Semantic Web Specification using Z-Notation

Sher Afzal Khan[1], Aamir Aziz Hashmi [2], Fahad Alhumaidan[3] and Nazir Ahmad Zafar[3]

[1]Department of Computer Sciences, Abdul Wali Khan University, Mardan, Pakistan
[2]Department of Networking, Virtual University of Pakistan, Islamabad, Pakistan
[3]Department of Computer Science, King Faisal University, Hofuf, Saudi Arabia
sher.afzal@awkum.edu.pk; ana1-isb@vu.edu.pk; {falhumaidan ,nazafar}@kfu.edu.sa

Abstract: Current World Wide Web means to display pages to end user, while the Semantic Web is a vision of a next-generation network focuses on "Meaning" instead of merely pasting arbitrary text on a page. An intelligent software agents use information to organize and filter data to meet the user's needs. DAML+OIL and Web Ontology Language OWL are the current environments to create Ontology over RDF and XML structures which are used to represent data intelligently among different Ontologies. To assure quality and accurateness in Ontologies in the early design stage, we used the Z-specification which is a formal language based on discrete mathematics such as predicate logic, sets, relations and functions to specify the behavior of Semantic Web. Further, we applied a transformation from schemas written in Z-specification to OWL. The formal specification is described and validated using Z/EVES tool. A fundamental goal of this research is to transform a verified and validated specification to OWL to design Ontologies.
[Khan SA, Hashmi AA, Alhumaidan F, Zafar NA. **Semantic Web Specification using Z-Notation.** *Life Sci J* 2012;9(4):994-1000] (ISSN:1097-8135). http://www.lifesciencesite.com. 153

Keywords: Semantic Web; Ontology; Formal Methods; Z-specification

1. Introduction

The Semantic Web (Berners, et al, 2003) is an intelligent extension of current World Wide Web. It describes Web services in such a way that computers or intelligent software agents can understand the meaning of Web pages. It is still in its evolving period, challenges are still there to meet the goals. Semantic Web's developers are looking to XML and RDF to meet these challenges. DAML+OIL (Harmelen et al., 2001) and OWL (Dean et al., 2004) are the current environments over the top of XML and RDF. They are playing remarkable role in designing Ontologies and interacting among different Ontologies over the Internet. These languages are based on descriptive logic and they are designed to be decidable (World Wide Web, 2003). The Z-notation (Khan et al., 2008, 2009, 2011) is a formal specification language based on descriptive logic, sets, relations and functions. Descriptive logic can be regarded as the subset of predicate logic. Therefore, Z is more expressive than other Ontology languages. Z/EVES (Meisels, 1997) is a proof tool for reasoning and checking Z-specifications. In this paper we use Z-specification to specify and verify the requirements of the Semantic Web. In the idea we first specify system in Z-specification then check its proof and syntax by Z/EVES tool. Further convert Z-model into OWL to design Ontology. The use of Z-specification and Z/EVES tool removes inconsistencies and ambiguities in Ontology. The transformation between Z-specification and OWL can confine the properties of Ontology that the OWL can not. As (Dong et al., 2004) describes that the intrinsic homogeneity between semantic bases of ontology languages and Z implies that Z can be regarded as an ontology meta-language and it can even capture properties that ontology languages cannot. Further, (Dong et al., 2002) expresses that Z-specification can capture various requirements of Semantic Web services including ontology and service functionalities. The research of using the transformation of Z-specification and Semantic Web has already been used. As (Dong et al., 2004(a)) use Z-specification on DAML+OIL to design and reuse ontology. As a forward approach proposed by J.S. Dong et al. (2002), they describe that the use of Z semantics to design Ontology would be easy to reduce Ontology flaws. In his technique they simply introduced a transformation of Z model into DAML+OIL to design Ontology and also provide some rules of transformation. The idea was simple but he left a reverse technique that is to transform the DAML+OIL to Z-specification for their future work. Further J.S Dong and his team propose the reverse technique in which they use Racer with Z-specification and introduced that DAML+OIL can be re transformed into Z Model to check inconsistencies of Ontologies. The idea became complete here with forward and reverse transformations together to remove Ontology related flaws by the description logic. Dong et al. (2002) describe the combined approach of DAML+OIL, Z-specification language, RACER and Alloy, in the research they first apply Z-specification to design Ontology, next RACER is used to identify any inconsistency in Ontology, after that Alloy is used to trace origin of an error(s) and lastly Z-specification is used to express complex Ontology properties. The only problem seen in this approach is a long procedure. Whereas, this can be possible more efficiently with the integration of Ontology Web Language OWL and Z-specification. The OWL has more and extra efficient features over other ontology's based languages, W3C recently introduced OWL for Semantic Web which is on top of XML/RDF.

In the paper we use OWL as Ontology language. The language OWL is derived from DAML+OIL which does not include qualified number of restrictions. The OWL can define symmetric properties and does not rename RDF-S primitives which increase the power of Semantic Web. The mentioned importance forced us to map Z-specifications to OWL with the motivation from basic idea described by (Dong et al., 2002, 2004).

The rest of the paper is organized as follows. In section 2, an introduction to formal method is given. In section 3 we describe an overview of Semantic Web, OWL, DAML+OIL, XML and RDF and further describe the conceptual model for Z approach to OWL. Section 4 consists of implementation and verification of model with Z/EVES tools and section 5 contains the conclusion of the study and a future work

2. Formal Methods

Formal methods are based on mathematical techniques and notations uses for describing and analyzing properties of software systems (Clarke et al., 1996; Khan et al., 2011(a); Zafar et al., 2012; Ahmad et al 2012; Ali et al., 2012). These mathematical techniques are based on discrete mathematics such as predicate logic, set theory, relations, functions, and graph theory. The process, to develop software systems using formal methods is shown in the Figure 1 (Liu et al.,1995).

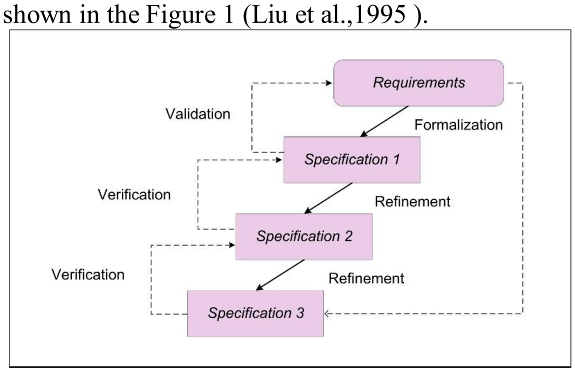

Figure 1. The Process of Software Development using Formal Methods

The "Requirements" are the result of requirements analysis and are normally described in informal language. 'Specification1' represents the stage of transformation from requirements to formal methods. Further, the process from 'Specification 2' to 'Specification n' corresponds to the stage of design. The process from 'Specification n' to 'Program' corresponds to the stage of implementation or coding. Validation and verification are the two basic principles that arise in system development. Validation addresses whether the produced system fulfills the requirements and verification check whether the software meets the requirements established in the previous phases. The aim of this approach is to demonstrate the process of development of a system from requirement to coding using formal specification. The use of this approach identifies errors and oversights early in the design life-cycle which are then easy to remove, with consequent high quality and cost saving software development. The Z-notation (woodcock et al., 1996), is model-oriented approach, it is used for specifying the behavior of abstract data type and sequential programs. The Z-specification divides the specification of complex system in different states called schemas. The schema consists of three parts; schema name, schema signature and schema predicate. These schemas can be combined to produce the overall description of the system. The paper addresses schemas in the specification part of the paper. Z-specification cannot typically be executed by computers, but the standard tools are available which are used for checking syntax and proof of the specification, leads to quality of specification and this allows mistakes to be detected and corrected sooner in the design life cycle.

3. The Semantic Web

Semantic Web is a future of current Web in which information is given with well-defined meaning in such a way that computers or intelligent software agents can understand the meaning of Web pages (Berners et al., 2002. Allwood et al., (2008) describe that the Semantic Web is an evolving extension of the World Wide Web in which Web content can be expressed not only in natural language, but also in a form that can be understand, interpret and use by software agents, thus permitting them to find, share and integrate information more easily. Semantic Web is a combination of different Ontology over Internet in such a manner that they can understand, interpret data intelligently without human involvement. Where ontology is a data model represents a set of concepts within a domain and the relationships between those concepts (Yang et al, 2008). Ontologies are represented by Ontology languages such as DAML+OIL and OWL, which are based on top of XML and RDF. XML is a set of rules for defining and representing information as structured documents for applications on the Internet. RDF is a model for describing Web resources. Differentiating from HTML, HTML is aimed to deliver data to end user while XML is an extensible language: a language to describe other languages. XML is focused on syntax of the document rather than text. RDF defines resources on the Internet and provides interoperability between applications to exchange data. So that RDF uses XML to exchange description of Web resources. RDF Schema provides the built in vocabularies for RDF library. It is used to define properties of Web resources. DAML is a semantic markup language based on XML/RDF for Web services. DAML combined with Ontology-Interface Layer is referred as DAML+OIL. By using existing classes and properties, new concept can be added. This enables the DAML+OIL to reuse the existing technology. In 2003, W3C proposed a new markup language for Semantic Web known as OWL. It is based on top of DAML+OIL. Main differences are, OWL does not include qualified number restrictions, further, it can define symmetric properties and does not rename RDF-S primitives. In other words, the power of Semantic Web is increased with OWL. It has three flavors: Lite, DL and Full with enhanced capabilities used according to demand.

The semantic Web and Z-Specification

While communicating over the Web, these Ontologies need to be proper functioning. If Ontologies are not properly defined then obviously wrong results will cause problems. Z-specification is a descriptive logic which can perform well for Semantic Web. We are providing the power of Z-specification to built Ontology to meet the challenges in designing, testing and verification stages.

3 Z-Specification towards OWL

In the paper initially we specify the system using Z-specification. Further check their proof and syntax by using Z/EVES. After verifying we transform Z-model (encoded in ZML (Sun et al., 2001) into OWL to design Ontology and after checking and testing, retransformation of OWL into Z-model (again encoded in ZML) to remove inconsistencies in Ontology. The above process is shown in Figure 2.

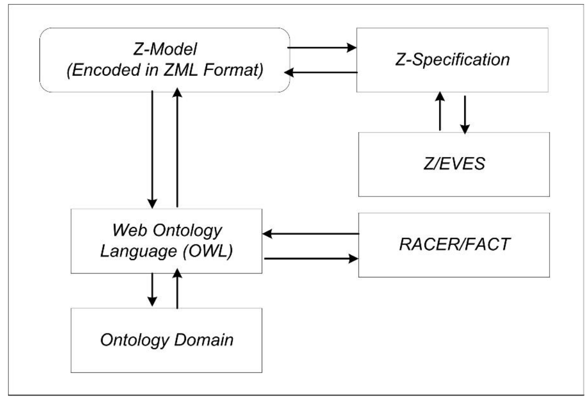

Figure 2. The Process of Z Approach to OWL

Transformation of Z-Specification to OWL

In order to transform Z semantics in OWL, we go through an example of IT talk discovery system TDS. The TDS is an on line Web portal offering the service of information about seminars. It also offers IT related upcoming talks that may appeal a register user according to his personal interests or schedules. It can be categorize in four agents.

- User's Calendar Agent
- Distance Map Agent
- User's Personal Agent
- Talk Discovery Agent

All these agents work collaboratively on behalf of human to extract the information about user's travel guide. In the next subsections we describe Z-specification in Z/EVES and then there corresponding OWL semantics to highlight the idea of transformation.

Z Model for IT Discovery System and OWL code

User's personal agent needs to consult with User's Calendar Agent to determine whether user is available or not. The calendar agent in IT Talk Discovery System can be defined in Z-specification as:

[TIME, DATE]
Where TIME and DTAE are of set types. The date and time can be defined as the schema DateTime.

Date_Time
d: DATE
t: TIME

The Status defined by Z free type definition shows that either user is busy or not.
Status ::= Free/Busy
The Calendar schema is as:

Calender
Schedule: Date_Time $\rightarrow$ *Status*

Transformation of Date and Time type definition into OWL can be written as:

```
<?xml version="1.0"?>
<rdf:Description rdf:about="d:DATE">
                <rdf:type>
                        <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#Class"/>
                </rdf:type>
        </rdf:Description>
        <rdf:Description rdf:about="t:TIME">
                <rdf:type>
                        <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#Class"/>
                </rdf:type>
        </rdf:Description>
</rdf:RDF>
```

Note: Code is simplified. Name spaces are omitted.

Distance Map Agent

After checking the availability of user, the user's personal agent needs to determine the distance between user's office and talk place. The Distance Map Agent outputs the distance for the user's personal agent.

We define [PLACE] is of a set type.

Map
$places : \mathbb{P}\ PLACE$
$dist. : places \times places \rightarrow \mathbb{R}^+$

The OWL code can be written as:

```
<rdf:Description rdf:about="d:dist">
                <rdf:type>
                        <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectP
roperty"/>
                </rdf:type>
                <rdfs:domain>
                        <rdf:Description
rdf:about="place:places"/>
                </rdfs:domain>
                <rdfs:range>
                        <rdf:Description
rdf:about="r:R"/>
                </rdfs:range>
                <rdfs:domain>
                        <rdf:Description
rdf:about="place:places"/>
                </rdfs:domain>
        </rdf:Description>
        <rdf:Description rdf:about="r:R">
                <rdf:type>
                        <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#Class"/>
                </rdf:type>
        </rdf:Description>
```

Note: For simplicity, Namespaces and classes are omitted.

User's Personal Agent

User's personal agent keeps user's personal information i.e. user's profile, user's office location, interests etc.

Z specification can be written as:

[NAME, TOPICS]

where NAME and TOPICS are of set type.

Personal
Name: NAME
Office: PLACE
Interest: $\mathbb{P}$ *TOPICS*

The OWL code can be written as:

```
<rdf:Description rdf:about="nam:NAME">
                <rdf:type>
                        <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#Class"/>
                </rdf:type>
        </rdf:Description>
        <rdf:Description
rdf:about="office:PLACE">
                <rdf:type>
                        <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#Class"/>
                </rdf:type>
        </rdf:Description>
        <rdf:Description
rdf:about="interest:TOPICS">
                <rdf:type>
                        <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#Class"/>
                </rdf:type>
        </rdf:Description>
        <rdf:Description rdf:about="int:Interest">
                <rdf:type>
                        <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#Class"/>
                </rdf:type>
                <rdfs:subClassOf>
                        <rdf:Description
rdf:about="interest:TOPICS"/>
                </rdfs:subClassOf>
        </rdf:Description>
```

Talk Discovery Agent

Finally, Talk discovery agent outputs various results based on users interests. The Talk schema is defined for a general talk type. Interested talks are recorded in interestedtalk for the user.

Talk
place: PLACE
dt: Date_Time
subject: ℙ *SUBJECT*

The OWL code for Talk schema can be written as in user's personal agent
<rdf:Description rdf:about="Int:Interested_Talk">
<rdf:type>
<rdf:Description rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</rdf:type>
<rdfs:domain>
<rdf:Description rdf:about="p:Personal"/>
</rdfs:domain>
<rdfs:range>
<rdf:Description rdf:about="T:Talk"/>
</rdfs:range>
</rdf:Description>

Implementation and verification

We checked the schemas's proof and syntax by the tool Z/EVES, the snapshot of the proof is shown in Figure 3.

