1st International Workshop on UML Consistency Rules (WUCOR 2015) – Post workshop report

Damiano Torre*‡, Yvan Labiche*, Marcela Genero‡, Maged Elaasar*,
Tuhin Kanti Dasβ, Bernhard Hoisl¥, Matthias Kowal∞

*Carleton University - Ottawa, Canada
‡University of Castilla-La Mancha - Ciudad Real, Spain
βQueen's University - Kingston, Canada
¥Vienna University of Economics and Business - Vienna, Austria
∞Technische Universität Braunschweig - Braunschweig, Germany

dctorre@sce.carleton.ca, labiche@sce.carleton.ca, marcela.genero@uclm.es, melaasar@gmail.com, das@cs.queensu.ca, bernhard.hoisl@wu.ac.at, m.kowal@tu-bs.de

DOI: 10.1145/2894784.2894801
http://doi.acm.org/10.1145/2894784.2894801

ABSTRACT

The Unified Modeling Language (UML), with its 14 different diagram types, is the de-facto standard modeling language for object-oriented software modeling and documentation. Since the various UML diagrams describe different views of one, and only one, software system under development, they strongly depend on each other in many ways. In other words, the UML diagrams describing a software system must be consistent. Inconsistencies among these diagrams may be a source of faults during software development and analysis. It is therefore paramount that these inconsistencies be detected, analyzed and – hopefully – fixed. The goal of this workshop was to gather input and feedbacks on UML consistency rules from the community. This workshop provided an opportunity for researchers who have been working in the area of UML consistency to interact with each other at a highly interactive venue, improve the body of knowledge on UML consistency rules and discuss ideas for further research in this area. This report summarizes details of the workshop and the results obtained that day.

Categories and Subject Descriptors

D.2.4 Software: Software Engineering: Software/Program Verification - *Model checking, Validation*

General Terms

Documentation, Design, Languages, Verification.

Keywords

Unified Modeling Language, UML, UML consistency, UML consistency rules, Verification.

1. INTRODUCTION

The Model Driven Architecture (MDA) [1] is an approach for 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 [3-5], which has resulted in models gaining more importance in software development. The Unified Modeling Language (UML) [6] is the Object Management Group's most frequently used specification, and is the de-facto standard modeling language for object-oriented modeling and documentation [7]. It is the modeling language most commonly used to implement the MDA although it might not be used in every single software development project [8]. The UML provides 14 diagram types [6] and 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 among stakeholders. These diagrams help to 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 must therefore be consistent. To be successful, any software development activity that employs a UML model made up of diagrams, such as those mentioned earlier, requires that those diagrams must be consistent. As UML is not a formal notation, inconsistencies may arise in the UML specification of a complex software system, especially when this specification requires multiple diagrams to describe different perspectives of the software [9]. When UML diagrams portray a contradictory or conflicting meaning, the diagrams are said to be inconsistent [10], and these inconsistencies may be a source of faults in the software system [11, 12]. It is therefore paramount that they be detected, analyzed and fixed [13], signifying that the consistency among the diagrams of a UML model must first be specified. It is possible to find some UML diagram consistency specifications in the UML standard itself [6], in which they are often referred to as well-formedness rules.

As discussed in the literature, and later in this document, it is possible to reason about consistency according to different dimensions: Horizontal vs. Vertical vs. Evolution consistency, Syntactic vs. Semantic consistency, and Observation vs. Invocation consistency. The UML standard itself contains consistency specification, and it is also possible to imagine consistency specification that is specific to a domain (e.g., telecom, aerospace), to an organization, to a project or to a team.

