Software Architectural Design Ontology

Muhammad Irfan Marwat, Sadaqat Jan, Syed Zafar Ali Shah

Abstract—Software Architecture plays a key role in software development but absence of formal description of Software Architecture causes different impede in software development. To cope with these difficulties, ontology has been used as artifact. This paper proposes ontology for Software Architectural design based on IEEE model for architecture description and Kruchten 4+1 model for viewpoints classification. For categorization of style and views, ISO/IEC 42010 has been used. Corpus method has been used to evaluate ontology. The main aim of the proposed ontology is to classify and locate Software Architectural design information.

I. INTRODUCTION

THE impressive increase in the software development during two decades has resulted in an incredible amount of information. The Internet is providing a large platform for handling this information but we are still far away from easy access. Unluckily, the information stored was designed for human-machine communication instead of machine-machine communication. The main reason of the problem is the free text nature of web information. To make the information machine readable and process-able, semantic web was first introduced by Berners-Lee, OraLassila and James Hendler. Semantic web provides meaning to the content of web pages for making it machine readable. Ontology is used to form information in such a way that it can be easily processed by machine and human. In the philosophy, the ontology is called the "Theory for kind of existence or nature of beings". The ontology was first developed by Greek philosophers Socrates and Aristotle. Socrates, for the first time, introduced the abstract idea of ontology by defining hierarchy and relationship between classes and instances.

The model was further extended with logical axiom for reasoning to form inference model. The term ontology has been adopted by computer science for their own needs. Ontology has been used to cross the barrier of machine not recognizing the meaning of the knowledge and to enable the user to search required knowledge based on meaning instead of syntax. A key problem is defining the standards to represent underlying structure. Different approaches have been adopted by different researchers to design, represent and construct ontologies. In our work we have fallowed the methontology

Muhammad Irfan Marwat is student of UET Peshawar Mardan Campus KPK Pakistan (phone: 0092-333-9988906 e-mail: outspoken666@gmail.com).

Sadaqat Janis is Assistant Professor and Head of Software Engineering Department in UET Peshawar Mardan Campus KPK Pakistan (e-mail: Sadaqat_jan@hotmail.com).

Syed Zafar Ali Shah is Lecturer in UET Peshawar Mardan Campus KPK Pakistan (e-mail: iamzafarali@gmail.com).

[1] methodology to create ontology. Designing and specifying an overall large and complex software system introduces new problems and research issues.

In this paper, we have proposed ontology for software architecture design based on IEEE model for software architecture and ISO/IEC 42010 for style and views categorization. This work will help to categorized and allocate information about software architecture online and in desktop systems.

The rest of this paper is structured as follows. Section II Provides related work. In Sections III and IV ISO/IEC 42010 (in short "Eye-so-forty-two-ten") and 4+1 model have been explained, respectively. Software architecture ontology is developed and presented in Section V. Section VI concludes the paper followed by references.

II. RELATED WORK

Different formal techniques have been applied to software architecture to achieve precise specification and rigorous verification of software architecture design. Kim and Garlan used Alloy language to verify and model architecture style. In their work, they have verified and translated few architecture styles based on ACME using Alloy analyzer [2]. Wong et al. [3] proposed a model called splitting approach for parallel verification and translation of the Alloy based architecture. Sun, Wang and Hu proposed ontology to support architecture verification, style recognition, check consistency, behavioral inference and ensure correctness [4]. Kampffmeyer et al. [5] proposed a design pattern intent ontology (DPIO) to efficiently retrieve software design patter during design stage of software development. Harb Bouhours and Leblance [6] proposed to extend DPIO with alternative model and strong point concepts. The purpose of this work is to help designer in integration of design pattern. Hois Bhatt and Kutzprposed a modular ontology for Architectural design based on the theory of ε-connection to formally bring together these different perspectives [7]. Philippe Kruchten proposed ontology of architectural design decision for complex software intensive system. The aim of the proposed ontology is to enable construction of complex graph and support reasoning about them [8]. Inostroza and Astudillo proposed an approach for abstraction based component characterization using semantic web technology. Ontology has been used to assert complex or more specific relations between high level architecture notions [9]. Henninger and Ashokkumar proposed ontology based Metamodel for developing patterns language. This work helps to build intelligent tool to select best selection for recurring problems [10]. Dietrich and Elgar proposed ontology to formally define design patterns and related concepts. This work helps in defining patterns and sharing of knowledge about

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:7, No:12, 2013

patterns [11]. Emdanat and Vakalo proposed ontology conceptual design in architecture based on shape algebra. The ontology is intended to facilitate knowledge sharing of shapes and their properties [12]. There are several proposed ontologies in software Architectural design area. However, no specificontology to classify information of software Architectural design for learning purpose has yet been explored in the recent research work.

III. SOFTWARE ARCHITECTURE AND USE OF ONTOLOGY

Software architecture is an important sub-discipline of software engineering. Architecture is the prudent partitioning of a whole structure into small parts, having specific relations among the parts. This partitioning helps in global development of software. Each group of different locations tries to write software in such a way that it is easy to integrate the work done by different developers. However, for different environment people, it is very difficult to communicate and work cooperatively because of different software engineering methodology and different terminology used. Ontology can be used to provide a common understanding for different terminologies and make the knowledge available for future use.

IV. ISO/IEC 42010 (IN SHORT "EYE-SO-FORTY-TWO-TEN")

It is the ISO standard for system/software engineering architecture description, published in 2007 for the first time. Eye-so-forty-two-ten is the fast track adoption of Std 1471-2000, developed by an IEEE working group. In view and beyond approach of 42010, views play a central role in documenting software architecture. According to ISO 42010 architecture; the description should contain fallowing:

- Identification of stakeholders.
- Identification of architecture related concerns.
- Architecture Viewpoints.
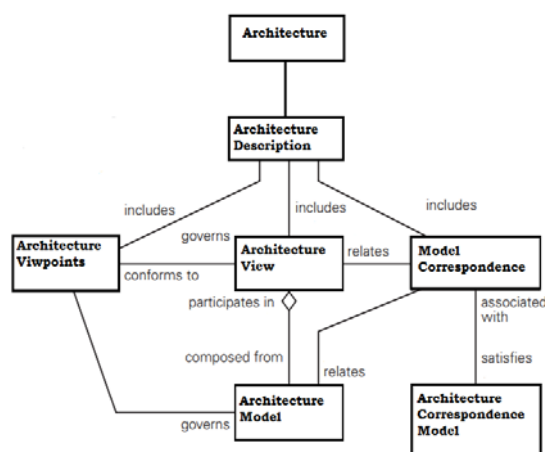- Architecture Models.
- Architecture rationale.

Fig. 1 IEEE model for architecture description

The View can be further categorized into styles: module style, component and connecter style, allocation style and Hybrid style. These four styles can be further divided into different classes with respect to their structure as shown by Paul Clement et al. in the Fig. 2.
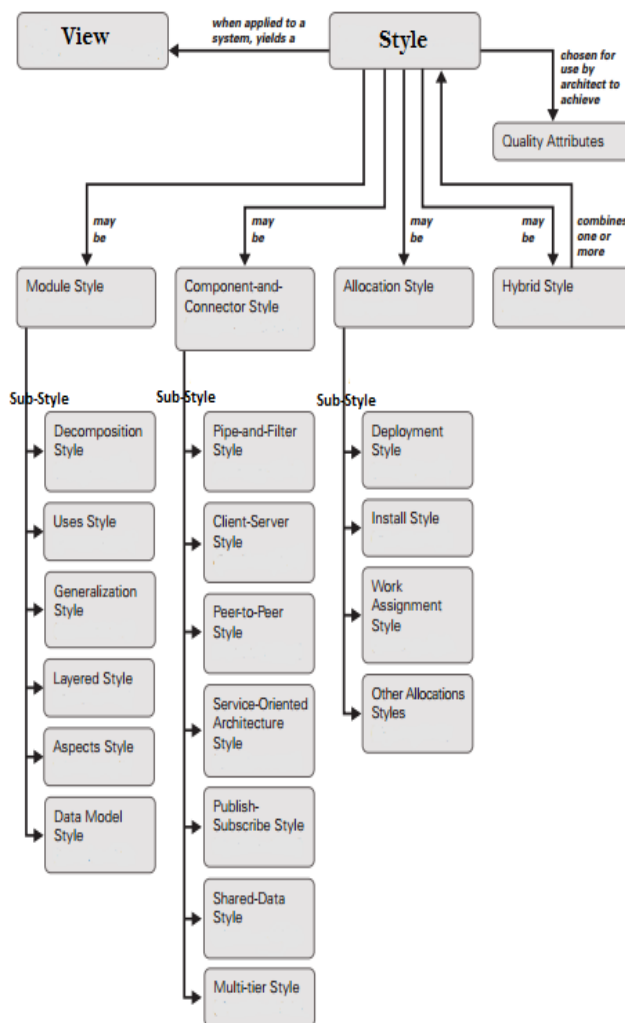

Fig. 2 Views and Beyond

V. 4+1 VIEW MODEL

The 4+1 view model is designed by Philippe Kruchten to organize description of intensive software system using five concurrent views addressing a specific set of concerns. The four views, logical, implementation, process and deployment are used to capture design decision and fifth view scenario is used to validate and illustrate them.

A. Logical View

It is concerned with the functionality providing system to the end user, And to describe the design' object model when an OOD (object oriented design) method is used. Rational/Booch approach has been used to represent the logical view through class diagram and templates [24].

B. Process View

It is concerned with the integrator, performance, scalability, and concurrency etc. of the system, or in other words its

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:7, No:12, 2013

describe the synchronization and concurrency aspect of the design. In process view we take into account some non-functional properties such that performance, availability etc. To create the process view the designers divide the software into partition.

C. Development View

This view is concerned with the description of the system with respect to programmers and manager's point of view, or development view is used to describe the static organization of software in its development environment. In this view we take into account the internal requirement related to the ease of the development [24].

D. Physical View

It is concerned with the description of system with respect to system engineer point of view or this view deal with the mapping of the software onto hardware. In this view we take into account the system non-functional requirement's throughput, fault tolerance, availability and reliability etc.

E. Scenario View

This view describe as a sequence of interaction b/w processes and b/w objects, also called as case view. This view show that the elements of the four views work together seamlessly. This view is redundant with the other views but play two vital role;

- Its work as driver to help the designer during architectural design to discover architectural elements.
- This view also validates and illustrates the architecture design both on paper and as the starting point for the test of an architecture prototype.
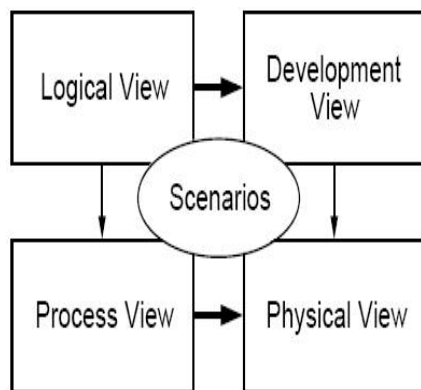

Fig. 3 "4+1 View Model"

VI. PROPOSED SOFTWARE ARCHITECTURAL DESIGN ONTOLOGY (SADO)

Based on the detailed information discussed in the previous sections, we propose software Architectural design ontology based on ISO/IEC 42010 an international standard for architecture description in this section.

A. Methodology/Method Used

Different researchers have used different methodology either predefined or proposed it to create ontology. Some proposed methodologies are: Lenat and Guah where they have introduced some general terms about cyc development in CycMethodology [13], Bernarasand colleagues introduced another methodology to create ontology in electrical domain networks called KACTUS [14]. Another methodology known as METHONTOLOGY was introduced by FIPA (Foundation for Intelligent Physical Agents) [1], SENSUS methodology was introduced in 1997 while creating SENSUS ontology [15]. Many other methodologies have been created for constructing ontology or for a specific task, e.g., ontology learning, ontology evaluation, ontology alignment etc. In our work we have fallowed the METHONTOLOGY because it is based on IEEE standard 1074-1995. Its framework enables construction of ontology at the knowledge level [1].

B. Ontology Construction Tools

Similarly various tools are used to construct ontology i.e. OntoEdit has been introduced by AIFB and marketed by Ontprise [16], OilED [17] and SMI (Stanford Medical Informatics) introduced Protégé [18] etc. In proposed ontology; Protégé has been used for construction of ontology because of its stand alone, extendable and easy to use architecture.

