Introduction to WUCOR (1st International Workshop on UML Consistency Rules)

Damiano Torre (Primary Contact)
Carleton University
Software Quality Engineering Laboratory
Ottawa, Canada
University of Castilla-La Mancha
ALARCOS Research Group
Ciudad Real, Spain
dctorre@sce.carleton.ca

Yvan Labiche
Carleton University
Software Quality Engineering Laboratory
Ottawa, Canada
labiche@sce.carleton.ca

Marcela Genero
University of Castilla-La Mancha
ALARCOS Research Group
Ciudad Real, Spain
marcela.genero@uclm.es

Maged Elaasar
Carleton University
Software Quality Engineering Laboratory
Ottawa, Canada
melaasar@gmail.com

INTRODUCTION

The Model Driven Architecture (MDA) [1] is an approach to the development of software systems that promotes the use of transformations between successive models from requirements to analysis, to design, to implementation, and to deployment [2]. Much attention has been paid to MDA by academia and industry in recent years [3], which has resulted in models gaining more importance in software development. The Unified Modeling Language (UML) [4] is the Object Management Group's specification most frequently used and is the de-facto standard modeling language for object-oriented modeling and documentation [5]. It is the most commonly used modeling language to implement the MDA although it should not be used in every single software development project [6]. The UML provides 14 diagram types [4] that can be used to describe a system from different perspectives (e.g., structure, behavior) or abstraction levels (e.g., analysis, design), which helps deal with complexity and distribute responsibilities between stakeholders. Those diagrams help support many software development activities, such as: transforming an analysis model into a design model, transforming a design model into an implementation, generating documentation, model-driven testing, model-driven validation and verification, performance estimation, and schedulability analysis. Since the various UML diagrams describe different perspectives of one, and only one, software under development, they strongly depend on each other and hence must be consistent. To be successful, any software development activity that consumes a UML model made of diagrams, such as the ones mentioned earlier, requires that those diagrams be consistent. As UML is not a formal notation, inconsistencies may arise in the UML specification of a complex software system when such specification requires multiple diagrams to describe different perspectives of the software [7]. When UML diagrams portray contradicting or conflicting meaning, the diagrams are said to be inconsistent [8]. Such inconsistencies may be a source of faults in the software system [9]. It is therefore paramount that they be detected, analyzed and fixed [10], which requires that consistency between the diagrams of a UML model be first specified. One can find some UML diagram consistency specifications in the UML standard itself, where they are often referred to as well-formedness rules. As discussed in the literature, one can reason about consistency according to different dimensions: Horizontal vs. Vertical vs. Evolution Consistency, Syntactic vs. Semantic consistency, and Observation vs. Invocation consistency [11]. One can find consistency specification in the UML standard itself. One can also imagine consistency specification that is specific to a domain (e.g., telecom, aerospace), to an organization, to a project or a team. Even though there is a need for UML diagram consistency, even though there exist different ways to reason about consistency rules, one can observe from the literature [11] that: 1) there is no well-accepted set, as complete as possible, of consistency specification rules, or simply rules, for UML diagrams (beyond the small set of well-formedness rules in the standard specification); 2) many researchers have proposed, explicitly or implicitly, rules to detect inconsistencies, without any effort to validate those rules; 3) the majority of the consistency rules target a small subset of the UML diagrams (mostly, class, sequence, and state machine diagrams); 4) a non-negligible set of consistency rules are provided over and over again by researchers (instead of, for instance, referring to an accepted list of such rules); 5) a non-negligible set of consistency rules presented by researchers are actually included in the UML standard itself; 6) the UML standard is far from providing a comprehensive set of consistency rules; 7) the vast majority of consistency rules are horizontal and syntactic (other dimensions are barely used in those rules). These observations motivated WUCOR, during which we sought the opinion of experts about the consistency rules researchers have been defining in the literature, and the

rules that may be missing. The goal of this workshop has been to gather community input and feedback on UML consistency rules in general. WUCOR provided an opportunity for researchers who have been working on UML consistency, or whose (research) activities require consistent diagrams, to engage with each other in a highly interactive venue so that the group could validate the rules that have been collected and pave the path for future initiatives. The objective of the workshop has been to bring together any one, either from the industry or academia, interested in consistency rules between UML diagrams of a given model, and to provide a platform for discussions, interactions and collaborations regarding this topic. One of the starting point for the discussion groups was the set of 190 unique consistency rules we have coalesced in our work [12]. We also asked for expert opinion about a subset of those rules that are deemed paramount, and should therefore always be enforced, and other rules that can be considered optional. The final program of the WUCOR is presented in TABLE I.

TABLE I. SCHEDULE OF WUCOR

Time	Duration	Activity
8:45am	5min	Welcome to WUCOR
8:50am	25min	Bernhard Hoisl and Stefan Sobernig. Consistency Rules for UML-based Domain-specific Language Models: A Literature Review
9:15am	25min	Dan Chiorean, Vladiela Petrascu and Ioana Chiorean. Proposal for Improving the UML Abstract Syntax
9:40am	40min	1st Actvity about dimensions of UML Consistency
10:20am	25min	Coffe Break
10:45am	1hr	2st Actvity about UML diagrams involved in UML Consistency
11:45am	1hr15min	Lunch Break
1:00pm	10min	Introduction to UML Consistency Rules
1:10pm	1hr50min	3rd Activity about UML consistency rules in Model-Driven Development
3:00pm	20min	Coffe Break
3:20pm	1hr25min	Discussion and Presentation of Results
4:45pm	15min	Conclusion, Summary and Next Steps

The WUCOR proceedings collect the two papers presented at the workshop (shown in TABLE I). Those submitted papers were peer-reviewed by three independent reviewers. The two accepted papers discuss 1) a review about UML-based Domain-specific Language Models, and 2) a proposal for Improving the UML Abstract Syntax; both papers were considered very related to UML Consistency rules issues.

We would like to thank the authors for submitting their papers to WUCOR. We are also grateful to the members of the Program Committee and to the MODELS 2015 organizers for their support during the workshop organization. For more information about WUCOR please visit the workshop website at https://wucor.wordpress.com. The Program Committee was composed by :

- Steve Cook, Hidden Symmetry Ltd, UK
- Alexander Egyed, Johannes Kepler University, Austria
- Kenn Hussey, Committerati Consulting, Canada
- Zbigniew Huzar, Wroclaw University of Technology, Poland
- Robert Karban, Jet Propulsion Laboratory, USA
- Florian Noyrit, CEA LIST, France
- Richard Paige, University of York, UK
- Gianna Reggio, Università di Genova, Italy
- Nicolas Rouquette, Jet Propulsion Laboratory, USA
- George Spanoudakis, City University London, UK
- Mehrdad Sabetzadeh, University of Luxembourg, Luxembourg
- Miroslaw Staron, University of Gothenburg, Sweden

ACKNOWLEDGMENTS

This work has been funded by the SIGMA-CC project (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-36904) .

REFERENCES

[1].Mukerji, J., Miller, J.: Overview and guide to OMG's architecture. Object Management Group (2003), http://www.omg.org/mda/
[2].Thomas, D.: MDA: Revenge of the modelers or UML utopia? IEEE Software 21, 15–17 (2004)
[3].Lucas, F.J., Molina, F., Toval, A.: A systematic review of UML model consistency management. Information and Software Technology 51, 1631-1645 (2009)
[4].OMG: OMG Unified Modeling LanguageTM - Superstructure Version 2.5. Object Management Group (2013)
[5].Pender, T.: UML Bible (2003)
[6].Petre, M.: UML in practice. Proceedings of the 35th International Conference on Software Engineering, pp. 722-731. (2013)
[7].Ibrahim, N., Ibrahim, R., Saringat, M.Z., Mansor, D., Herawan, T.: Consistency rules between UML use case and activity diagrams using logical approach. International Journal of Soft. Engin. and its Applicat. 5, 119-134 (2011)
[8].Simmonds, J., Straeten, R.V., Jonkers, V., Mens, T.: Maintaining Consistency between UML Models using Description LogicZ. RSTI – LMO'04 10, 231-244 (2004)
[9].Muskens, J., Bril, R.J., Chaudron, M.R.V.: Generalizing Consistency Checking between Software Views. Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture, pp. 169-180. (2005)
[10].Spanoudakis, G., Zisman, A.: Inconsistency management in software engineering: Survey and open research issues. In: Chang, S.K. (ed.) Handbook of Software Engineering and Knowledge Engineering, pp. 329-380. (2001)
[11].Torre, D., Labiche, Y., Genero, M.: UML consistency rules: a systematic mapping study. (EASE 2014). (2014)
[12].Torre, D., Labiche, Y., Genero, M., Elaasar, M.: A systematic identification of consistency rules for UML diagrams. Carleton University (2015), http://goo.gl/TFMgnE