Despite the fact that there is a need for UML diagram consistency, and even though different ways in which to reason about consistency rules exist, literature [14] shows that: 1) there is no well-accepted set (that is 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 explicitly or implicitly proposed rules with which to detect inconsistencies, without any effort being made 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) researchers have repeatedly presented non-negligible sets of consistency rules (rather than, for example, referring to an accepted list of such rules); 5) non-negligible sets of consistency rules presented by researchers are actually included in the UML standard itself; 6) the UML standard is a long way from providing a comprehensive set of consistency rules; 7) the vast majority of consistency rules are horizontal and syntactic (the remaining dimensions have barely been considered in those rules). These observations motivated a proposal for a workshop, during which we sought experts' opinions about the existing consistency rules we collected from a literature search, and the rules that may be missing. This workshop provided an opportunity for researchers who have been working on UML consistency, or whose (research) activities require consistent diagrams, to work together at a highly interactive venue with the objective of validating previously collected rules [15], and discuss ideas for further research in this area.

This paper is structured as follows. In Section 2, we provide a brief discussion on previous events on UML consistency. This is followed by the specification of the workshop goals and proceedings (Section 3), and a summary of the papers presented (Section 4). A detailed description of the activities is presented in Section 5. A preliminary discussion, along with the main findings and the limitations of the research, are provided in Section 6. Finally, Section 7 draws the conclusions and provides directions for future works.

2. PREVIOUS EVENTS ON UML CONSISTENCY

We are not aware of any conference that is specifically dedicated to the issue of consistency among UML diagrams. Our systematic mapping study [14], during which we carried out a rigorous and systematic search, did not find any such event. We are only aware of a workshop entitled "Workshop on Consistency Problems in UML-based Software Development" which took place in conjunction with the UML conference in 2002, 2003, and 2004. These workshops differ from the WUCOR, celebrated within MODELS 2015, in two main ways. First, they were seeking contributions from authors on any kind of issue regarding the consistency of UML diagrams (e.g., consistency rule specifications, tooling support with which to check rules, inconsistency repair strategies), whereas we wished to specifically focus on the consistency rules/specifications that seem to be needed by the research community in order for one set of consolidated rules to be defined. Secondly, as it will become clearer later in this document, WUCOR was seeking paper contributions that would be presented not only in a similar way to the majority of the workshops that take place concurrently to conferences, in a format akin to a small conference. On the contrary, we primarily encouraged working groups to debate specific issues during the workshop.

3. WORKSHOP GOALS AND PROCEEDINGS

The objective of the workshop was to bring together anyone, from either industry or academia, who was interested in consistency rules among the UML diagrams of a given model, and to provide a platform for discussions, interactions and collaborations regarding this topic. The goal was to gather input and feedback on the topic of UML consistency rules from the community. One of the starting points for the discussion groups was the set of 190 unique consistency rules that we had coalesced in our previous work [15]. We also asked for expert opinion on the subset of those rules that should be deemed paramount, and should therefore always be enforced, and the other rules that can be considered optional.

The papers presented at the workshop were collected in the WUCOR proceedings [16] and were peer-reviewed by three independent reviewers.

4. SUMMARY OF THE PRESENTED PAPERS

In this section we summarize the main results of the two papers that were accepted at WUCOR 2015.

Hoisl and Sobernig [17] analyzed consistency aspects extracted from 84 UML-based domain-specific modeling language (DSML) designs collected via a systematic literature review [18]. They focused exclusively on consistency rules defined at the level of a DSML. For the evaluation of UML consistency aspects, they adopted criteria from closely related work [14, 15]. They then interpreted the consistency-related data extracted in order to discuss frequently identified defects in UML-based DSML language models. Their study showed that a UML-based DSML is predominantly formalized via the definitions of profiles, which mostly specialized the class, activity, component, and package diagrams. They also noted that constraints, which are specified in natural language or OCL, are most frequently used in combination with these profiles to define consistency rules in a single model for verification purposes. In the majority of cases, they found that the DSML papers do not describe any tool support with which to enforce these rules.

