



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 5, May 2022

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.118

9940 572 462

6381 907 438

ijrset@gmail.com

www.ijrset.com

A Novel Framework for Error Checking in UML and UML to XML Conversion

Dr.Sumangala Patil

Associate Professor, Department of CSE, Lingaraj Appa Engineering College Bidar, Karnataka, India

ABSTRACT: The UML (unified modeling language) diagrams collections are modeled and it is called as a software design. The large number of various UML diagrams covering the software system for the different aspects. Which is the big risk, the incompleteness and inconsistent bring out in the overall system specification. First of all, we consider the UML diagrams alone for validation. In this paper, we have tested the various types of errors in the UML diagram alone and how it is solved by our proposed method. In addition, the UML model diagrams are converted into the XML for the better results and we compare this with the UML diagram validation alone using the techniques of attribute grammar for checking the XML document semantics consistency .we proposed a novel approach for consistency checking the model of UML diagrams. When it is translated the equivalent XML. By employing the approach of XML semantics UML diagrams have been converted into equivalent XML document, as it is the key idea so finally in this paper, the types of errors in the UML diagrams and errors has checked.

I. INTRODUCTION

As the technology has boon that gives that the Models make (software) platforms more accessible by lifting the level of concept and enabling the reuse of assets across platform transforms [7]. The Unified Modelling Language (UML) is an illustrious dominant object oriented modelling language for the design process of IT systems. In addition, there are various methods for authentication (check that the modelling is correct) and validation (check that the system fills the initial requirements) of UML diagrams [11] that are used to execute the comprehensive investigation about the design or program developed by a human. One method is Guided Inspection that used to test cases, to conclude if the model is complete, correct and consistent. Moreover, the Guided Inspection enhances the checklist with the testing concept of “coverage”. As a result of that, the coverage measures the establishment of how much the product being inspected and it has been examined. The test cases are selected from the plan so that, each case is represented by at least one test case. According to the previous papers, the authors suggest that the principles for the inspection process which will result the benefits (Objectivity, Traceability, Testability and Coverage) can be achieved. The Object- Oriented Reading Techniques (OORTs) [6] Inspection is a method to detect the early deficiency in software artifacts.

In this method, one of the major obstacles which have made error detection [19] is so hard in the flexibility of software development. As software development is a complex process, where the validity of a model depends on many factors in software development. There were numerous investigation applied to present a set of consistency rules [5] in order that the software errors can be reported when an application model violates the rules. However, these consistency rules are usually based on a specific development in the environment. This is similar to a compiler of a language. One compiler of a language can find errors in a program created in the language however it fails to find any error in a program which was created in other languages.

The Unified Modelling Language (UML) [17,4] has been widely recognized as standardized, general-purpose, object-oriented language for specifying, visualizing, constructing and documenting software systems. According to the outline mechanism in UML was originally proposed to expand the Unified Modelling Language to analyse the assured requirements from different domain applications. But the profile mechanism is also providing engineers an opportunity to design a meta-model where the validity of application models is defined. In comparably the profile mechanism is similar to the mechanism employed by some compiler generation tools such as YACC (BISON) [8] where the syntax of a new programming language is defined. In support to compiler generators, once specifications in a YACC/BISON file are defined, a compiler for the new language is automatically generated. As a result, any syntax error in a program written in the language can be detected. It is one of the most important features in YACC/BISON, if users revise the specification, then another compiler is generated so that different errors can be found.

In this study, we have projected the success of a novel technique, the XML element tag [18] attributes with the information of semantic are attached and these are append the semantic for XML documents. It is called as *XML semantics* [2]. Through the attributes of programming language the static semantics is checked and attached as the concept of AG's and based on the above we implemented this approach. The consistency of Unified Modelling Language (UML) is checked and we expand our semantics of XML for above checking. The artifacts of software



systems is specified, constructed, visualized and documented using the graphical language of UML which is the standard of OMG standard [3] [13]. The overall specification of the system becomes incomplete or inconsistent by the unlike types of diagrams which is covered by many aspects of the system and which is the major issue in the UML and it brings out the big risk. Therefore, the UML model completeness and consistency checking is very important. In the life cycle of software, using XML Metadata Interchange (XMI) the conversion of software tools from the UML and which is the key idea in this paper work. Between the various repositories, middleware and tools, the interchange mechanism standard [13] is a XMI. Using the XML semantics the XML documents are exported by implementing the approach, and the consistency and completeness of UML documents can be analysed.

