al., 2002), Web Engineering for Semantic Web Applications (WEESA) (Reif et al., 2005), Semantic Web Development with WSDM (Plessers et al., 2005) have been proposed. Most of them are still in their preliminary stages, and not mature enough to be used. Also, these approaches are not based on the principles of software engineering, and they cannot be considered as a complete software development methodologies.

The rest of the paper is organized as follows. In section 2, we give a short overview of existing methodologies and approaches for non web-based, web-based and semantic web-based software development. We propose a model for the development of semantic web software in section 3, and the paper is concluded with future work directions.

2. Literature Review

In this section, we give review and analysis of three types software development methodologies, i.e., i) non web-based, ii) web-based, iii) semantic web.

2.1 Methodologies for Non-Web based software

In the beginning of the data processing era many software development methodologies such as Structured Design (Bergland, 2002), Structured Analysis and Design Technique (SADT), Data Structured System Design, Jackson System Development (Michael, 1992; Web) were proposed. Main emphasis in most of these methodologies was on function gathering, and the approach used in them is referred to as the function-oriented approach.

In these function-oriented methodologies, some formal methodologies have also been proposed to develop high-quality software systems (Yuhui et al., 2002). Their main objective was the systematic development and meeting user requirements at a reasonable cost and minimum maintenance cost after their development. After the introduction of object-oriented approaches, many object-oriented software development methodologies such as Object Modeling Technique (OTM), Booch Methodology, Yourdon object-oriented methodology, Fusion Methodology, Object-Oriented Design Methodology (OODM) and many more were developed and reported in the literature (OODM, 2003; Web).

The main emphasis of the object-oriented approach and methodologies is on the identification and gathering of object-classes of a system that is under-development. These methodologies suggest schemes for the object-oriented analysis and design, and phases of the development. The object-oriented methodologies differ in their processing steps (or phases) to do analysis, design and other phases. A complete comparison and study of some popular object-oriented methodologies is available in (John, 1995; Embely et al., 1995).

2.2 Methodologies for development of web-based Systems

The systemically development process to develop web-based software systems is referred to as the web engineering. To the best of knowledge for the development of web-based software systems, OODM is reported in the literature (Gomez et al., 2000). The web engineering adopts and encompasses many software engineering principles. It suggests many new approaches and techniques, and their main objective is to meet the unique and different requirements of the web-based systems.

Beside OODM (Shah, 2003), many other methodologies, tools and techniques such as Web Modeling Language (WebML) (Ceri et al., 2000), Object-Oriented Hypermedia Design Method (OOHDM) (Schwabe et al., 1996), UML-based Web Engineering (UWE) (Koch and Kraus, 2002), XML-Based Web Engineering (Reif et al., 2005), Component Based Web Engineering (CBWE) (Berg, 1997), Relationship Management Methodology (RMM) (Isakowitz et al., 1995), Cocoon (Ceri et al., 2000), MyXML (Kerer et al., 2000) and Object-Oriented Hypermedia Method (OOH) (Gomez et al., 1996) have been used (with and without modifications) for the development of web-based systems. A detailed review and comparison of some of them are reported in the literature like (Shah, 2003; Barna et al., 2003).

2.3 Methodologies for Development of Semantic Web-Based Systems

Web ontology is considered as the backbone of a semantic web system as it models a domain. During modeling domain ontology, its terms are defined for making them machine understandable, and relationships between them are also defined. Semantic annotations via ontologies have already been started for the semantic web systems. It is a process that transforms a web system into a semantic web system by augmented their contents with metadata that formally defines and makes them machine understandable. This metadata is generated using the languages such as RDFS (Brickley et al., 2004), DAML+OIL (Connolly et al., 2001) and OWL (Peter et al., 2004).

These languages are recommended by World Wide Web Consortium (W3C). Also, several annotation techniques/tools have been proposed and developed such as Creating RElational Annotation-based Metadata for the Semantic Web CREAM (Handschuh et al., 2001), Simple HTML Ontology Extensions (SHOE) (Heflin et al., 1999), Semantic Markup, Ontology and RDF Editor (SMORE) (Kalyanpur et al., 2003). Most of these supporting tools only allow annotating static websites, page-by-page at the implementation level (phase) although it is totally a misconception of the idea of semantic web stated by T. B. Lee, i. e., semantic web is not about marking HTML documents, it is about the data stored in relational databases, XML documents, spreadsheet, etc (Updegrove, 2005).

