Contribution of Restricting Rules on Real Time Applications using UML/OCL: A Survey

Shikha Singh, PhD Scholar,
Dr. Manuj Darbari, Professor,
Department of Computer Science & Engineering,
School of Engineering, BBD University
Lucknow, Uttar Pradesh
India

Abstract. Designing is one of the most important part in building of a real time system. A proper and efficient design ensures the seamless behaviors of the system in future. There are different approaches for designs. Object-oriented is one of the popular approaches used. One of the popular choice of language by designers for developing real time systems is UML, for enhancing UML notation in modeling of real time application several approaches have been developed. OCL acts as a cherry on the top of UML to ensure system behaves correctly. The Unified Modeling Language (UML) is a universally useful visual demonstrating dialect in which we can determine, imagine, and report the segments of programming frameworks. It catches choices and comprehension about frameworks that must be developed. UML has turned into a standard displaying dialect in the field of programming designing.
The Object Constraint Language (OCL) is a declarative language that portrays constraints or requirements on object-oriented models. A constraint is a limitation or restriction on at least one or more values of an object-oriented model. OCL is an industrial standard for object-oriented analysis and design (OOSD). The Object Constraint Language (OCL) began as a supplement of the UML documentation with the objective to beat the restrictions of UML (and in general, any graphical notation) in terms of precisely specifying detailed aspects of a system framework structure design. From that point forward, OCL has turned into a key part of any model-driven building (MDE) strategy as the default language for communicating a wide range of (Meta) model query, manipulation and specification requirements. Among numerous different applications, OCL is much of the time used to express model changes (as a major aspect of the source and target examples of transformation rules), well-formedness rules (as part of the definition of new domain-specific languages), or code-generation templates (as an approach to express the generation pattern examples and rules).This paper expects to give an exhaustive perspective of Object Constraint Language(OCL), its numerous applications and available tool support as well as the latest research developments and open challenges around it, especially in the territory of ecommerce.

Keywords: E-commerce, Security Features, OCL

I. Introduction

The Object Constraint Language (OCL) is an expression language that depicts constraints or limitations on object-oriented languages and other modeling artefacts. A limitation can be viewed as a confinement on a model or a framework. OCL is a piece of Unified Modeling Language (UML) and it assumes a vital job in the investigation period of the software lifecycle. Object Constraint Language (OCL), is a formal language to express side effect free constraint. Clients of the Unified Modeling Language and different language can utilize OCL to determine limitations and different articulations appended to their models. OCL is the expression language for the Unified Modeling Language (UML). To comprehend OCL, the segment parts of this announcement ought to be inspected. Therefore, OCL has the qualities of an expression language, a modeling language and a formal language.

1.1 Expression Language

OCL is a pure expression language. Accordingly, an OCL expression is destined to be without side effect. It can't transform anything in the model. This implies the state of the system will never show signs of change in light of an OCL expression, despite the fact that an OCL expression can be utilized to indicate such a state change (e.g., in a post-condition). All values for all objects, including all links, won't change. At whatever point an OCL expression is assessed, it just gives a value.

1.2 Modeling Language.

OCL is not a programming language, but a modeling language. It is beyond the realm of imagination to expect to compose program logic or write a flow-control statement in OCL. You particularly can't invoke processes or actuate non-query tasks inside OCL. Since OCL is a modeling language, in any case, not everything in it is guaranteed to be straightforwardly executable. As a modeling language, all execution issues are out of degree and can't be expressed in OCL. Each OCL expression is conceptually atomic. During evaluation, state of the objects in the system cannot change amid evaluation.

1.3 Formal Language

OCL is a formal language where all develops have a formally characterized importance and meaning. The determination of OCL is a piece of the UML specification.OCL isn't proposed to supplant existing formal languages, as VDM, Z and so forth

II. Why Formal Language?

In object oriented modeling, any graphical model, for example a class diagram, isn't sufficient to an exact all the specifications without ambiguity. There is a need to use extra features or constraints which may be able to extract the actual scenario about the objects in the model. Such limitations or constraints are easily and regularly portrayed in natural language. Practice has demonstrated that the evaluation of these diagrams will dependably result in ambiguities. To compose unambiguous limitations/constraints so-called formal languages have been developed.