II. PROPOSED METHOD FOR UML DIAGRAMS CHECKING

The proposed method is based on the idea where the compiler generators such as YACC/BISON and in addition we developed an automatic tool called ICER which allows software engineers in a profile to define the “syntax” of their application models. As a result of that, a model compiler is automatically fashioned where as it can discover the error which violates the “syntax”. As an application of the ICER tool, we apply the design patterns to elucidate that how the errors in an application model can be detected based on different reifications of a design pattern. The Design patterns have been widely used in software development because some classes of recurring design problems can be resolved by one design pattern. Thus, the reusability is maximally ensured during software development. Yet, the rectification of a design pattern can vary depending on development environments. The application of ICER can explain how a model compiler is employed to find errors in depending upon the different reifications of a design pattern.

The unified Modelling Language (UML) is an application of the four-level meta-modelling architecture, where each successive level is labelled from $M3$ to $M0$. Each level is usually named as meta-model, class diagram, and object diagram respectively. In the structural design, a model at the M_i level can be regarded as an instance of a model at the M_{i+1} level. The UML meta-model from the UML specification belongs to the $M2$ -level. The Object Management Group (OMG) [13] is proposed the profile mechanism to allow engineers to design a profile, which is a UML model at the $M2$ -level; therefore the existing UML notations can be extended. Based on the specific profile, engineers can design some application models at the $M1$ -level.

The existed system towards the central to the profile mechanism is the *stereotypes* and *instance of* relation between an application model at the $M1$ -level and its meta-model profile at the $M2$ -level. A stereotype is a meta-class which broadens the existing meta-class to represent some new concept in an application model. For instance, in the Factory Method design pattern, we can define a stereotype *Product* which extends the meta-class *Class* to represent a special class playing the *Product* role. After introducing stereotypes, developers have defined two kinds of constraints. Firstly, kind of constraints is called the graphic constraints, which are given in a profile such as the multiplicity of an association which connects two stereotypes. Secondly, the constraints are given by the Object Constraint Language (OCL) [20] [1]. Once a profile has been designed, software engineers can design an application model based on the profile. The *instance-of* relation connects each element in an application model with a meta-element in a profile. Thus, the two kinds of constraints given in a profile can be investigated. And, if we find a violation of constraints it should be reported as an error message between an application and its corresponding profile. In order to help software developers validate UML models at the application level, we have developed a tool, called ICER, whose major purpose is to perform the *instance-of* relation checking. Generally speaking, the term *instance-of relation checking* refers to the process in which a M_i -level model is checked to determine whether it is a correct instance of the corresponding M_{i+1} -level (meta) model. Through this research study, we are fascinated in the *instance-of* checking between $M2$ and $M1$ level models. Initially, the ICER tool is built in as part of IBM Rational Rose [4]. It reads in UML models represented as XML files that can be generated by modelling tools such as Rational Rose. Thus, the ICER tool is not restricted to IBM Rational Rose. Any UML CASE tool which exports to XML format can apply ICER to perform the *instance-of* relation checking. In this method, The ICER tool validates the graphical constraints before translating the UML model and OCL constraints into ASML [12], a high-level specification language based on abstract state machines. An external ASML compiler compiles the generated ASML code and the resulting executable code is finally run to validate OCL constraints. Furthermore, We have a graphical front-end to the tool in the form of an add-in to Rational Rose shown in Figure 1. In sequence to the above investigation about the tool, readers are referred to. Our tool can adapt to different applications depending on the UML profile that is provided as input. For instance, if a UML profile representing Enterprise Java Beans (EJB) [10] is given, we can check a corresponding class diagram that uses stereotypes defined for EJB. An additional example is if a UML profile representing class diagram refinement rules is given, we can authenticate the refinement of class models based on the rules. As this kind of malleability is one of the most important for the different developers to have different ideas of constraints to administer the same UML profile. By implementing our methodology consents them to create or change a profile based on different development environments and thus apply it to check application models.