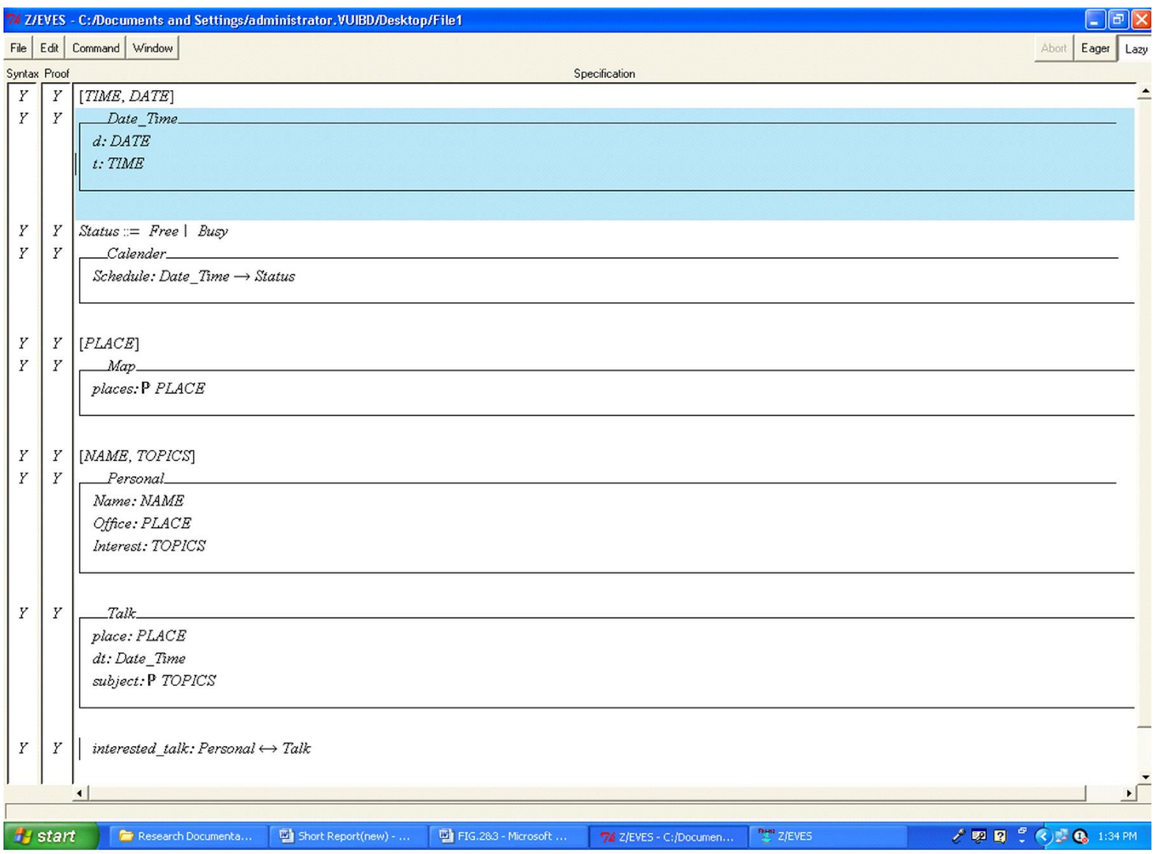

Figure 3. Z/EVES Schemas for IT Talk Discovery

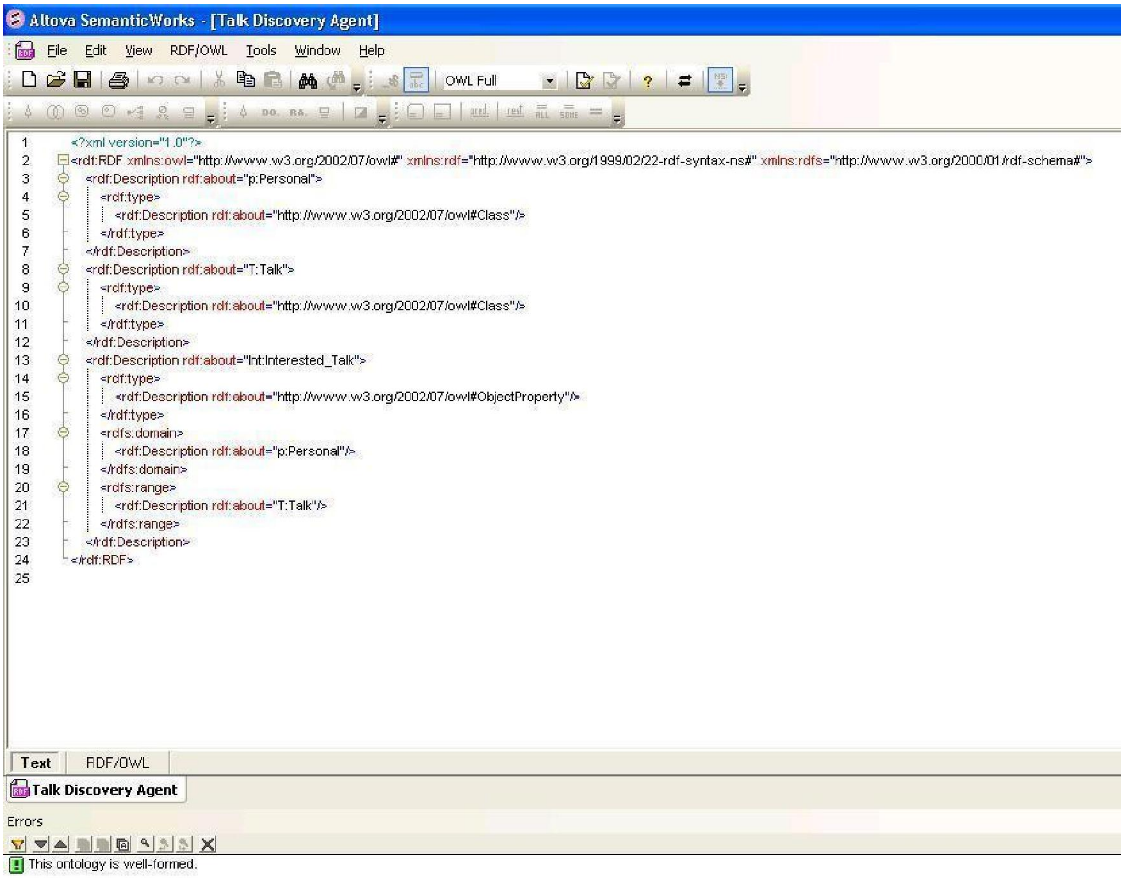

Figure 4. OWL code for IT Talk Discovery

Further we transform schemas to their corresponding OWL code in Altova Semantic Works 2008 which is shown in Figure 4.

Conclusion

In this paper, we have discussed an idea of designing Ontology, using Z-specification schemas. The verification is done by Z/EVES and further transformation is done into OWL code. To observe the approach, we have used an example of IT Talk Discovery System which is an on line Web portal offering the service of information about seminars and IT related upcoming talks that may appeal a register user according to his personal interests or schedules. We have shown, via an example, a one to one correspondence between Z-schemas and OWL code which provides an environment to transform Z-schemas to OWL after removing an inconsistencies and ambiguities from a specification.