D. Chiorean, Petrascu, and I. Chiorean [19] proposed a change of attitude regarding the definition of the UML's abstract syntax that would improve the quality of the standard specification. They described this improvement as a condition for attaining the value of model-driven technologies and paradigms. Their proposal is supported by examples taken from the UML specification [6]. They argue that the first requirement is for the well-formedness constraints in the specifications to have complete, accurate and clear natural language descriptions. Once this requirement is met then the constraints should have associated formal specifications in OCL. Finally, the OCL specifications for those constraints have to be compared to those of the natural language and any synchronization between the two must, if necessary, be carried out.

5. DESCRIPTION OF THE ACTIVITIES

In this section, we describe the three activities that were conducted during WUCOR (see the details of the WUCOR program [16]). These activities consisted of:

- Activity 1 (A1) focused on the definitions of the types and dimensions of UML consistency rules. Its main goal was to ensure that the attendees would work with common definitions during the remaining activities. This activity was divided into two parts, as explained in Section 5.1, and was allotted 40 minutes.
- Activity 2 (A2) provided an overview of the state of the art in terms of the involvement of UML diagrams in UML consistency rules. This activity was divided into two parts, as explained in Section 5.2, and was allotted one hour.
- Activity 3 (A3) concerned the validation of some of the 190 UML consistency rules that we had collected in [15]. This activity was allotted one hour and 50 minutes.

The results of these three activities were discussed in the last session of WUCOR in one hour and 25 minutes.

A detailed description of these three activities is provided in the following three subsections.

5.1 UML Consistency rules and dimensions (A1)

In the first part of activity A1, we provided the attendees with the most frequently used definitions [14] of the three UML consistency dimensions (Horizontal, Vertical, and Evolution consistency) and of the two UML consistency types (Semantic and Syntactic consistency). We then asked them the following questions:

1. Is there something that you would like to modify/improve in the wording?
2. Are there any aspects of a dimension or type that are not covered by a definition?
3. Would you please leave your comments in the boxes provided after the definitions?

In the second part of activity A1, we explained that during our systematic mapping study [14] we observed that the vast majority of UML consistency rules focus on the Horizontal dimension and Syntactic type, and very few rules are related to Vertical or Semantic consistency, while not a single rule covers the Evolution dimension. With this introduction we asked:

1. Does this suggest that dimensions (other than Horizontal) and types (other than Syntactic) of consistency are not relevant to UML or does it just suggest that rules are missing?
2. Would you please leave a comment? Feel free to present examples of UML consistency rules that cover dimensions and types of UML

consistency that are not very popular (such as Evolution, Vertical and Semantic consistency).

5.2 UML diagrams involved in UML consistency rules (A2)

In the first part of activity A2, we explained that the class diagram is the UML diagram most involved in UML consistency rules presented by researchers to date, followed in importance by the interaction diagram and the state machine diagram [14]. According to a recent study [20], the activity diagram is the second most frequently used UML diagram after the class diagram. We, however, found very few rules involving the activity diagram (we presented a list of 28 rules we found that involved the activity diagram [15]). We then asked:

1. Should research into UML consistency focus more on the activity diagram? What would additional consistency rules involving the activity diagram be?
2. Would you please leave your comment on this page? Feel free to present examples of new UML consistency rules that involve UML activity diagrams.

In the second part of activity A2, we showed that in our previous research [14] we did not find a single rule involving the package, profile, component, timing, interaction overview and deployment diagrams. We therefore asked:

1. According to your expertise, why did we obtain those results?
2. Should research into UML consistency focus more on these diagrams?
3. Would you please leave your comment below? Feel free to present examples of new UML consistency rules that involve the previously cited UML diagrams.

5.3 UML Consistency rules in MDD (A3)

In activity A3, we presented a questionnaire on each of the 190 rules collected [15]. Each rule was presented on a single piece of paper on which we described the rule with an example (unless we considered the rule trivial or easy to understand without an example). For each rule we asked the attendees the following questions:

1. Do you understand the rule?
    a. Yes;
    b. No.

    Would you like to re-phrase it? How?
2. Do you think this is a valid rule? Please check one of the following options and explain your decision:
    a. It is a valid rule and should be enforced in all UML models;
    b. It is a valid rule in some situations but not always;
    c. This rule should not be enforced;
    d. I am not sure.

    The expression "valid rule" refers to the fact that a specific rule could be relevant: always (a), in some situations (b), or never (c).
3. UML consistency rule complexity: the complexity of a UML consistency rule is directly related to the complexity of the consistency it attempts to specify. Complex UML consistency rules are generally difficult to understand, making it harder to detect such problems in a diagram or among diagrams.

    According to your expertise, how would you consider this rule? Please choose one of the following options and explain your decision:
    a. Very Complex;
    b. Complex;
    c. Normal Complexity;
    d. Simple;
    e. Very Simple;
    f. Other.

6. RESULTS

In this section, we describe only the analysis of the results obtained from the questions in activity A3 and the discussion generated on this activity. A limited number of people attended WUCOR (ten attendees), and we are therefore of the opinion that the results of this activity (A3), considering the high number of rules to check, are the only ones that may be considered a valid initial point for future improvements and replications for this research. Without a larger number of respondents, the results of activities A1 and A2 would only provide some personal opinions (of the ten participants) instead of general trends.

During activity A3, we were able to implement the questionnaire (presented in Section 5.3) for 81 out of the total of 190 (42.63%) UML consistency rules [15]. 73 of the 81 (90.12%) UML consistency rules were understood by the attendees. These 73 rules were then used as a basis to define a bubble plot in which to report the frequencies of combining the results from questions 2 and 3 of activity A3 (Section 5.3). A bubble plot is a two x–y scatter plots with bubbles in category intersections. This synthesis method is useful, since provides a map and a rapid overview of a research field [21]. After combining the results for questions 2 and 3 of activity A3, we obtained (Figure 1) the mapping of the complexity of the rules depending on whether or not the rules were considered valid.

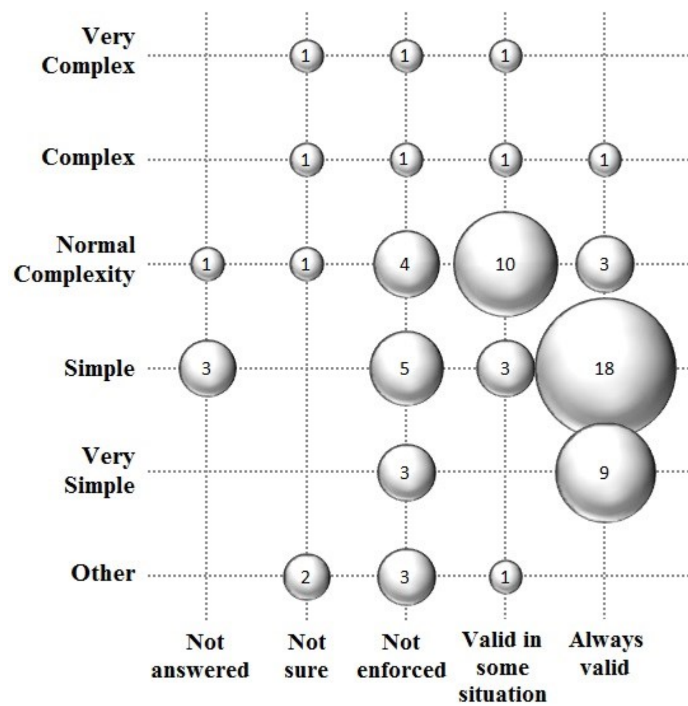

Figure 1 Combining question 1 and 2 of A3

Figure 1 shows that the attendees identified 18 (24.65%) simple rules as always valid. These rules were followed by another 10 (13.69%) with a normal complexity that are valid in only some situations. 56.16% (41 out of 73) of the rules checked were considered to be simple or very simple. Another important aspect that should be noted is that 47 of the 73 (64.38%) understood rules were positively considered as valid rules to be enforced (31 always valid, and 16 valid in some situations). The attendees, however, stated that only 23.28% (17 of 73) of the rules checked should not be enforced.