III. XML SEMANTICS

The tag attributes of XML element is classified into the types: Dynamic attributes and static attributes [15] for the XML documents consistency checking by adding the semantic in our technique. The consideration of attributes as lexical by the former methods. The separation of semantics information into three subtypes: copy attributes, evaluated attributes and context attributes and it is used as intentional attributes is latter used. The evaluation algorithm are used to evaluate all these attributes and comparing the results with the static ones. The dynamic attributes is evaluated and the values are checked. In a formal (meta) language, the specification of semantics is specified using the techniques of XML which the another document of XML. Like the AG's, the role of the attribute evaluation is played by this specification document. And it is referred by SLXS (Specification Language of XML Semantics) [13] [5].

IV. AN ARCHITECTURE FOR UML CONSISTENCY CHECKER MODEL

Through the XML, for checking the consistency of UML models, a novel system is build and we are working for that. Based on the semantics checker [14] model of XML is the proposed system and it is represented in the figure 1. The division of the system takes place for the simplicity into three main parts.

a. Design Part of UML Model

This is the starting point of our model as in the figure 1. The designing of the UML model with the incompleteness and inconsistency problems by the user is started in this part. Using the existence tools of UML modelling this part can be takes place. Argo UML or IBM Rational Rose is some of the example.

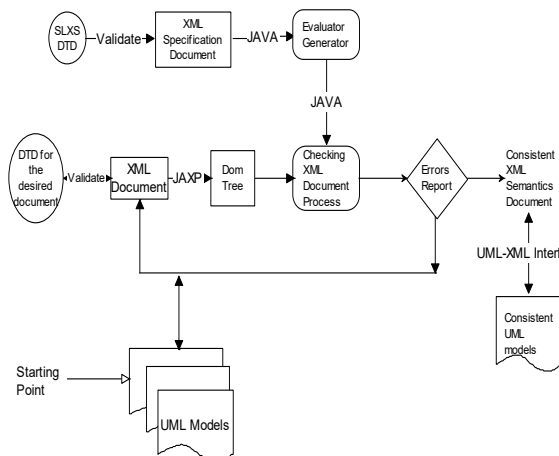


Figure 1: XML Semantic Checker Model

a. Checking Part of UML Model

The XML documents consistency is checked in this part by deployed our approach. The description of this model is as follows: Using the interface of UML-XML [9], the equivalent XMI documents is generated by translation the UML models. By using our checker model of XML semantics, the XML document is checked then. Due to the various incompleteness and inconsistency, the report is generated as output by the system at last. The consistent UML model is obtained either manually or automatically by validating these issues.

b. The part of UML-XMI Interface

The corresponding XMI documents and UML models translation is liable by this interface. Between the different tools of middleware and repositories, the standard mechanism of interchange is used. Among the metadata repositories (based on MOF OMG [1][13] [10]) and modelling tools (UML OMG based) the interchanging is easier and it is the main purpose of XMI which is distributed the heterogeneous environments. The three key industry standards are incorporated in XMI: An OMG modelling standard is Unified Modelling Language (UML), a W3C [1] [13] standard is Extensible Mark-up Language (XML), and the standard of metadata repository, modelling OMG and Meta Object Facility (MOF) [1]. The XMI is the consolidation of these three standards which agrees the technologies of modelling and the best of W3C metadata and OMG this makes the object models to share over the distributed systems by the developers over the internet and also with other metadata.



V. RESULTS OF THE PROPOSED METHOD

a. Types of Errors in UML Diagrams and UML to XML conversion:

The types of errors that may be occurred in the UML diagrams are: 1. Traceability: It refers that the diagrams are not clarity which is belong to the same system, 2. Layout Errors: The arrangements of diagrams are not same or unequal, 3. Notation: Not using correct UML notation, 4. Semantics: The concepts of target domain representation is invalidated, 5. Naming convention: The convention of naming is not accordance with the adopted convention, 6. Documentation: The performance of analysis and design decision missing justification through descriptions of natural language which is defected by the above. These types of errors can be checked and corrected by our proposed method. In this method, there might be a chance for consistency and incorrectness problems. These issues can be clarified by the above proposed method.

b. Advantages of Using UML diagrams to XML conversion:

The advantage of UML is the modification of domain-specific in easy manner and it is an open source to the user. And the main disadvantage of UML and advantage of UML to XML conversion is not bringing any ambiguity in the model interpretation and among the models and these results are using semantic rules and this is the lack of precise in UML diagrams. This makes errors in the UML diagrams while developing the software tools.

Analysis of the Result and Conclusion:

In earlier, there were many investigations have been carried out in the UML community to support the error detection. Based on these researches, we found that the most of the tools have provided a set of fixed rules in advance as a result of that any violation of these rules is reported as an error to engineers. However, we believe that the performance of the software development heavily depends on the experience of software engineers, who should be given the ability to define the errors in a model before developing a software system. The ICER tool, based on the profile mechanism, is one of the key attempts to return the rights of defining errors in a model to software engineers.

Through the early researches we opined that the application of the meta-model mechanism in defining the errors has drawn some interest. However, the previous investigations had not found in capable of validating the correctness between an application model and a profile. Despite the fact that some of these works introduced some new notations to define the correctness of an application instead of a profile, we hope that this study preambles the new notations, where it add some burden on software engineers to learn these new notations. Additionally, the new notations will increase some difficulty in implementing them into existing UML tools. In further, the application of Object Constraint Language (OCL) in supporting error detection is also important. As far as we know, the ICER tool is one of the few tools which fully support OCL. As an application of the ICER tool, we described how the Factory Method pattern is applied. In fact, the application of design pattern is widely used because they allow successful design solutions to be reused in software systems. However, design patterns are usually given in terms of structured, informal descriptions. Accordingly, a method is needed to validate an application model based on a particular design pattern. To employ this method, we proposed to implement a UML profile to reify a design pattern and thus to use the ICER tool to validate an application model against the particular profile. Since design patterns are informally described for a general solution, developers can reify a pattern quite differently depending on environments. This investigation brings new hope that by executing our methodology allows the developer to modify or create new profiles based on design patterns to satisfy the requirements from a specific development environment. With the emergence of model-driven software development, the ICER tool will play more and more important roles in helping developers to find errors in models during early phases of software development.

In this proposed study, the UML models competence and consistency is checked by executing our novel model. This model builds over the previous XML semantic checker model. This paper can simplify the following areas: The stiff checker model of UML Consistency is build, in the aspects of UML models the various completeness and consistency checking is success, The group of OMG and W3C our approach can be standardized.

REFERENCES

- [1].Abdelwah Hamou-Lhadj, Abdelouahed Gherbi and Jagadeesh Nandigam, “**The Impact of the Model-Driven Approach to Software Engineering on Software Engineering Education**”, ITNG -09 Proceeding of The 2009, 6th International Conference on Information Technology New Generations, pp.no: 719-724
- [2].Bhavana .V. Kumbhare and S. J. Karale, “**A Semantic Search over an XML data**”, International Journal of Advanced Research in Computer Science and Software Engineering, volume3,Issue5 may-2013 ,pp.no:27-30.



