

Report about about the Workshop on Use Case Modeling at UML-2004

“Open Issues in Industrial Use Case Modeling”

Gonzalo Génova, Carlos III University of Madrid

Juan Llorens, Carlos III University of Madrid

Pierre Metz, Cork Institute of Technology, Ireland

Rubén Prieto-Díaz, James Madison University, VA, USA

Hernán Astudillo, Universidad Técnica Federico Santa María, Chile

Abstract

Use Cases have achieved wide use as a specification tool for observable behavior of systems. However, there is still much controversy, inconsistent use, and free-flowing interpretations of use case models, in fact, not even experts widely recognized in the community agree on the meaning of concepts. Consequently, use case models are dangerously ambiguous, and there is an unnecessary divergence of practice. The purpose of the workshop was to identify and characterize some sources of ambiguity. It gathered specialists from academia and industry involved in modeling use cases to exchange ideas and proposals, with an eye to both clear definition and practical application. Some presented topics were discussed in-depth (the UML metamodel for use cases, use case instances, use cases in MDD/MDA, use case model vs. conceptual model, and tools for use cases specification), while others were left as open issues for future research. We hope our suggestions will be useful to improve the understanding of use cases, and stimulate further research to reach a stronger coupling between the use case model and other static, behavioral and architectural models.

1 MOTIVATION AND GOALS

In UML there are two main representations for use cases: *textual specifications* and *diagrams*. From a methodological standpoint, these “two worlds” have been evolving in isolation to each other. A full semantic connection between use case specification items and UML use case diagrams as initially desired by Jacobson et al. in the OOSE method does not exist. This important topic is still open to discussion and agreements, and the original impetus of the workshop comes from this dichotomy between textual vs. graphical representations for use cases.

The Graphical World

UML has been aiming to formalize use cases through object-oriented techniques by declaring the metamodel element UseCase as a subtype of Classifier, which contains Attributes, Operations and Methods, while not defining use case documentation properties or providing a tailorable use case template. This has given use cases an explicit OO formalization as desired by Jacobson et al. in OOSE. However, the absence of a reconciling explanation of this formalization with textual use case specifications as promoted by the literature, and of guidance on how to actually document use cases, has caused a certain lack of understanding among both practitioners and researchers.

The graphical world of diagrams is dominated by use case relationships. UML's explanations of Include and, in particular, of Extend remain vague, and may even seem to be contradictory, tending to confuse readers about when to use Include or Extend. Precise and unambiguous definitions of terms are missing. Therefore, UML's explanations for Include and Extend are still subject to ongoing debates. Moreover, UML appears to mix the instance and the type view when defining the use case relationships Include and Extend.

Another aspect where UML fails to be fully clear is regarding the meaning of use case *Generalization*: there is no indication whether subtyping and/or inheritance are meant.

The Textual World

The literature commonly emphasizes and promotes written use case specifications for functional requirements capture, which are organized according to a template; there is an implicit commitment to what a use case template should include. In contrast, use case diagrams have been used merely as an adequate graphical view on, or "entry point" to, these written specifications.

Practitioners and experts in the community frequently warn against over-emphasizing use case diagrams and strenuously advise never to neglect the use case textual specifications: in practice a use case diagram serves as a support for text but not vice versa ("a bubble does not tell us the story"). Furthermore, the techniques in the textual world are much more expressive and powerful compared to the use case relationship capabilities in UML. Finally, UML does not provide graphical modeling means for many aspects used in the textual world such as linking use cases through pre- and post-condition relations.

The current literature avoids making any commitment and prefers to highlight UML's current elusive use case relationship semantics, add to these semantics, or even arbitrarily modify these semantics, thereby keeping the practical use case concepts fuzzy. Some authors even fully discourage the use of particular use case relationships, or recommend getting rid of variety and having only a single but powerful use case relationship.



2 OPEN AREAS FOR RESEARCH

Some of the open areas identified before the workshop were:

- Alignment of textual specification and graphical representation: use case relationships, use case standard templates, use case contracts, any information missing or extra in the two representations.
- Little semantic connection between use case specification items and UML use case diagrams. In particular, UML lacks support for the connection proposed by Jacobson et al. in OOSE.
- Collaboration vs. participation among actors of a use case. Actors may have a collaborative or participatory role in a use case, yet UML diagrams do not allow distinguishing them.
- Functional vs. structural view of use cases. Use cases may be expanded to represent functional characteristics of parts of systems, yet this expansion is not possible in UML's graphical view.
- Relationships among use cases, composition: UML allows include and extend, yet composition has a different semantics; some meaningful relationships could be borrowed from other notations, such as "precedes" from OPEN/OML and "mitigates" from Misuse Cases; dependency information encoded in pre- and post-conditions cannot be depicted graphically either.

3 ORGANIZATION

The workshop was organized by Gonzalo Génova (Carlos III University of Madrid, Spain), Juan Llorens (Carlos III University of Madrid, Spain), Pierre Metz (Cork Institute of Technology, Ireland), Rubén Prieto-Díaz (James Madison University, VA, USA) and Hernán Astudillo (Universidad Técnica Federico Santa María, Chile).

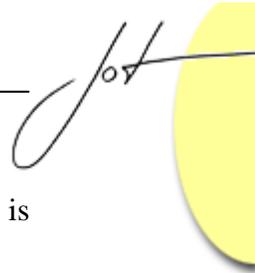
Submitted papers were reviewed by an international team of experts composed by the organizers and Shane Sendall (University of Geneva, Switzerland), Roderick Coleman (Free consultant, Germany), Wolfgang Weber (University of Applied Sciences, Darmstadt, Germany), Sadahiro Isoda (Toyohashi University of Technology, Japan), Joaquin Miller (X-Change Technologies, USA), Guy Genilloud (IT/business consultant, Switzerland), Paul Bramble (Independent consultant, USA) and Bruce Anderson (Managing Consultant, Application Innovation, IBM Business Consulting Services, UK).