Even the approaches (such as CREAM (Handschuh et al., 2001)) which support the annotation of the dynamically generated websites (by annotating the database), and create a direct link between the implementation structures of the database. It is also observed that for both static and dynamic websites, every time if someone changes the implementation of the website or database, even though nothing has been changed in the semantics of the presented data, then the defined linkage between the web pages or database and the ontologies can be affected. This situation urges a need of a conceptual schema of a web system to generate machine understandable semantics of web contents (Brickley et al., 2004). Another reason for the need of conceptual schema of a semantic web system is to retrieve the relevant information from the Web now becomes very difficult and time consuming, because the existing Web contents are in huge volume, not semantically rich, no relationships among them, and not machine-understandable.

Table 1. Summary of approaches used for extension of XML-based & WSDM methodologies

Parameters	WSDM	WEESA
Conceptual Mapping method	Manual	Manual and Support tool
Type of software	A complex architecture based software	Based on a support tool and mapping algorithm Processor
Static pages Annotation	Yes	Yes
Dynamic pages Annotation	Yes	Yes but restricted
Mixed content annotation	No	Yes
Database Annotation	Yes	To a certain Extend
Integration with web engineering approaches	Yes	Yes
Existing frame work Integration	No	Yes: with the Cocoon is proposed
Multiple Ontologies Support	Yes	Yes
Reuse of Annotation	Yes	No
Scope of the Software	More Generic model for annotation of web documents.	Target only XML pages

Most of the available web engineering methodologies, tools and techniques do not (or partially) meet this important need, because the present web engineering is different from the classical software engineering because in the web engineering, the principles of classical software engineering are not followed. Also, semantic web engineering and web engineering are different from each other because of the same reasons. Therefore, we are badly in need of methodologies for the development of semantic web systems, and these methodologies should be based on the principles of the classical software engineering. In other words, these semantic web system development methodologies should have all software development phases, i.e., Analysis Phase, Design Phase, Implementation and Testing Phase and Maintenance Phase. As far as the authors are aware of, the more appropriate models for development of semantic web systems are ‘Semantic Web Development with WSDM (Plessers et al., 2005) and WEESA (koch et al., 2002). They are briefly compared in the Table 1 and each parameter is briefly described below:

- ***Conceptual Mapping:*** In WEESA the conceptual mapping is done manually and also a support tool is available for define the mapping. This mapping is central part of the WEESA technique so it is very time consuming because it is done manually. In WSDM the mapping is done in the task modeling phase and is done manually.
- ***Type of Software:*** WEESA consists of a GUI based support tool and a java method processor for annotation purposes. On the other hand the WSDM is complex architecture software for annotation purposes.
- ***Static pages Annotation:*** WEESA support the static page annotation, for this purpose we have to define a mapping from that page schema and ontologies. WSDM also fully supported the annotation on static pages.
- ***Dynamic pages Annotation:*** WEESA support the dynamic page annotation, for this purpose we have to define a mapping from that page schema and ontologies but there is a limitation that the database keys should be accessible through the XML tags. On the other hand WSDM also fully supported the annotation on dynamic pages. For this purpose a very generic phase is proposed in WSDM called Data Design Phase.
- ***Mixed content annotation:*** WEESA annotates page on the basis of the structure of the XML document so the mixed contents cannot be annotate through it but it could be cover through the concepts in the Ontologies but there is not well define method in it. On the other hand WSDM used the object chunks to annotate the page so it can annotate the mixed contents very well.
- ***Database annotation:*** WEESA could annotate the data base but there is a limitation that the database keys should be accessible through the XML tags but WSDM fully supported the database annotation by making a BIM model for displaying the information. And generate related database from it.
- ***Integration with web engineering approaches:*** Both WEESA and WSDM can be integrated in the existing web engineering methodologies. Actually these are proposed extensions in previous web engineering approaches. WEESA is integrated in the XML based web engineering and WSDM is extension of previous WSDM web engineering approach. WEESA proposed integration in XML web engineering frame work called the CoCoon. No practical integration of WSDM is proposed. It is totally a new methodology.

- ***Multiple Ontologies Support:*** Both WEESA and WSDM support the multiple Ontologies concept in their approaches.
- ***Reuse of Annotation:*** In WSDM we can reuse the annotation for one page for another page but using the object chunk of that page but in WSDM no such method is given for reuse of the annotation.
- ***Scope of the Software:*** WEESA particularly targets the XML documents and supports XML syntax on the other hand WSDM is more generic approach for annotation of pages.