2.1 Selecting OCL over other Formal Languages.

Conventional formal languages are useable to people with a solid mathematical foundation, yet hard to use for the normal business or system modeler. Object constraint language has been created to fill this space. It has been produced as a business modeling language inside the IBM Insurance division, and has its underlying foundations in the Syntropy technique, being founded on a set hypothesis and predicate logics and having a formal mathematical semantics.OCL is a formal language, which stays simple to peruse and compose. In object oriented modeling a graphical model, similar to a class diagram, isn't sufficient for an exact and unambiguous particular. There is a need to depict extra limitations about the items in the model. Such requirements are regularly depicted in natural language. Practice has demonstrated that this will dependably result in ambiguities. So as to compose unambiguous requirements, alleged formal languages have been produced. The drawback of traditional formal languages is that they are useable to people with a string mathematics foundation, however troublesome for the normal business or system modeler to utilize.

III. Applications of OCL

OCL can be utilized for various distinctive purposes:

- To determine invariants on classes and types in the class model.
- To determine type invariant for Stereotypes
- To depict pre-and post conditions on Operations and Methods
- To portray Guards
- As a navigational language
- To portray constraints on the methods

IV. History of OCL

The alleged Object Constraint Language (OCL) is a piece of the authority OMG standard for UML (from variant 1.1 on). OCL was produced at IBM in 1995, where an antecedent of OCL was characterized as a "business designing language", i.e. it was utilized as a tool in the modeling of business applications. OCL was especially motivated by the "Syntropy" language and technique, and one of the creators of Syntropy (Steve Cook) had close contacts at IBM with the lead creator of OCL (Jos Warmer). OCL turned out to be a piece of the IBM- led submission to the OMG for UML, and it was received as a formal detail language inside UML. In 1996, the OA&D Domain team at the Object Management Group (OMG) issued a demand for proposal on Object Analysis and Design. IBM and ObjecTime Limited jointly submitted a proposal in January 1997. A vital part of this proposal was the consideration of the Object Constraint Language, or OCL. Amid 1997 IBM and ObjecTime have worked together with the Unified Modeling Language (UML) accomplices and consolidated parts of their proposal with the UML proposition. This has brought about the current UML 1.1 proposal IBM's essential commitment to UML 1.1 is OCL. OCL was created by Jos Warmer as a language for business modeling inside IBM, got from the Syntropy strategy for Steve Cook and John Daniels. It is utilized inside UML both to help formalize the semantics of the language itself, and to give an office to UML clients to express exact requirements on the structure of models. An OCL imperative is a Boolean expression or predicate, which yields a value, either true or false, when assessed. To obtain Boolean values, the relational operators and OCL collection operators can be utilized.

**There are two main basic blocks of OCL, method definition and invariant declaration. Each OCL expression is defined with a data type. OCL provides a special kind of data type, namely OclAny, which includes two major sets of data types, primitive and collection data types. Primitive types include Integer, Real, String and Boolean. The collection data types include Set, OrderedSet, Sequence and Bag. Additionally, OCL expressions can be defined using basic operators, namely Boolean, relational, string and arithmetic operators. OCL expressions can be derived both from the user requirements specification (during the requirement specification phase) or design documents (during the design phase). OCL specification begins with a statement context, which is either used to define a class in the form of invariants or a class operation using preconditions and post-conditions.

V. Toolkit Available

Though UML has become the industry de facto standard notation for documenting software architectures, yet only a few UML tools offer any OCL support. OCL tool can be used to perform syntax check, semantic check of OCL expressions. A tool can be used to perform the dynamic validation of models. By generating code from OCL expressions, assertion tests providing automatic checking of invariants and pre/post conditions could be integrated into application code and executed at run time. A list of some of the popular OCL tools is as given below.

5.1 Dresden OCL Toolkit

The Dresden OCL Toolkit [35] is a modular toolkit for OCL designed and developed at the Drseden University of Technology. The tool consists of four modules: a parser, semantic analyzer, normalizer and a code generator. The semantic analyzer provides type checking and consistency check. Normalizer is used to reduce the code generator complexity. The injector tool generates Java code from OCL expressions, which provides evaluation of OCL expressions at run time.

5.2 ArgoUML

ArgoUML [36] is an open source tool that uses Dresden compiler and provides full OCL syntax and type checking.

5.3 Poseidon

Gentleware‘s Poseidon 1.4 [37] is a UML CASE tool evolved from ArgoUML. Poseidon offers specification of OCL constraint with syntax and semantics checking.

5.4 MagicDraw 5.5

MagicDraw [38] is a commercial UML modeling CASE tool ,and offers automatics OCL syntax validation and highlighting.

VI. Motivation of using OCL in Real time Application

With new technologies come new risk, this must be dealt with systematically. With growing digitalization grows the risk of unauthorized access. We have a handful of real time scenarios demonstrating the threats caused by unauthorized access. This situation demands the fulfillment of different security requirement. Proper constraints planned during modeling the structure of the system helps in enforcing security to private data of users. This motivates for applying constraints in restricting unauthorized access to data.
Data breaches happen daily, in too many places at once to keep count. But what constitutes a huge breach versus a small one? CSO compiled a list of 17 of the biggest or most significant breaches of the 21st century.

- This list is based not necessarily on the number of records compromised, but on how much risk or damage the breach caused for companies, insurers and users or account holders.
- In some cases, passwords and other information were well protected by encryption, so a password reset eliminated the bulk of the risk.

With the alarming picture of the effects of unauthorized access of data it became utmost important to ensure security of the data flowing digitally.

- But implementing such higher-level organizational authorization policies in computer systems can be cumbersome.
- Maintaining such a huge rule book would require huge amount of care and attention.

This process would even be inefficient as there would be high risk of failure.

VII. Frame work of approach

Application of constraints on access would contribute to the protection of digital information. Proper modeling of architecture keeping privacy in mind is required in the process of protecting information. Role based access control would distribute the duty of protection among different roles. This would depict a clean and transparent picture of the manipulation of the digital data. The process of identifying potential threats/risk factors and implementing improvement efforts to alleviate or prevent possible negative events resulting into enhancement of the application would be carry out.

VIII. Related Work

In today’s online digital world, protecting the privacy of the user data is a big challenge. Recent data breach incidents and data harvesting issues has necessitated the researchers to take this issue on priority and compelling them to find an immediate and effective solution to tackle this alarming problem. As a solution to the security problem, security architecture for privacy preservation and identity management is proposed in [1].

In [2], Alison concluded that "Anything you put the internet has the potential to be made public". He was wondering about sharing personal information with total strangers on the internet in social networks. He stated the dangers regarding having public user profile, especially, children who are well publicized. He added that the sheer volume of personal information that people are publishing online is changing the nature of personal privacy.

There are multiple projects that are being taken at national level to address the problem of privacy breach and identity theft. Some of the project are stated in [3,4].

The problem of managing privacy must be considered mandatory at the design phase of ambient intelligence services. This is the principle of " Privacy by Design".[5] propose a semantic framework incorporating a meta-model and a middleware that helps any ubiquitous system designer to easily implement mechanisms to manage users' privacy policies that are effective, adaptive and semantically interoperable.

[6] Focuses on authorization policies to demonstrate how software engineering techniques can help validate authorization constraints and enforce access control policies. The approach leverages features and functionalities of the UML/OCL modeling methods as well as model driven approach to represent and specify authorization model and constraints.

[7] Make use OCL UML for building a tool to automate verbalization of business rules. They synthesizes and compares the three approaches to designing business rules together.

An approach for controlling the social actions that web 2.0 applications allow users to execute [8].The control over these actions is defined with UML/Object Constraint Language (OCL) and then demonstrated through a prototype system.

A framework to support formal modeling and contracts for data-centric web services is demonstrate to be used as to verify correctness properties for composition of services [10].

Verbalization of business rules translates the rules expressed in a design language into semi-natural expressions. This allows business experts to validate models expressed in a design language without implying any skill on this language.[11] A transformation tool is proposed to automate verbalization and applied to OCL (Object Constraint Language) constraints in the utility domain.

More formal programming methods of OCL have been introduced in [16] also the powerful combination of UML and OCL are discussed. Mode Driven Architecture (MDA) is also utilizing the application of OCL and UML as well.

Being the part of UML, OCL has been utilized in many theories and object oriented system modeling. Such contributions are well discussed in [17]. For implementing security Role Based Access control (RBAC) has been developed in [18]. For object oriented analysis and design OCL has been used. Using OCL it describes that what should be constraints for the RBAC approval. OCL has been used as constraint for modeling element in a textual specification scenario [19]. OCL expressions and constraints are translated into Java language. A run time assertion checker is used in JML tool.

OCLLib, OCLUnit and OCLDoc are proposed in [20]. OCL lip makes easier for the development of OCL expressions and constraints. It makes a high reuse factor which is configurable and testable named as OCL unit and OCL DOC. For model comprehensibility and maintenance, OCL has been used in [21]. We have utilized this concept in development of RTCN framework.

For achieving significant advantage, OCL constraints UML have been introduced. OCL can be identified as similar constraint language called progress, called path expressions. The development of progress and OCL with different extensions has been proposed in [22]. An OCL based unified modeling language has been proposed in [23], for model driven representation of distribution systems, also Role based access control with authorization constraints. The overall objective is to develop be Secure Distributed System (SDS).

A security policy using Object Constraint Language has been developed in [24] secure MOVA tool utilized to answer the result of the proposed approval verification of non trivial security properties. An access control Meta model has been developed [25] using unified modeling language (UML). This modeling language is independent of access control requirement using genetic mutation along with independentness from specific implementations.

An experiments framework has been proposed in [26], for evaluation the constraints in object constraints language, along with Automated support for OCL refactoring. Modeling Business process is one of the important issues in maintaining competitions and dealing with challenges in business environment.

IX. Conclusion

UML along with OCL can be used for the application of constraints during modeling. Moreover, owing to the fact that OCL has proved its applicability in several industrial applications, OCL is a good means for such a practically relevant process like the design of access control policies. User data can be divided into sensitive and non sensitive information. Emphasis should be laid in the protection of sensitive data. Some probable features for ensuring user data security can be:

- User is given complete ownership to protect his credentials.
- Database owner will not be able to update the data directly through database query (Update is prohibited).
- Only users will be able to add/update data through front end interface.
- Such role based access will add greatly to the security of data.

X. Future Plan

Study the existing constraints for protecting data. Detection of the missing constraints or scope to improve existing constraints. Implementation of the constraint for improvement. We also plan to explore other fields of modeling to inspect the possibility of fusion of other approaches like ontology, fact based approaches etc. to be advantageous to the process.