Each paper received between 2 and 4 reviews before being accepted for presentation at the workshop. Revised versions of those papers are published now in this special issue.

4 INITIAL POSITIONS OF THE AUTHORS

The initial two sessions of the workshop were devoted to presentation of the accepted papers, which represented a good mixture of experiences and researches both from academia and industry, as was one of the goals of the workshop. The authors came to the workshop with the following positions:

- Bruce Anderson. Use case modelling is widely used as a technique for requirements gathering but does not always lead to clear agreement between users and developers, or to effective system development. This is often because the model does not have a clear role in a clear process, with a corresponding lack of agreed standards and techniques. Taking a considered approach and tailoring the available guidance to the situation at hand can produce more appropriate use cases that are more useful in the overall process. This paper outlines a sound approach in a context of ideas and technique, and discusses several common issues in use case modelling, with suggested resolutions.
- Nelly Bencomo, Alfredo Matteo. The software life cycle of Distributed Object applications spans requirements specification to design and implementation. Support for traceability has been established as an important task in the life cycle. Concepts in analysis and design should have a clear correspondence to implementation artifacts. Our article describes artifacts associated with Use Case, Analysis, Design, Implementation and Deployment models when developing Distributed Object applications. The work proposes a clear traceability from Analysis to Implementation and Deployment models based on the use cases approach. An example involving web access to bank accounts is presented.
- Clay Williams, Matthew Kaplan, Tim Klinger, and Amit Paradkar. We argue that use case modeling should be done in the context of a rich conceptual model. Use cases are written in terms of this model using structured natural language. We also discuss problems that arise when trying to align this representation with the UML 2.0 metamodel, including metaclass misalignment and the lack of a representation for use case content. We close by discussing four applications of our representation: prototyping, estimation, refinement to design, and test case creation
- Michal Smialek. Use cases should have precisely defined notations which are comprehensible by various groups of people in a software development project. In order to meet these diverse views, several notations are necessary. These notations should be easily transformable and should have clear mappings to other models including the conceptual model.
- Sadahiro Isoda. The current UML's use-case specification has a lot of problems and even nonsense. All these problems are due to three fundamental defects originated in OOSE. These are the illusionally "actors call use cases" conjecture, mixing-up designer's simulation with real execution and poor understanding of



-
- OO. The problems can be easily solved by recognizing anew what a use case is and then modeling it guided by plain OO technology.
- Gonzalo Génova, Juan Llorens. In UML, use cases are meta-modeled as classifiers. Classifiers specify a set of instances, and use case instances are said to be concrete system-actor interactions. But it is not clear how an interaction can have classifier features such as attributes, operations and associations. Therefore, we challenge the notion that use case instances are interactions. We also propose a notion of use case (a coordinated use of system operations) that is very close to the traditional protocol, therefore concluding that use cases and protocols are not essentially different things.
 - Guy Genilloud, William F. Frank. The UML ontology is unnatural and limited (at odds with the categories of thought people use for engineering in natural languages such as Japanese and in mathematics). As a consequence, the UML standard confuses use case specifications, types, and instances, as well as confusing a use case model with what it is a model of. The Extends relationship illustrates these problems. ISO's RM-ODP provides a richer ontology based on logical theory. ODP explains Extends as a relationship between specifications, while opening the door for relationships between the actions so specified, and reconciling diagrammatic and textual use case techniques.
 - Joaquin Miller. I suggest we take an indirect approach to finding techniques to specify use cases using UML: look at use cases from the ODP viewpoint; choose ODP concepts well suited to specifying a use case; find corresponding UML constructs; adapt the UML constructs as required. I arrive at: A particular use case of a certain system is a part of the community contract of a community of a certain type. That community is represented as a UML collaboration. I discuss how that community can be specified using UML.

5 WORKSHOP RESULTS

The remaining two sessions of the workshop were devoted to discussions and synthesis work, trying to reach agreement wherever it was possible. We first established a list of open issues and related them to the presented papers. Then the issues related to two or more papers were discussed in-depth: the UML metamodel for use cases [Williams, Isoda, Génova, Genilloud, Miller], use case instances [Isoda, Génova, Genilloud], use cases in MDD/MDA [Anderson, Bencomo, Smialek], use case model vs. conceptual model [Williams, Smialek], and tools for use cases specification [Smialek, Isoda]. Other issues related to only one paper, or not particularly related to any of the papers, were not specifically discussed, due to lack of time.

A full account of workshop discussions can be found in [1].

6 CONCLUSIONS AND FUTURE WORK

The summary of workshop discussions [1] shows that our main interests lie all around two main poles: a) problems posed by deficiencies in the UML2 Specification regarding use cases, i.e. *semantics* of use cases; and b) the use case model in the context of MDD/MDA, i.e. *pragmatics* of use cases. We hope our suggestions will be useful to improve the understanding of use cases, and stimulate further research to reach a stronger coupling between the use case model and other static, behavioral and architectural models. These two issues are closely related too, i.e. the quality of the UML Specification has a practical impact: a better explanation of the intended meaning of use case concepts is required to achieve a wider consensus about good practices in writing use cases and to build tools that can effectively support a use case driven development (MDD/MDA); also, UML2 should not ignore well-established industrial practices.

The workshop discussions were extremely participative and fruitful. This workshop has been the first in a series of Workshops on Use Case Modeling (WUsCaM) that will be continued at future UML Conferences (from now on called MoDELS Conference).

More information can be found on the workshop web sites [2, 3].

REFERENCES

- [1] Gonzalo Génova, Juan Llorens, Pierre Metz, Rubén Prieto-Díaz, