IRBAS
www.irbas.academyirmbr.com
April 2017
International Review of Basic and Applied Sciences
Vol. 5 Issue.4

Formal Specification and Verification of Web Semantic Design Methodology (WSDM)

NAZIR AHMAD
Assistant Professor; Government Postgraduate College Sahiwal, Pakistan.
Email: nazir313@gmail.com
Tel: 00923046241104

ZIA-UR-REHMAN
MS Scholar; PMAS- Arid Agriculture University Rawalpindi, Pakistan.
Email: zia120rehman@gmail.com
Tel: 00923045434212

MUHAMMAD TAHIR MUMTAZ
MS Scholar; PMAS- Arid Agriculture University Rawalpindi, Pakistan.
Email: tahirmscs@gmail.com
Tel: 00923467490248

MUHAMMAD SAJJAD
Assistant Professor; Government Postgraduate College Sahiwal, Pakistan.
Email: sajjad@gpgcs.edu.pk
Tel: 00923346286759

GHULAM ALI
University of Central Punjab Lahore, Pakistan.
Email: ghulamali@ucp.edu.pk
Tel: 00923016010309

Abstract

Current web based technologies are not good and accurate for the purpose of finding correct data from the web resources. They also failed to extract the required data. Data representation, data interpretation and data maintenance are the main issues of current web technologies. These problems can be reduced by using semantic web vision by giving contents well-defined meaning via ontologies to enable them machine-understandable. Web portals provide forums to read, write and exchange change data over the web to a group of people having shared interests. The existing web portals have been developed using current web technologies. As a result there are a number of issues for data interchange between users. To overcome these limitations, Semantic Web technologies have been developed and being used. In this paper, a new approach based on formal methods to develop web portals has been proposed. Formal methods are the techniques based on mathematical logic and have been used for the specification, design and analysis of software systems. WSDM, Web Semantic Design Method has been used as a basis for this paper. The core components such a Mission Statement, Audience Classification, Audience characterization, Task Modeling and Information Modeling has been formalized in Z, a formal specification and modeling language. The resulting models have been verified by using Z/EVES theorem prover.

Keywords: *Semantic web, WSDM, Ontology, Formal Methods, Z notation, Z/EVES, Verification.*

Introduction

A Web portal is a website which provides data and information to users in the form of entertainment, news, games etc. Web Portals are customized entry points to web information and a better customization and personalization is required (Bajracharya et al. 2010). A number of web portals have been developed to provide a forum for users to discuss and share ideas with other users (Kwak et al. 2011) and (Lendyuk & Rippa 2011a). Presently, there are many issues with current web technologies being used for the development of web portals (Ding et al. 2010). These problems can be reduced by using semantic web technology in which contents are given well defined meanings by using ontologies to enable them machine-understandable (Christensen et al. 2010) and (Arnold et al. 2010)and this technology enables the use and access of automated information based on machine process able semantics of data. Ontologies provide a similar approach as of human thinking methodology and have roots in philosophy (Goméz-Peréz et al. 2004. The semantic web provides a mechanism so that web can be made “understandable” by the machines and the users (Vij et al. 2012). In this paper, we propose a technique for developing ontology-based web portals. The main focus of this research is on the design methodology of semantic web portals, not on the implementation side. The main issue of existing web portals is the presence of inconsistencies and incompleteness in their design. There is no way to analyze the design before implementation. This new technique is based on the use of formal methods for the design of web portals. Formal methods are the design techniques based on mathematical logic. Formal methods have the analysis and verification facilities in the form of model checkers and theorem provers. Z, a formal specification and modeling language has been used for the specification of this methodology. Z/EVES, theorem has been used to analyze the resulting models in Z. We have formalized a subset of Web Semantic Design Methodology (WSDM). This is the first step in this direction. We will extend our research to completely formalize WSDM. The models have been be analyzed by using Z/EVES theorem prover. The remaining sections of this paper are as follows: Section 2 describes the relevant literature review, section 3 describes formal methods, section 4 is about the methodology, section 5 presents the formal analysis of the models developed in sections 4 and section 6 concludes the paper. At the end there are references to literature consulted during the writing of this paper.

Related Work

The vision of semantic web came from Tim Berners-Lee who is known as the “father of Web”. It is the ability of the current web to store and spread large amount of information to very large audience very quickly but finding specific information in a very short time is very difficult (Ringwald et al. 2011) and (Aine et al. 2011). In finding specific information, the user retrieves a large amount of irrelevant material and misses the actual one (Preiser-Houy & Navarrete 2010) and (Whetzel et al. 2011). A web portal is a website that provides a number of information and data about news, games, movies etc to its users (Ding et al. 2010). Through the use of web portals users can have an opportunity of an open discussion forum to access and present information in an effective and efficient way (Lendyuk & Rippa 2011b). Ontologies are very important components of semantic web and are used for knowledge management and representation (Christensen et al. 2010) and (Bais et al. 2010). In Semantic Web, World Wide Web is enhanced and meaning are given to web contents for the purpose of making the web understandable for machines and human users (Ringwald et al. 2011) and (Pommier & Le-Trong 2011). Some web development methods are based on model driven approaches like HERA , WebML, UWE , SHDM/OOHDM. But these model do not describe the process that how to integrate different data sources and distributed web applications into a single instance of web portal. The solution to this problem has already been observed in several portal solutions by using ontologies. Different portal solution techniques and different projects like OntoPortal, The AKT Project, OntoWeaver-S etc. are ontology based solutions to web portal.