References

[1]. A Conceptual Security Architecture for Privacy preservation in Identity Management System
[2] Alison George. Living online: The end of privacy? New Scientist, 2569, 2006. URL http://www.newscientist.com/channel/tech/mg19125691.700-living-online-the-end-ofprivacy.
[3] Liberty-Alliance. Liberty ID-FF Architecture Overview. Version: 1.2-errata-v1.0.
http://www.projectliberty.org/specs/liberty-idffarch- overview-v1.2.pdf, 2003. Liberty 2003
[4] OPENID, http://openid.net/, 2010. Openid2010
[5] Semantic Framework for Managing Privacy Policies in Ambient Intelligence
[6] Articulating and Enforcing Authorisation Policies with UML and OCL
[7] Verbalization of business rules: Application to OCL constraints in the utility domain
[8] Z.Maamar, V.Buregio, N.Faci, D.Benslimane, Q.Z.Sheng, "Controlling " Web 2.0 applications in the workplace,19th International Enterprise Distributed Object Computing Conference(EDOC), pp.191-200, 2015.
[9] L.Haibing, Z.Ning ,L.Yonglin,L.Xiaobo ,Z.Yifan "EMF Based Validation Methods of the Static Semantics of Models", 2nd International conference on Information Science and Control engineering, IEEE, pp. 207-211, 2015.
[10] R.Baghli, B.Traverson ,"Verbalization of business Rules Application to OCL Constraints in the Utility Domain", 2nd International conference on Model Driven engineering and software development , IEEE, pp. 348-355, 2014.
[11] I.Saleh, G.Kulczycki, M.B.Blake, "Formal Specification and Verification of Data Centric Service Composition", International conference on Web Services (ICWS) IEEE, pp. 131-138, 2010.
[12] T. Tilley, R. Cole, P. Becker, P. Eklund, A survey of formal concept analysis support for software engineering activities, LNCS, vol. 3626, pp. 250-271, 2005.
[13] F. Oquendo, ∏-method: a model driven formal method for architecture – centric software engineering, ACM SIGOFT Software Engineering Notes, vol. 31, no. 3, pp. 1-13, 2006.
[14] M. Moller, E. –R. Olderog, H. Rasch, H. Wehrheim, Integrating a formal method into a software engineering process with UML and Java, Formal Aspects of Computing, vol. 20, pp. 161-204, 2008.
[15] R. V. D. Straeten, V. Jonckers, T. Mens, A formal approach to model refactoring and model refinement, Software & Systems Modeling, vol. 6, no. 2, pp. 139-162, 2007.
[16] J.WARMER, A.KLEPPE. The objet constraint language: Getting your models ready for MDA, Addison- Wesley, Booston, NA, 2003.
[17] T. CLARK, J. WARMER (Eds), Object Modeling with the OCL , the relational behind the object constraint language, Berlin, Heidelberg, Network, Springer(2002.).
[18] G.J. Ahn, M. E.SHIN, Role Based authorization constrains specification using Object Constraint language, Tenth IEEE International workshop are Enabling technologies: Infrastructure for collaborative ,USA, PP,157-162,2001.
[19] A. HANIE, Translating the object constraint language into the Java modeling language ACM symposium on Applied computing, 2004 PP. 1531-1535.
[20] C.-O. Joanna, OCLLib, OCLUnit, OCLDoc: pragmatic extensions for the object constraints language, LNCS vol. 5795, 2009, 665-669.
[21] L. C. Briand, L. Yvan, H. D. Yan, N. Dipenta, A controlled experiment on the on the impact of the object constraint language in UML based maintenance, 20th IEEE International Conference on Software Maintenance, (2004), Canada, pp. 380-389.
[22] A. Schurr, adding group transformations concepts to UML constraint language OCL, Electronics in theoretical computer science, vol. 44 Nov, July 2001, pp. 93-06.
[23] T. Lodderstedt, D. Basin , J. Doser, Secure UML: AUML- based modeling language for Model-Driven security, LNCS,vol.2460,2002,pp.426-441.
[24] D. Basin, M. Clavel, J. Doser, M. Eaea, automated analysis of security design models intonation and software technology, vol. 51 no. 5, 2009,pp 815-831.
[25] S. Nadera, K. Hemanth, A. Kamel, L. Luegi, VACML: unified access control modeling language, 2011 4th IFIP international conference on new technologies, modeling and security (NTMS), Canada, pp.1-8.
[26] A.L. Correa, C.M.L. Werner, M. Barros, Refactoring to improve the under standability of specifications written in object constraint language, IET, software, vol. 2, pp-69-90.
[27] C. amelunxen, A. Schurr formalizing model transformation rules for UML/MOF2,IET software, vol. 2 no 3,(2008).pp. 204-222.
[28] F. Barbier, B. Henderson-Seuers, formalizing of whole part relationship with unified modeling language ,IEEE transactions on software engineering, vol. 29, no. 5, pp. 459-470, 2003.
[29] K. Sohr, M. Drouineaud, G.J. ahn, M. Gogolla Analyzing and managing role based access control polices, IEEE transactions on knowledge and data engineering, vol. 20(2008),pp. 924-939.
[30] L.C. Brjand, Y.Labiche, M .Dipenta, H. yan-Bondoe, an experimental investigation of formality in UML based development, IEEE transaction on software engineering, vol. 31 No. 10,(2005) pp. 833-849.
[31] F. Buttner, M. Gogolla, Modular embedding of the object constraints language into a programming language, LNCS, vol.7021, 2011 pp. 124-139.
[32] F. S. de Boer , Object oriented and constraint programming for Time Critical Application , LNCS , Vol. 1743 , 1999 , pp. 255-256.
[33] P. Bouyer, F. Laroussinie, P. A. Reynier, Diagonal constraints in timed automata: forward analysis of timed system, Formal Modeling and Analysis of Timed Systems, LNCS, vol. 3829, pp.112-126, 2005.
[34] Kjetil Mage, "A Practical Application of the Object Constraint Language OCL", Agder University College (2002).
[35] Dresden OCL Toolkit, Dresden University of Technology.
http://dresden- ocl.sourceforge.net/index.html
[36] ArgoUML. http://argouml.tigris.org/
[37] Poseidon 1.4, Gentleware. http://www.gentleware.com/products/index.php3
[38] MagicDraw 5.5. http://www.magicdraw.com/