3. Proposed Model

Semantic web software consists of machine understandable content as well as human understandable content so-called web pages. In the proposed model we have focused on both of these points. The sketch of our proposed model is shown in Figure 2. In the proposed model, the two major activities; i) generation of web pages, ii) construction of logical content (or ontology), are carried out in parallel (see Figure 2). Then, their integration and testing is performed to produce a machine understandable as well as human understandable final product. The phases involved in the model are described in Figure 1.

3.1 Analysis Phase

The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished. In this phase it is determined, what are the client's needs along with what the client wants. The deliverable result at the end of this phase is a requirement document. Since there are two types of requirements: human as well as machine understandable contents, business analyst, web engineer and ontology engineer are involved to determine and analyze these requirements. Different activities involved in this phase are presented in Figure 2 and each parameter is briefly described below:

Requirement Determination**:** In this phase, the target humans are identified and they are grouped into classes, having the same functional requirements. For each class usability requirements are listed. Web engineer completes this task with the collaboration of business analyst.

Knowledge Acquisition**:** This activity is the prerequisite of ontology construction track as mentioned in figure 1. Several relevant knowledge sources such as organization charts, employee role descriptions, business plan, internal documents, dictionaries, index lists, regulations etc are collected and analyzed. A document of descriptive knowledge about system domain is prepared.

Formal Vocabulary**:** It is the fundamental activity for the semantic web software systems. Controlled vocabulary is prepared with consensus of ontology engineer and web engineer to avoid semantic heterogeneity, with respect to system domain. This vocabulary is the basis for the next activities as show in Figure 2.

Requirements Modeling**:** The output of 'requirements determination' activity is rewritten using controlled vocabulary in some formal way according to the modeling standards.

Formal Specification**:** The descriptive knowledge obtained in the knowledge acquisition activity is organized in classes, subclasses, properties, sub-properties, relationships, constraints & rules using controlled vocabulary.

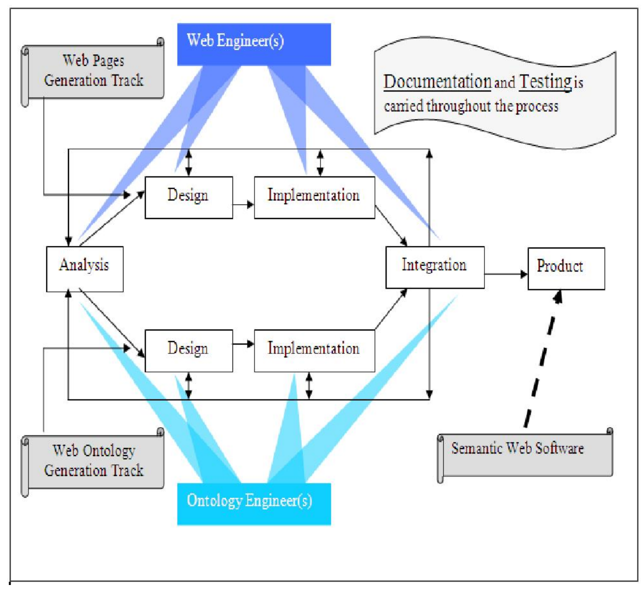

Figure 1. The architecture of proposed model

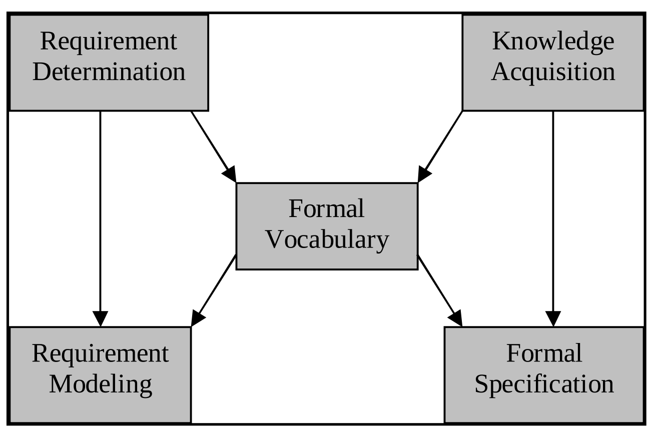

Figure 2. Activities of analysis phase

3.2 Design Phase

As stated in the previous phase, there are two types of requirements for a semantic web application: one for human and other for machine. For the first type of requirements the following components are designed by using pre-defined software engineering standards for the output of previous phase i.e. requirements modeling document: Navigation, database, pages, templates and presentations. Whereas, for the second types of requirements. i.e., machine understandable content generation. Classes are categorized in term of domains.