C. Ontology Languages

Different languages have been adopted to construct ontology where some of the important languages are as follows: Ontolingua, being the most expressive language, was introduced by combining first order predicates and KR paradigm of frame [19]. SHOE was first presented as an extension of HTML and later on was converted to a language based on XML syntax [20]. Similarly, RDF was introduced by w3c to describe web sources semantically. RDF Schema was an extension to the RDF and it is much less expressive [21] OIL formal semantic was based on description logic and introduced frame based KR primitive to RDFS [22]. OWL was introduced by w3c in 2001 and was approved as a w3c standard in 2004 [23]. In our work we have used DL (description logic) generation known as OWL-DL because of its adoptability, expressiveness, completeness and decidability.

D. Reasoners

Reasoners have been used to extract inferred information from ontology based on logical reasoning. Different reasoners have been used by researchers to inferred data, based on their requirement and language used to construct ontology. In our work we used Fact++ reasoners in protégé. In the fallowing figure all styles are the subclasses of the Model class; where actually we don't have all styles in Model. All these inferred classification has been done based on intelligent reasoning provided to the reasoners.

Acronyms are stored in separate classes to remove the ambiguity and make classification of knowledge easy.

The proposed ontology having 50 classes, 200 individuals, 10 object properties and 5 data properties.
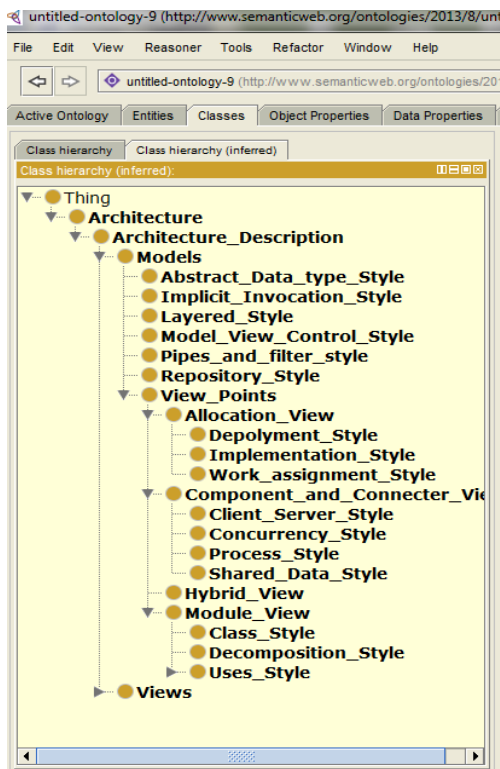
World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
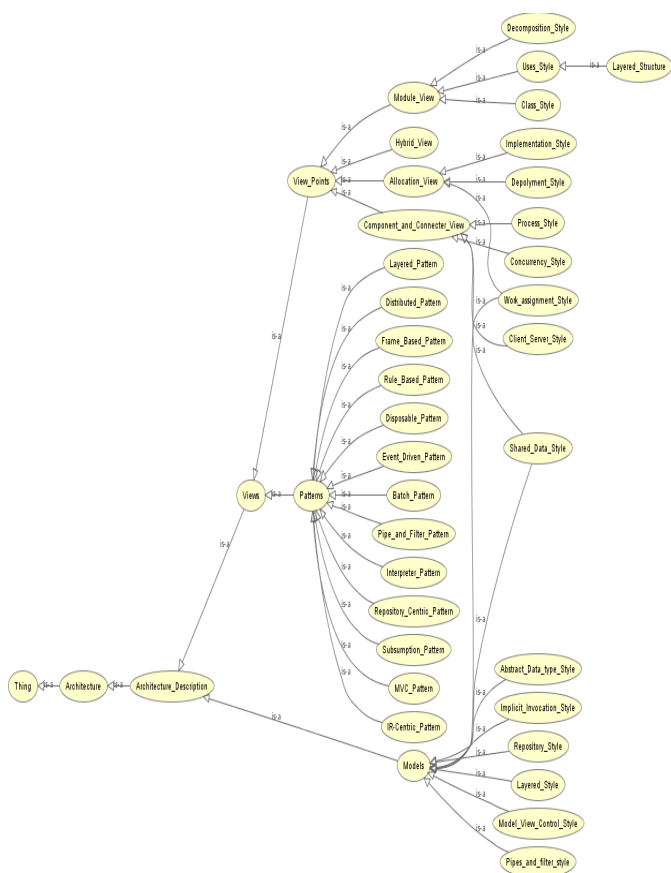Vol:7, No:12, 2013

Fig. 4 Inferred Hierarchy

Fig. 5 Fragment of the proposed

VII. CONCLUSION

The proposed ontology has been created to classify information about software Architectural design automatically and make it easily available for future use. This is a prototype ontology for software architecture and can be easily further extended by adding newly explore architecture models, views and styles.

REFERENCES

[1] F. López et al.. "Methontology: from ontological art towards ontological engineering", 1997.

[2] J. S. Kim and D. Garlan. Analyzing Architectural Styleswith Alloy. InISSTA 2006 Workshop on Role of Software Architecture for Testing and Analysis, pages 70–80, NY, USA, 2006. ACM.

[3] S. Wong, J. Sun, I. Warren, and J. Sun. A Scalable Approach to Multi-Style Architectural Modeling and Verification. In13th IEEE International Conference on Engineering of Complex Computer Systems, pages 25–34. IEEE Press, 2008.

[4] Sun, Jing, Hai H. Wang, and Tianming Hu. "Design Software Architecture Models using Ontology." *SEKE*. 2011.

[5] Kampffmeyer H., Zschaler S., Engels G., Opdyke B., Schmidt D. C., Weil F., "Finding the Pattern You Need: The Design Pattern Intent Ontology", in MoDELS, Springer, 2007, volume 4735, p: 211-225.

[6] Harb, Dania, Cédric Bouhours, and Hervé Leblanc. "Using an ontology to suggest software design patterns integration." *Models in Software Engineering*. Springer Berlin Heidelberg, 2009. 318-331.

[7] Hois, Joana, Mehul Bhatt, and Oliver Kutz. "Modular Ontologies for Architectural Design." *FOMI*. 2009.

[8] Kruchten, Philippe. "An ontology of architectural design decisions in software intensive systems." *2nd Groningen Workshop on Software Variability*. 2004.

[9] Inostroza, Pablo, and Hernán Astudillo. "Emergent architectural component characterization using semantic web technologies." *Proc. Second Int'l Workshop Semantic Web Enabled Software Eng*. 2006.

[10] Henninger, Scott, and Padmapriya Ashokkumar. "An ontology-based metamodel for software patterns." (2006).

[11] Emdanat, S., and E. Vakalo. "An ontology for conceptual design in architecture." *CAADRIA*. Vol. 98. 1998.

[12] Dietrich, Jens, and Chris Elgar. "Towards a web of patterns." *Web Semantics: Science, Services and Agents on the World Wide Web* 5.2 (2007): 108-116.

[13] D.B Lenat and R. V. Guha, "Building large knowledge-based systems; representation and inference in the Cyc Project," Addison-Wesley Longman Publishing Co., Inc., 1989.

[14] G. Schreiber et al., "The KACTUS view on the'O'word," In IJCAI workshop on basic ontological issues in knowledge sharing, 1995, pp. 159-168.

[15] A. Gómez-Pérez et al., "Towards a method to conceptualize domain ontologies," (1996).

[16] Y. Sure et al. "OntoEdit: Collaborative ontology development for the semantic web,"Springer Berlin Heidelberg, 2002.

[17] L. Li and H. Ian, "A software framework for matchmaking based on semantic web technology," International Journal of Electronic Commerce 8, no. 4, 2004, pp.39-60.

[18] N. Noy et al., "The knowledge model of Protege-2000: Combining interoperability and flexibility," In Knowledge Engineering and Knowledge Management Methods, Models, and Tools, Springer Berlin Heidelberg, 2000, pp. 17-32.

[19] J. Domingue et al., "Knowledge Modelling in WebOnto and OCML: A User Guide," Knowledge Media Institute, Milton Keynes, UK, 1999.

[20] A. Gómez-Pérez and O. Corcho. "Ontology languages for the semantic web," Intelligent Systems, IEEE 17, no. 1, 2002, pp.54-60.

[21] G. Klyne et al., "Resource description framework (RDF): Concepts and abstract syntax," W3C recommendation 10, 2004.

[22] D. Fensel et al., "OIL: An ontology infrastructure for the semantic web," Intelligent Systems, IEEE 16, no. 2 2001, pp. 38-45.

[23] D.L. McGuinness and F. V. Harmelen. "OWL web ontology language overview." W3C recommendation 10, no. 2004-03, 2004, pp.10.

[24] Kruchten, Philippe B. "The 4+ 1 view model of architecture." Software, IEEE 12.6,1995, pp. 42-50.