References

1. Allwood, J.: Some remarks on the relationship between the semantic and the pragmatic Web. In: Proceedings of the 3rd international Conference on the Pragmatic Web: innovating the interactive Society, (2008)
2. Berners, L.T., Miller, E.: The semantic Web lifts off. Ercim News Online Edition, No. 51, (2002)
3. Berners, L.T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, (2001)
4. Clarke, E., Wing, J.: Formal methods: State of the art and future directions. ACM Computing Surveys 28, 4, pp. 626-646 (1996)
5. Dean, M., Schreiber, G. (editors).: OWL Web Ontology Language Reference. W3C Recommendation, http://www.w3.org/TR/owl-ref/, (2004)
6. Dong, J.S., Lee, C.H., Li, Y.F., Wang, H.: Verifying DAML+OIL and Beyond in Z/EVES. In: Proceedings of the 26th International Conference on Software Engineering. pp. 201-210 (2004)
7. Dong, J.S., Sun, J., Wang. H.: Z Approach to Semantic Web. In: International Conference on Formal Engineering Methods. LNCS, Springer-Verlag, pp. 156-167 (2002)
8. Dong, J.S., Lee, C.H., Li, Y.F., Wang, H.: A Combined Approach to Checking Web Ontologies. In: Proceedings of 13th World Wide

Web Conference, New York, USA, pp. 714-722 (2004(a))
9. Harmelen, F.V., Peter, F.P.S., Horrocks, I. (editors).: Reference description of the daml+oil ontology markup language. http://www.daml.org/2001/03/reference.html, (2001).
10. Khan, S.A and Zafar, N.A, Promotion of Local to Global Operation in Train Control System, Journal of Digital Information Management (page 228-233).,2007.
11. Khan, S.A., Zafar, N.A.: Towards the formalization of railway interlocking system using z-notations. In: Proceedings of the 2nd IEEE International Conference on Computer, Control and Communication, pages 750-755 (2009)
12. Liu, S., Adams, R.: Limitations of Formal Methods and an Approach to Implement. Technical Report ,Hiroshima City University, (1995)
13. Meisels, I. and Saaltink, M.: The Z/EVES reference manual. TR-975493-03, ORA, Canada, (1997)
14. Sun, J. Dong, J.S. Liu, J. and Wang, H. Object-Z Web Environment and Projections to UML. In: 10th International World Wide Web Conference, pp. 725-734 (2001)
15. World Wide Web Consortium (W3C). Web Ontology Language (OWL) Use Cases and Requirements. http://www.w3.org/TR/webont-req/, (2003)
16. Woodcock, J. and Davies, J.: Using Z: Specification, Refinement, and Proof. Prentice-Hall International, (1996)
17. Yang, H. and Callan, J.: Human-Guided Ontology Learning. Second Workshop on Human-Computer Interaction and Information Retrieval. Microsoft Research, Redmond, WA, USA (2008)
18. Khan, S.A and Zafar, N.A, Improving Moving Block Railway System using Fuzzy Multi-Agent Specification Language, Int. J. Innov. Computing, Inform. Control, 7(7(B)):4517-34, 2011.
19. Khan, S.A, Zafar, N.A and Ahmad, F, Petri Net Modeling of Railway Crossing System using Fuzzy Brakes, International J. Phy. Sci, 6(14): 3389-3397, 2011(a).
20. Zafar, N.A, Khan, S.A and Araki, K, Towards the Safety Properties of Moving Block Railway Interlocking System, Int. J. Innov. Computing, Inform. Control, 8(8): 2012 .
21. Khan, S.A, Zafar, N.A, Extending promotion for the management of moving block interlocking components., International J. Phy. Sci, 6(31), 7262-70. 2011(b).
22. Ahmad, F. Khan, S.A, Module-based Architecture for Periodic job-Shop Scheduling problem, Computers & Mathematics with Applications, 64(1), 1-10, 2012.
23. Ali, G., Khan, S.A., Ahmad, M.F. and Zafar, N.A, Visualized and Abstract Formal Modeling towards the Multi-Agent Systems, International Journal of basic and Applied Sciences 2(8)8272-8284, 2012
24. Ali, G., Khan, S.A., Ahmad, M.F. and Zafar, N.A, Formal Modeling towards a Dynamic Organization of Multi-Agent Systems Using Communicating X-Machine and Z-Notation, Indian Journal of Science and Technology, Vol. 5 No. 7, 2012(a)

9/23/2012