- [3] Bran.V.Selic, "On the Semantic Foundations of Standard UML 2.0", Journal of Object Technology, 12/2008,7, pp.no: 87-207.
- [4] "Best Practices for Software Development Teams-Rational Unified Process", Rational Software White Paper TP026B, Rev 11/01.
- [5] Dragan Djurić, Dragan Gašević, Vladan Devedžić, "Ontology Modelling and MDA", Journal of Object Technology, vol4, Jan-Feb 2005, pp.no:109-128.
- [6] Guilherme H. Travassos, Forrest Shull, Jeffrey Carver, Victor Basili, "Reading Techniques for OO Design Inspections", International Journal of Advanced Research in Computer Science and Software Engineering, volume3,issue6,June-2013 , pp.no:593-596.
- [7] James.R.Cordy, Thomas R. Dean, Andrew J. Malton Kevin A. Schneider, "Source Transformation in Software Engineering Using the TXL Transformation System", Journal of Information and Software Technology ,2002,vol 44, pp.no:827-837.
- [8] Jaroslav Porubän, Michal Forgac, and Miroslav Sabo, "Annotation Based Parser Generator", Computer Science and Information System 2010,vol7,issue2,pp.no: 291-307.
- [9] Jason E. Robbins and David F. Redmiles, "Cognitive Support, UML Adherence, and XMI Interchange in Argo/UML", Journal of information and software technology, special issues, The best of COSET'99 Jan 2000,vol42,no2, pp.no: 79-89.
- [10] Jawwad Wasat Shareef, Rajesh Kumar Pandey, "Component-Based Software Development with Component Technologies: An Overview", International Journal of Computer Science and Information Technology ,2012, vol 3(1) ,pp.no: 3029 – 3036.
- [11] Jayant.K.P, Renu Garg, Vinod Kumar, Prof. Ajaya Rana, "An Approach of Software Design Testing Based on UML Diagrams", International Journal of Advanced Research in Computer Science and Software Engineering, volume4,issue2 ,Feb-2014 ,pp.no:148-153.
- [12] Mike Barnett, Wolfgang Grieskamp, Lev Nachmanson, Wolfram Schulte, Nikolai Tillmann, and Margus Veanes, "Towards a Tool Environment for Model-Based Testing with AsmL", In proceeding Fates 2003,vol 2931,of LNCS Springerl verlag Berlin Heidelberg-2004,pp.no: 252-266.
- [13] Mohammed Abdalla Osman Mukhtar, Mohd Fadzil B. Hassan, Mohamed Nordin B. Zakaria, "A Proposed Engine Implementation Mechanism to Execute the Code of Relations Query/View/Transformation Language", ARPN Journal of System and Software vol 1, no-6 ,Sept 2011 ,pp.no: 209-214.
- [14] Praveen Madiraju, Rajshekhar Sunderraman and Shamkant B. Navathe, "Semantic Integrity Constraint Checking for Multiple XML Databases", Journal of Data Base Management ,vol 17, issue 4, 2006, pp.no: 1-19.
- [15] Pallavi Samdolikar, Sanchika Bajpai, "An Optimized Xml Routing Translation: X plus Path Discovery", International Journal of Advanced Research in Computer Science and Software Engineering, volume4,issue4 Nov-2014 , pp.no: 546-548.
- [16] Rosziati Ibrahim and Siow Yen Yen, "Formalization Of The Data Flow Diagram Rules For Consistency Check", International Journal of Software Engineering and Application, vol 1, no-4, Oct-2010 ,pp.no:96-111.
- [17] Rupinder Singh, Vinay Arora, "Technique for Extracting Subpart from UML Sequence Diagram", International Journal of Advanced Research in Computer Science and Software Engineering, volume3,issue6 , June-2013 pp.no:593-596.
- [18] Ting Chen, Tok Wang Ling, Chee-Yong Chan, "Prefix Path Streaming: a New Clustering Method for XML Twig Pattern Matching", pp.no: 1-10.
- [19] Vikas Gupta, Dr. Chanderkant Verma, "Error Detection and Correction: An Introduction", International Journal of Advanced Research in Computer Science and Software Engineering, volume2,,issue11 Nov-2012 pp.no:212-218.
- [20] Yoonsik Cheon, Carmen Avila, Steve Roach, and Cuauhtemoc Munoz, "Checking Design Constraints at Run-time Using OCL and AspectJ", International Journal of Software Engineering ,vol 2, issue 3, 2009,pp.no: 1-25



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details