The reasons why eight of the 81 (9.87%) rules were not understood (question 1 of A3) by some of the attendees were:

- one attendee was not familiar with a particular concept presented in one rule;
- one attendee did not understand the specification of one rule and asked for an example scenario involving it;

- two attendees considered two rules to depend on a specific development methodology, which made them difficult to understand;
- one attendee considered the description of one rule to be incomplete;
- three attendees did not provide any justification for three rules, although they were explicitly asked to justify/explain their answers.

7. CONCLUSIONS

In recent years, researchers have put a great deal of effort into identifying UML consistency rules so as to detect inconsistencies between UML diagrams [14]. However, no previous workshop has been organized to discuss the state of the art in terms of UML consistency rules. WUCOR 2015 has been the first workshop entirely focused on UML consistency rules to gather feedback from the community, either industry or academia, on this topic. The format of WUCOR [16] was a mix of two presentations, three activities developed with working groups, and a final activity dedicated to discussions. We had two presentations from authors who reported their studies on distinct software engineering subjects using different research methods (see Section 4 for more details). The two presentations were followed by three activities (see Section 5 for activities and Section 6 for results), and finally the last section of WUCOR consisted of a discussion on the three activities during which the participants interacted with each other.

We plan to hold future versions of WUCOR, with the long-term goal of creating an active research community with a focus on consolidating the body of knowledge on UML consistency rules. We hope that the results obtained will lead to an improvement in the software engineering practices related to Model Driven Development.

8. ACKNOWLEDGMENTS

We would most sincerely like to thank the authors of the papers submitted to WUCOR 2015; the Program Committee (PC) members for reviewing these papers; the MODELS 2015 organizers for their support when organizing the workshop; the presenters of the two papers accepted; and the workshop participants without whom the activities would not have taken place. For more information about WUCOR PC members and authors, please visit the workshop website at https://wucor.wordpress.com.

This work has been funded by the SIGMA-CC project (Ministerio de Economía y Competitividad and FondoEuropeo de Desarrollo Regional FEDER, TIN2012-36904) and an NSERC discovery grant.

9. REFERENCES

[1] Mukerji, J. and J. Miller, 2003. Overview and guide to OMG's architecture. http://www.omg.org/mda/ (last accessed Dec. 2015).

[2] Thomas, D., 2004. MDA: Revenge of the modelers or UML utopia? *IEEE Software* 21, 3, 15–17. DOI= http://dx.doi.org/10.1109/MS.2004.1293067.

[3] Lucas, F.J., F. Molina, and A. Toval, 2009. A systematic review of UML model consistency management. *Information and Software Technology* 51, 12, 1631-1645. DOI= http://dx.doi.org/10.1016/j.infsof.2009.04.009.

[4] Genero, M., A.M. Fernández-Saez, H.J. Nelson, G. Poels, and M. Piattini, 2011. A Systematic Literature Review on the Quality of UML Models. *Journal of Database Management* 22, 3 (July-September 2011), 46-70.

[5] Usman, M., A. Nadeem, K. Tai-hoon, and C. Eun-suk, 2008. A Survey of Consistency Checking Techniques for UML Models. In *Proceedings of the Conference on Advanced Software Engineering and Its Applications* (Hainan Island, China, December 13-15 2008), IEEE Computer Society, 57-62. DOI= http://dx.doi.org/10.1109/asea.2008.40.

[6] OMG, 2015. OMG Unified Modeling Language Version 2.5. Object Management Group

[7] Pender, T., 2003. UML Bible (1st ed.). John Wiley & Sons, Inc., New York, NY, USA.