An ontology diagram is produced for each domain. Rules are described in terms of constraints and triggers – when occurs and what actions need to performed. Since ontologies can be reused and shared, so before designing a new ontology for any domain, first of all existing ontologies are examined in order to find the suitable ontology for that domain. If some relevant ontologies are found, they are included in the output document of this phase. Since different terms can be used to represent the same sense ((price, rate), (big, large, and huge)), (named entity: item, product, stock). In this phase the relevant terms are determined and their mappings to the compatible term in domain model are performed. RDF graph annotated with integrity constraints, domain, range specifications, and cardinalities is produced.

As it is said “Writing code isn't the problem, understanding the problem is the problem". Ontology implementation languages are available such as OWL (Peter et al., 2004). Also several tools are available for the assistance of ontology creation, such as Altova SemanticWorks™ 2006 (Golbeck, et al., 2002). This is very simple tool, it allows you to graphically create ontology document in OWL and RDFS, and so you can create valid documents quickly and easily. Similarly, for developing pages, applets, procedures for business logic, number of tools are available those allow developers to make development very quickly and easily.

Logical content i.e. ontology is hooked with web software in this phase. The knowledge base is populated with ontology instances, and it is updated dynamically. This will be used for semantic search engines as well.

4. Conclusions and Future Work

In this paper two semantic web engineering methodologies WEESA and Semantic Web Development with WSDM have examined, along with some web engineering approaches. It is determined that most of them are focus on data rather than knowledge during development of software. We have proposed a model to develop semantic web software. The main theme is the incorporation of declarative knowledge into the web software in some formal way (i.e. RDFS, OWL) process-able by machine. First it is transformed into ontology then it is used for annotation. The proposed model can be used as a benchmark to propose informal methodology for the development of software systems for the semantic web.

Acknowledgements:

This work was supported by the Directorate of Research Extension and Advisory Services, University of Engineering and Technology (UET), Lahore-Pakistan.

Corresponds to:
Mohammad Junaid Arshad, PhD
Department of Computer Science & Engineering,
University of Engineering and Technology,
Lahore-Pakistan
Email: junaidarshad@uet.edu.pk

References

Moura, S., Schwabe, D.: Interface Development for Hypermedia Applications in the Semantic Web. In Proceedings of LA Web 2004, Ribeirão Preto, Brasil. IEEE CS Press, ISBN 0-7695-2237-8 (2004) 106-113

Reif, G., H. Gall, M. Jazayeri. WEESA - Web Engineering for Semantic Web Applications. 14th International World Wide Web Conference, Chiba, Japan, 2005.

Yuhui, J., Stefan Decker, and Gio Wiederhold. Ontowebber: Model-driven ontology based web site management. In Semantic Web Working Symposium (SWWS*)*, Stanford, California, USA, August 2001.

Yuhui, J., Sichun Xu, Stefan Decker, and Gio Wiederhold. Managing web sites with ontowebber. In Proceedings of the 8th International Conference on Extending Database *Technology*, pages 766.768, London, UK, 2002. Springer-Verlag.

Frasincar, F., Houben, G.-J.: Hypermedia presentation adaptation on the semantic web. In Adaptive Hypermedia and Adaptive Web-Based Systems, Second International Conference, AH 2002, volume 2347 of Lecture Notes in Computer Science. Springer, ISBN 3-540- 43737-1 (2002) 133-142

Shah, A., OODM: An Object-Oriented Design Methodology for Development of Web Applications a Book Chapter: Pages: 189 - 229. Year of Publication: 2003, ISBN: 1-591-40050-3, King Saud University, Kingdom of Saudi Arabia

Ceri, S., P. Fraternalli and A. Bongio, “Web Modeling Language (WebML): a Modeling Language for Designing Web Sites”, *WWW’00*, Elsevier, Amsterdam, The Nederlands, May, 2000, pp. 135-157. http://members.aol.com/kaizensepg/methods.htm

Tim Berners-Lee, James Hendler, and Ora Lassila. The SemanticWeb. *Scientific America*, 284(5):34.43, 2001.

Brickley, D., and Ramanathan V. Guha eds. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 10 February 2004.

http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

Berg, K., Component-Based Software Development: No Silver Bullet, Object Magazine, 3-97, 1997.