The developing environment of the tools was similarly the reason for numerous growth linked issues, as like restricted options to look for understanding as well as provision launch traditional designer groups. Next, the innovation of semantic net tools covered a lot of benefits, as like allowing scholars and designers additional innovative liberty in progress of scientific way out and importantly inspiring instructor and learner participation in the improvement procedure (Jesper & Nina. 2016).

Formal methods are approaches based on mathematics and have tool support to analyze and verify software systems. Formal methods are applicable to specification, architecture, design, and implementation and testing. A very limited use of formal methods is to specify requirements with the help of automated tools to check completeness, traceability, verifiability and reusability (Haxthausen 2010), (Hall 1990), (Clarke & Wing 1996), (Bowen & Hinchey 1995b) and (Bowen & Hinchey 1995a). Formal methods have been used very successfully for the specification and designing of a number of software as well as hardware systems (Woodcock et al. 2009). Design of software systems has been constructed by using formal methods (Jackson 2001) anrd (Woodcock & Davies 1996). There are a number of useful formal methods to verify the code.

There are a number of software verification tools by which models can be verified for issues such as inconsistencies, incompleteness etc. All these tools are based on formal methods (Saaltink 1997b), (Woodcock et al. 2009) and (Meisels & Saaltink 1997). There are a number of formal specification and modeling languages such as Z which are being used for writing software specifications (Jacky 1997). Other examples of formal specification languages are VDM++, Alloy, B-method etc.

The model checking tools takes a model as input and can show that it is correct with reference to requirements. All this process is done automatically by model checking tools. If there is an error in the model or found an inconsistency, the model checker generate counter examples to show where the error is. Model checkers are very efficient tools but there is a problem of state space. When a model grows, the number of possible spaces to reach, increases exponentially. A number of approaches have been designed and developed to overcome the problem of state space. For example, Binary Decision Diagrams. Model checking tools being developed which has been used in industry successfully include SPIN, Alloy, etc. With the help of Model checking tools, we can check logical inconsistencies in the specification, incomplete properties, deadlocks, race conditions etc.

Theorem provers are tools to formally verify and validate formal specifications automatically and semi-automatically. Some theorem provers prove theorems about required properties automatically such as HOL, Z/EVES etc. Some theorem provers require guidance from a human expert to verify complex properties (Saaltink 1997a), (Saaltink 1997b) and (Meisels & Saaltink 1997).

Some projects where formal methods are successfully applied are: The Transputer Project, where programming language was Occam and floating-point unit was specified in Z. Railway Signalling and Train Control, where SACEM system was specified and verified by using B-method which increased the network traffic 25% for PER railway system. Mondex Smart Card, an electronic cash system for cash transactions. It was developed by using formal methods as per ITSEC level E6 standard. Z formal method was used to produce proofs of security model of Mondex. AAMP Microprocessor was formally verified by PVS. Airbus has used SCADE for the development and verification of all software components (Woodcock et al. 2009).

Formal Methods

Formal methods are techniques and tools based on mathematical logic. Formal methods are used for specification, design and verification of software as well as hardware systems. Formal methods have shown great success in these areas. By using formal methods, we write statements in mathematical logic and formal verification is performed by deducting rules in that logic. By using formal methods, we can analyze the whole state space of a software as well as hardware system and we can show that a property is correct and true for all possible inputs. This is a great benefit which is impossible with testing. Formal methods can be applied at various stages of software development process ranging from requirements gathering to implementation. Application of formal methods reduces the overall cost of software development. There are a number of formal methods but there is no single formal method which serves all the purposes. Therefore, every formal method has a specific purpose. Some formal methods are used for writing specifications of software systems while others are used for analysis and generating formal proofs of software systems. Formal methods and other software design methods are different in the process of verification. The actual principle is that the software system must be proved correct before it is implemented. Traditional system design methodology uses huge testing methods in order for the verification of software systems but using testing, we can verify only a subset of input values. Other domains of input remain unverified which is a major cause of system failure. On the other hand, by using formal methods, we can show that system behaves as accepted by proving theorems about the properties of the system. It must be noted that formal verification does not remove the need of testing the software systems. By formal verification, we cannot remove bad assumptions in the design of software systems.

Methodology

Informal Model

WSDM is a design methodology to develop semantic web applications. There are a number of phases in this methodology ordered in the sequence (figure 0 shows all the phases). A sub phase may inherit data from super phase. All the phases are in sequence where the output of each phase becomes the input of other phase. For each phase there is a method to produce output from a given input. WSDM is based on audience-driven approach where different visitors and their requirements are base components for the design and structure of web systems. Different navigational paths, each one for different kind of visitor, are originated from home page. The mission statement is the starting component of the methodology during which three components: purpose, the subject and the target users are identified. The target users are users of interest to our system. The subject of the web system is the course related to the purpose and the target users of the web system. The subject must be updatable with reference to target users and in accordance with the purpose of the web system. Topics in the purpose must be listed clearly. Also target users must be identified. These may be very technical users instead of general users. The subject must also clearly be specified and listed. The subject includes all the functionality that is required by a particular user in the system. The information and functionality will be different for professionals and common public. The mission statement can also be used for validation purpose. To state the methodology, we will use a case study of online movie web site. The case study is about a movie store which provides information about movies, games, movie show times etc to users.

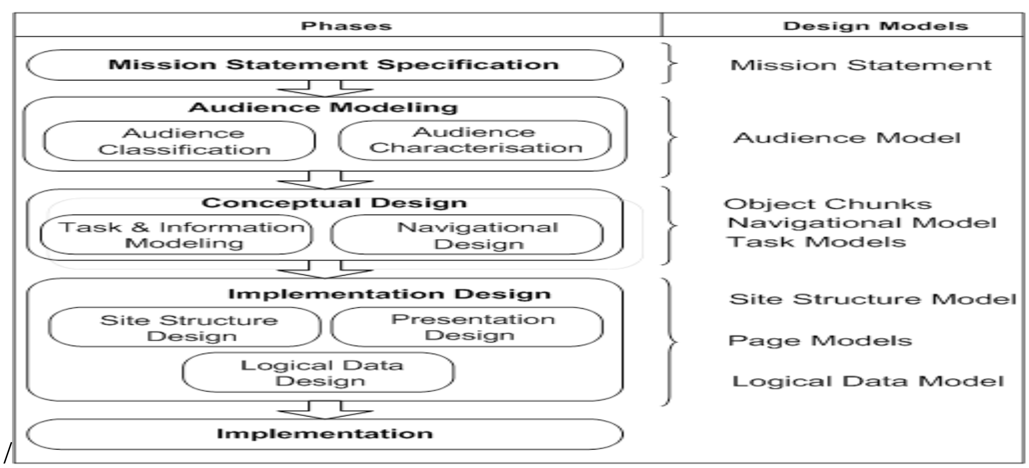

Figure 0: Main components of WSDM

Formal Model of WSDM in Z Specification Language:

In this formal model of WSDM, the formal model of only the Mission Statement, Initialization Schema and some operations is presented. The complete formal model is out of scope of this paper. In this paper, we have used Z, a formal modeling and specification language.

Here, only an abstract model of Web Semantic Design Methods is presented in Z There are two formal models in Z. The statics model define the basic types which user defined types to represent purpose, subject, task, target users, identity and user characteristics. The dynamic model defines operations and the results of those operations on the model. These operations define the behavior of the system. The following are the basic types for purpose, subject, task, target users, identity of users and user characteristics.

[*PURPOSE*, *SUBJECT*, *TASK*, *TARGETUSERS*, *IDENTITY*, *USERCHARACTERISTICS*]

$[PURPOSE, SUBJECT, TASK, IDENTITY, CHARACTERISTICS]$

MissionStatement

$movie\_show\_times : \mathbb{P}\ PURPOSE$
$movie\_tickets : \mathbb{P}\ PURPOSE$
$opinion : SUBJECT \rightarrow \mathbb{P}\ PURPOSE$
$movies : \mathbb{P}\ SUBJECT$
$actors : \mathbb{P}\ SUBJECT$
$directors : \mathbb{P}\ SUBJECT$
$producers : \mathbb{P}\ SUBJECT$
$cinemas : \mathbb{P}\ SUBJECT$
$games : \mathbb{P}\ SUBJECT$
$users : SUBJECT \rightarrow \mathbb{P}\ IDENTITY$
$movie\_lovers : \mathbb{P}\ SUBJECT$
$game\_lovers : \mathbb{P}\ SUBJECT$

$users \neq \emptyset$
$movie\_lovers \subseteq \mathrm{dom}\ users$
$game\_lovers \subseteq \mathrm{dom}\ users$
$\mathrm{ran}\ users \neq \emptyset$
$movies \neq \emptyset$
$\mathrm{dom}\ opinion \subseteq \mathrm{dom}\ users$

Figure 1. Mission Statement Schema

Add_Users

$\Delta MissionStatement$
$u? : SUBJECT$
$i? : \mathbb{P}\ IDENTITY$

$u? \in \mathrm{dom}\ users$
$users' = users \cup \{(u? \mapsto i?)\}$

Figure 2. Add users Schema

Add_Movies

$\Delta MissionStatement$
$m? : SUBJECT$

$m? \in movies$
$movies' = movies \cup \{m?\}$

Figure 3. Add Movie Schema

Add_Opinion

$\Delta MissionStatement$
$u?: SUBJECT$
$op?: \mathbb{P}\ PURPOSE$

$u? \in \mathrm{dom}\ users$
$op? \notin \mathrm{ran}\ opinion$
$opinion' = opinion \cup \{(u? \mapsto op?)\}$

Figure 4. Add Opinion schema

Remove_Movies

$\Delta MissionStatement$
$m?: SUBJECT$

$m? \in movies$
$movies' = movies \setminus \{m?\}$

Figure 5. Add Movies schema

Remove_Users

$\Delta MissionStatement$
$u?: SUBJECT$
$i?: \mathbb{P}\ IDENTITY$

$u? \in \mathrm{dom}\ users$
$users' = users \setminus \{(u? \mapsto i?)\}$

Figure 6. Remove Users Schema

Add_Games

$\Delta MissionStatement$
$g?: SUBJECT$

$g? \notin games$
$games' = games \cup \{g?\}$

Figure 7. Add Games Schema

Remove_Games

$\Delta MissionStatement$
$g?: SUBJECT$

$g? \in games$
$games' = games \setminus \{g?\}$

Figure 8. Remove Games Schema

Remove_Game_Lovers

$\Delta MissionStatement$
$u?: SUBJECT$

$u? \in \mathrm{dom}\ users$
$u? \in game\_lovers$
$game\_lovers' = game\_lovers \setminus \{u?\}$

Figure 9. Remove Game Lovers schema

Remove_Movie_Lovers

$\Delta MissionStatement$
$u?: SUBJECT$

$u? \in \mathrm{dom}\ users$
$u? \in movie\_lovers$
$movie\_lovers' = movie\_lovers \setminus \{u?\}$

Figure 10. Remove Movie Lovers schema

Remove_Game_Information

$\Delta MissionStatement$
$g?: SUBJECT$

$g? \in games$
$games' = games \setminus \{g?\}$

Figure 11. Remove Game Information schema

Remove_Movie_Information

$\Delta MissionStatement$
$a?: SUBJECT$
$d?: SUBJECT$
$p?: SUBJECT$
$c?: SUBJECT$
$s?: PURPOSE$
$t?: PURPOSE$

$a? \in actors \wedge d? \in directors \wedge p? \in producers \wedge c? \in cinemas$
$actors' = actors \setminus \{a?\}$
$directors' = directors \setminus \{d?\}$
$producers' = producers \setminus \{p?\}$
$cinemas' = cinemas \setminus \{c?\}$
$movie\_show\_times' = movie\_show\_times \setminus \{s?\}$
$movie\_tickets' = movie\_tickets \setminus \{t?\}$

Figure 12. Remove Movie Information schema

Add_Movie_Information

$\Delta MissionStatement$
$a?: SUBJECT$
$d?: SUBJECT$
$p?: SUBJECT$
$c?: SUBJECT$
$s?: PURPOSE$
$t?: PURPOSE$

$a? \in actors \wedge d? \in directors \wedge p? \in producers \wedge c? \in cinemas$
$actors' = actors \cup \{a?\}$
$directors' = directors \cup \{d?\}$
$producers' = producers \cup \{p?\}$
$cinemas' = cinemas \cup \{c?\}$
$movie\_show\_times' = movie\_show\_times \cup \{s?\}$
$movie\_tickets' = movie\_tickets \cup \{t?\}$

Figure 13. Add Movie Information schema

Add_Game_Lovers

$\Delta MissionStatement$
$u?: SUBJECT$

$u? \in \text{dom } users$
$game\_lovers' = game\_lovers \cup \{u?\}$

Figure 14. Add Game Lovers schema

Add_Game_Information

$\Delta MissionStatement$
$g?: SUBJECT$

$g? \notin games$
$games' = games \cup \{g?\}$

Figure 15. Remove Game Information schema

Add_Movie_Lovers

$\Delta MissionStatement$
$u?: SUBJECT$

$u? \in \text{dom } users$
$movie\_lovers' = movie\_lovers \cup \{u?\}$

Figure 16. Add Movie Lovers schema

Formal Analysis using Z/EVES

The formal analysis of the formal models developed in the previous section, were generated by using Z/EVES. In the specification window of the Z/EVES theorem prover, "Y" in the proof column shows that our specification is correct. Following are the snapshots taken from Z/EVES theorem prover for the Mission statement, audience classification, audience characterization, task modeling and information Modeling.

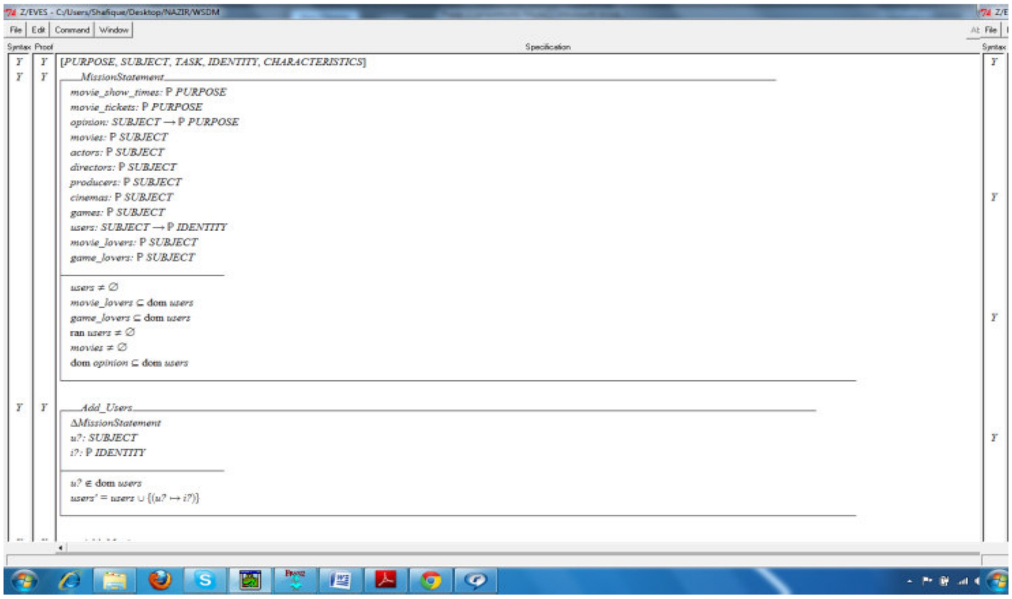

Figure 17. Proof of Mission Statement

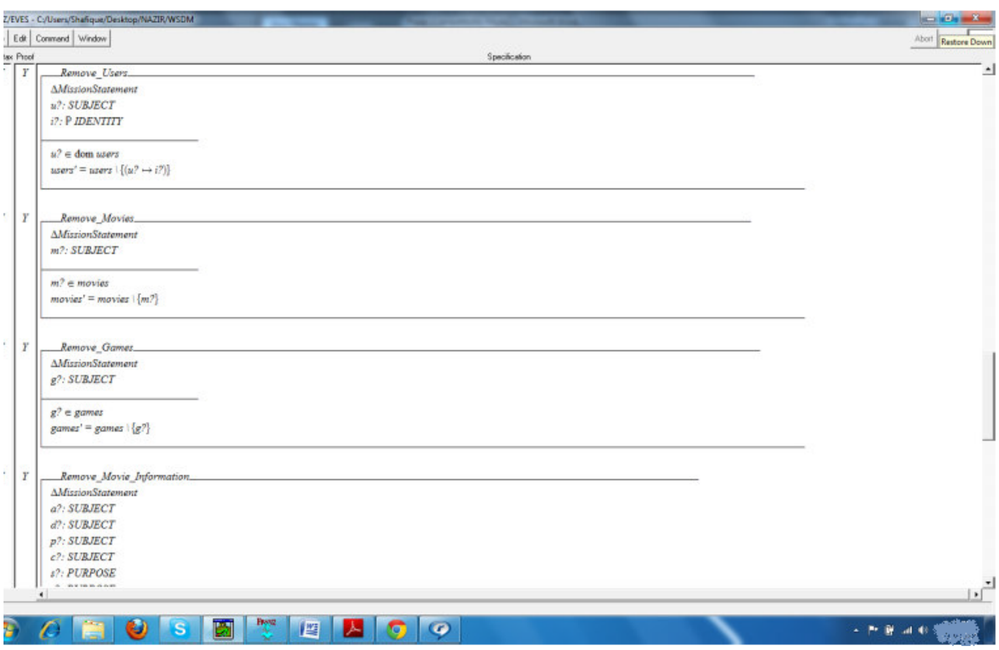

Figure 18. Proof of remove Users, Games, Movies, Movies information

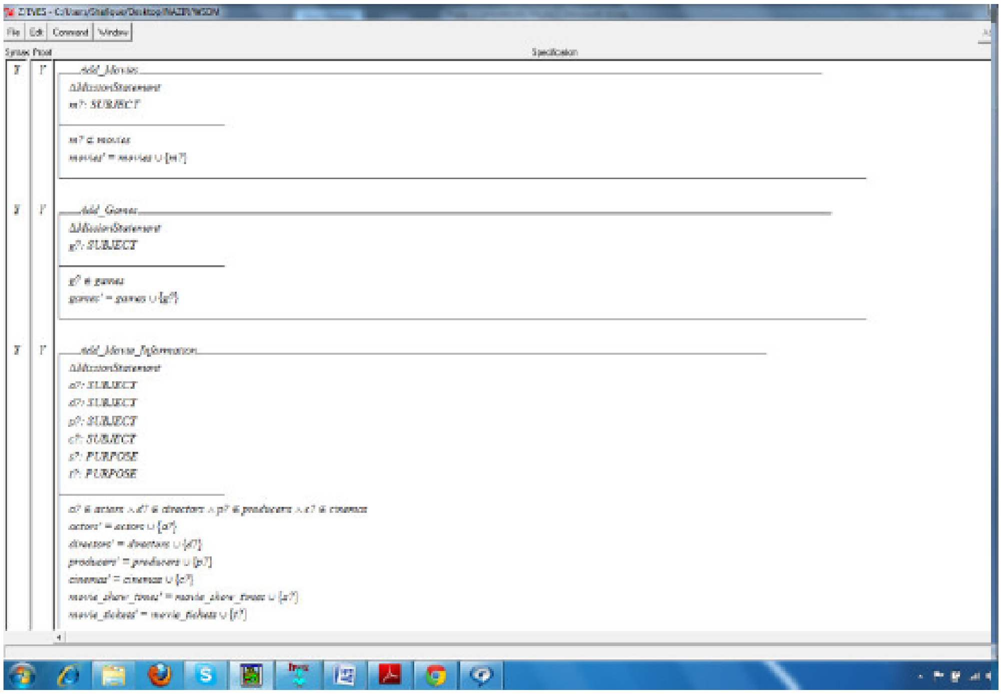

Figure 19. Add Movie, Games, Movies information

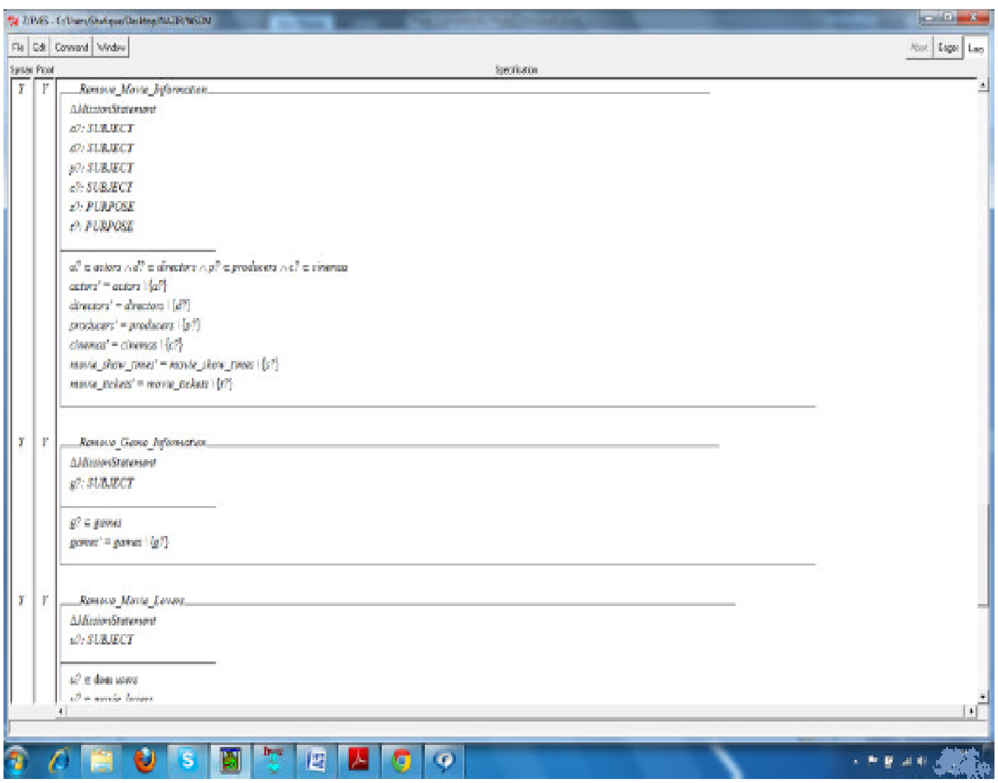

Figure 20. Remove Movie information, game info, Movie lovers

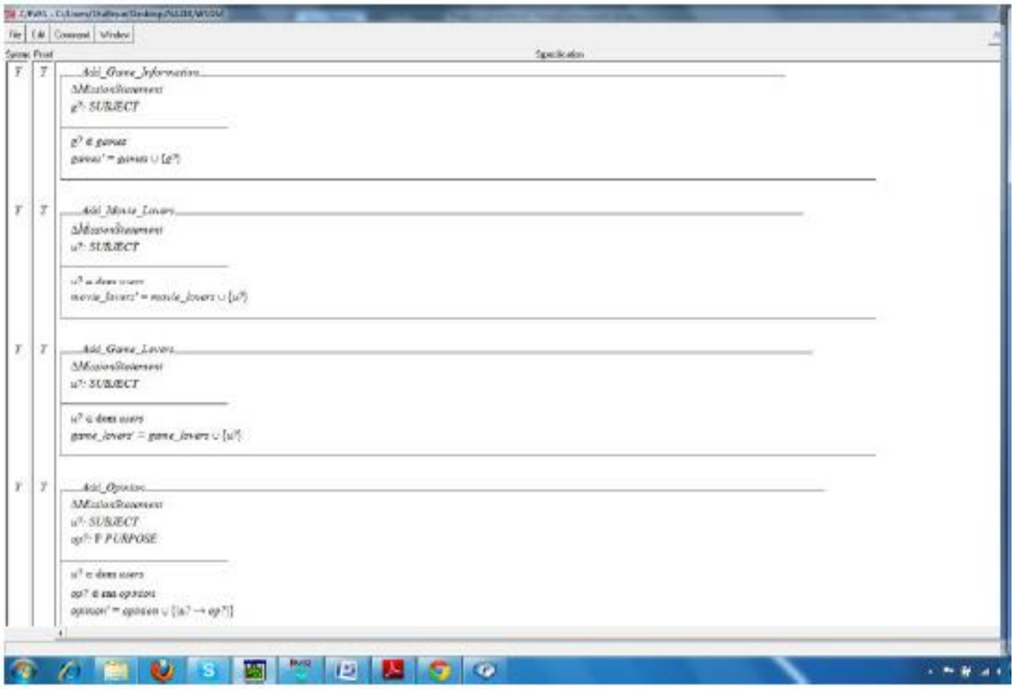

Figure 21. Add game information, Movie lovers, Games lovers, opinion

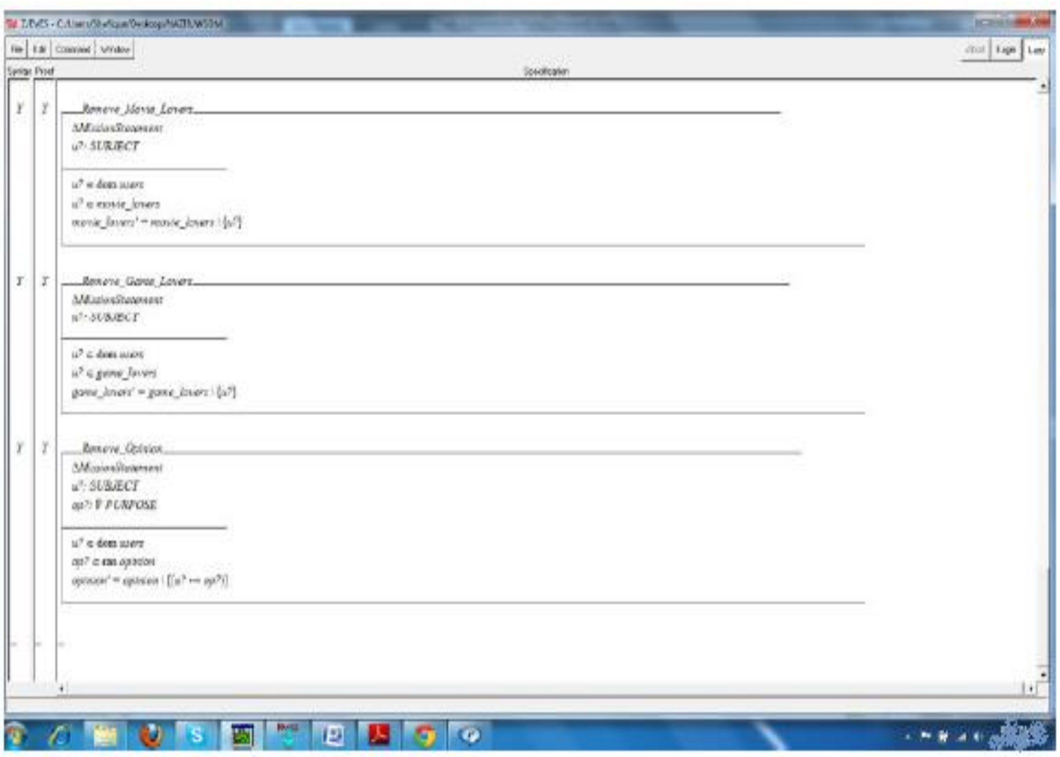

Figure 22. Remove Movie lovers, Games lovers, Opinion

Conclusion

In this paper, only few components such as Mission Statement, Audience Modeling and Conceptual Design are formalized in Z specification language. The formal models of these components are verified automatically by Z/EVES, a theorem prover based on EVES prover system. The Web Semantic Design Methodology is quite a large model and many components such as Navigational Paths, Logical Model etc are out of scope at this time. In the future, the whole approach will be formalized using a combination of formal methods such as Z, Colored Petri Nets, Graph theory etc. The output will be an automatic tool, after designing a suitable interface to automate the whole process. Also, work on some other important issues such as concurrency, deadlocks, race conditions; resource availability can be done in the future. Another important issue is the investigation of security properties such as authentication, authorization, confidentiality, integrity, non-repudiation and availability. This is also a future work.