[8] Petre, M., 2013. UML in practice. In *Proceedings of the 35th International Conference on Software Engineering* (San Francisco, CA, USA, May 18-26 2013), IEEE Press, 722-731.

[9] Ibrahim, N., R. Ibrahim, M.Z. Saringat, D. Mansor, and T. Herawan, 2011. Consistency rules between UML use case and activity diagrams using logical approach. *International Journal of Software Engineering and its Applications* 5, 3, 119-134.

[10] Simmonds, J., R.V. Straeten, V. Jonkers, and T. Mens, 2004. Maintaining Consistency between UML Models using Description Logic. *RSTI – L'Object LMO'04* 10, 2-3, 231-244.

[11] Huzar, Z., L. Kuzniarz, G. Reggio, and J.L. Sourrouille, 2005. Consistency problems in UML-based software development. In *Proceedings of the International Conference on UML Modeling Languages and Applications* (Lisbon, Portugal 2005), Springer-Verlag, 1-12. DOI= http://dx.doi.org/10.1007/978-3-540-31797-5_1.

[12] Muskens, J., R.J. Bril, and M.R.V. Chaudron, 2005. Generalizing Consistency Checking between Software Views. In *Proceedings the 5th Working IEEE/IFIP Conference on Software Architecture* (Pittsburgh, Pennsylvania, USA, November 6-10 2005), IEEE Computer Society, 169-180. DOI= http://dx.doi.org/10.1109/wicsa.2005.37.

[13] Spanoudakis, G. and A. Zisman, 2001. Inconsistency management in software engineering: Survey and open research issues. In Handbook of Software Engineering and Knowledge Engineering, S.K. Chang Ed. World Scientific Publ. Co., Singapore, 329-380.

[14] Torre, D., Y. Labiche, and M. Genero, 2014. UML consistency rules: a systematic mapping study. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE 2014)* (London, UK, 13-14 May 2014), ACM.

[15] Torre, D., Y. Labiche, M. Genero, and M. Elaasar, 2015. A systematic identification of consistency rules for UML diagrams. Carleton University. http://goo.gl/TFMgnE.

[16] Torre, D., Y. Labiche, M. Genero, and M. Elaasar, 2015. Introduction to WUCOR 2015. In *Proceedings of the 1st International Workshop on UML Consistency Rules (WUCOR 2015) co-located with ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2015)* (Ottawa 2015), CEUR. http://ceur-ws.org/Vol-1508/.

[17] Hoisl, B. and S. Sobernig, 2015. Consistency Rules for UML-based Domain-specific Language Models: A Literature Review. In *Proceedings of the 1st International Workshop on UML Consistency Rules (WUCOR 2015) co-located with ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2015)* (Ottawa, Canada, September 28 2015), CEUR.

[18] Sobernig, S., B. Hoisl, and M. Strembeck, 2016. Extracting reusable design decisions for UML-based domain-specific languages: A multi-method study. *Journal of Systems and Software* 113, 140-172.

[19] Chiorean, D., V. Petraşcu, and I. Chiorean, 2015. Proposal for Improving the UML Abstract Syntax. In *Proceedings of the 1st International Workshop on UML Consistency Rules (WUCOR 2015) co-located with ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2015)* (Ottawa, Canada, September 28 2015), CEUR.

[20] Reggio, G., M. Leotta, F. Ricca, and D. Clerissi, 2013. What are the used UML diagrams? A Preliminary Survey. In *Proceedings of the 3rd International Workshop on Experiences and Empirical Studies in Software Modeling - CEUR Workshop Proceedings (EESSMod 2013)* (Miami, Florida - USA, October 1 2013), 3-12.

[21] Petersen, K., R. Feldt, S. Mujtaba, and M. Mattsson, 2008. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE 2008)* (Bari, Italy 2008), British Computer Society, 71–80.