Isakowitz, D., E. Stohr and P. Balasubramanian, RMM: A methodology for structured hypermedia design", Communication of the ACM, 38(8), 34–44, 1995.

Maedche, A., S. Staab, R. Studer, Y. Sure and R. Volz, " Seal - Tying up Information Integration and Web Site Management by Ontologies", IEEE Data Engineering Bulletin, 25(1), pp. 10-17, 2002.

Frasincar, F., G. Houben and R. Vdovjak, "Specification Framework for Engineering Adaptive Web Applications", In Proceedings of the Eleventh International World Wide Web Conference, 2002.

Schwabe, D., G. Rossi and S. Barbosa, "Systematic Hypermedia Application Design with OOHDM", In Proceedings of ACM Hypertext'96 Conference, pages 116–128, 1996.

Gómez, J., C. Cachero and O. Pastor, "Extending an Object- Oriented Conceptual Modelling Approach to Web Application Design", In Proceedings of CAiSE'2000, LNCS 1789, Pages 79–93, Stockholm, 2000.

Koch, N., and A. Kraus, "The Expressive Power of UML-based Web Engineering", Second International Workshop on Web-oriented Software Technology (IWWOST 2002), May 2002.

Connolly, D., Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. DAML+OIL (March 2001) Reference Description. W3C Note, 18 December 2001. http://www.w3.org/TR/daml+oil-reference.

Peter, F., Patel-Schneider, P. Hayes and I. Horrocks, "OWL Web Ontology Language Semantics and Abstract Syntax", W3C Recommendation, 10 February 2004. http://www.w3.org/TR/owl-semantics/.

Golbeck, J., M. Grove, B. Parsia, A. Kalyanpur, J. Hendler, "New Tools for the Semantic Web", Proceedings of EKAW 2002, LNCS 2473. Springer (2002) 392–400

Handschuh, S., S. Staab, A. Maedche, "CREAM – Creating Relational Metadata with a Component-based, Ontology Driven Framework", Proceedings of K-Cap, Victoria Canada, 2001.

Heflin, J., J. Hendler and S. Luke, "SHOE: A knowledge Representation Language for Internet Applications", Technical report CS-TR-4078 (UMIACS TR-99-71), 1999.

Booch, G. "Object-Oriented Design with Applications", Redwood City, California Benjamin Cumming. 1991.

Wirfs-Brock, R., et al., "Designing Object-Oriented Software", Englewood Cliffs: NewJersey: Prectice Hall 1991.

Embely, D., R. Jekson and S. Woodfield, "Object-Oriented Analysis: Is it or isn't it?", IEEE Software, 19-33, July 1995

Kerer, C., and E.Kirda. "Web Engineering, Software Engineering and web application development", In 3rd Workshop on Web Engineering at 9th WWW conference. May 2000, Amsterdam, the Netherlands, May 2000.

Apache Cocoon Project Homepage– www.Cocoon.apache.org

Updegrove, A., The Semantic Web: An interview with the Tim Berners-Lee, June 2005.

Torres, V., et al., "Getting Ready Web Engineering Methods for the Semantic Web", Putting Ontologies into Practice (2004).

Kalyanpur, A., J. Hendler, B. Parsia and J. Golbeck, "SMORE – semantic markup, ontology, and RDF", editor. Technical report, University of Maryland, 2003. ttp://www.mindswap.org/papers/SMORE.pdf.

Plessers, P., S. Casteleyn, O. Troyer, "Semantic Web Development with WSDM", In 5th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2005), Galway, Ireland (2005).

Michael J., The Jackson Development Methods in Wiley Encyclopaedia of Software Engineering, Marciniak, 1992.

Bergland, G.D., Structured Design Methodologies, Bell Telephone Laboratories, Inc. Information Processing Research Department Murray Hill, New Jersey, 2002. http://en.wikipedia.org/wiki/Web_engineering

Software design methods: http://www.levela.com/software_design_swdoc.htm

Barna, P., F. Frasincar, G. Houben and R. Vdovjak, "Methodologies for Web Information System Design", ITCC 2003: pp. 420-424.

John, A., J. Hamilton, "A survey of object-oriented methodologies", Annual International Conference on Ada, Proceedings of the conference on TRI-Ada '95: Ada's role in global markets: solutions for a changing complex world Anaheim, California, USA, pp. 226–234, 1995.

6/15/2010