References

Aine, C. et al., 2011. MEG-SIM: A Web Portal for Testing MEG Analysis Methods using Realistic Simulated and Empirical Data.Neuroinformatics, pp.1–18.

Arnold, S.E.J. et al., 2010. FReD: the floral reflectance database—a web portal for analyses of flower colour. PloS one, 5(12), p.e14287.

Bais, P. et al., 2010.PlantMetabolomics.org: A web portal for plant metabolomics experiments. Plant physiology, 152(4), pp.1807–1816.

Bajracharya, R.M. et al., 2010. The Integrated Web Portal (IWP). Mountain Research and Development, 30(2), pp.178–181.

Bowen, J.P. & Hinchey, M.G., 1995a. Seven more myths of formal methods. Software, IEEE, 12(4), pp.34–41.

Bowen, J.P. & Hinchey, M.G., 1995b. Ten commandments of formal methods. Computer, 28(4), pp.56–63.

Christensen, H. et al., 2010. Beacon: a web portal to high-quality mental health websites for use by health professionals and the public. Medical Journal of Australia, 192, pp.40–44.

Clarke, E.M. & Wing, J.M., 1996. Formal methods: State of the art and future directions. ACM Computing Surveys (CSUR), 28(4), pp.626–643.Communications of the ACM, Vol. 45, No. 2, pp. 61–65.

Ding, Y. et al., 2010. Semantic web portal: a platform for better browsing and visualizing semantic data. Active Media Technology, pp.448–460.

Gómez-Pérez, A.; Manzano-Macho, D. (2004) An overview of methods and tools for ontology learning from texts. The Knowledge Engineering Review. 19(3), pp. 187–212.

Hall, A., 1990. Seven myths of formal methods. Software, IEEE, 7(5), pp.11–19.

Haxthausen, A.E., 2010. An Introduction to Formal Methods for the Development of Safety-critical Applications,

Jackson, D., 2001. Lightweight formal methods. FME 2001: Formal Methods for Increasing Software Productivity, pp.1–1.

Jacky, J., 1997. The way of Z: practical programming with formal methods, Cambridge Univ Pr.

Jensen, J., & Dohn, N. B. (2016). Challenges and possibilities for Design Based Research with semantic web technology. In Proceedings from 10th International Conference on Networked Learning, Lancaster.

Kwak, J.H. et al., 2011. HERO Web Portal: On-demand portal for co-allocating computing and network resources on top of TIGRIS middleware platform. In 2011 IEEE Ninth International Conference on dependable, Autonomic and Secure Computing. pp. 796–797.

Lendyuk, T. &Rippa, S., 2011a. Information portal of E-learning system in Semantic Web environment. In Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on. pp. 637–641.

Meisels, I. & Saaltink, M., 1997. The Z/EVES reference manual (for version 1.5). Reference manual, ORA Canada.

Pommier, A. & Le-Trong, E., 2011. 'SIGMELTS': A web portal for electrical conductivity calculations in geosciences. Computers and Geosciences, 37(9), pp.1450–1459.

Preiser-Houy, L. &Navarrete, C.J., 2010. Benefits and Challenges of Using Community-Based Research to Develop an Educational Web Portal.

Ringwald, M. et al., 2011. The IKMC web portal: a central point of entry to data and resources from the International Knockout Mouse Consortium. Nucleic acids research, 39(suppl 1), p.D849.

Saaltink, M., 1997a. The z/eves system. ZUM'97: The Z Formal Specification Notation, pp.72–85.

Saaltink, M., 1997b. The Z/EVES user's guide.

Vij, A., Bhattacharyya, S. & Ramaprasad, A., 2012. Collaborative Ontological Design of a Web Portal. In 2012 45th Hawaii International Conference on System Sciences. pp. 336–342.

Whetzel, P.L. et al., 2011. BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. Nucleic Acids Research, 39(suppl 2), p.W541.

Woodcock, J. & Davies, J., 1996. Using Z: specification, refinement, and proof, Prentice Hall.

Woodcock, J. et al., 2009. Formal methods: Practice and experience. ACM Computing Surveys (CSUR), 41(4), pp.1–36.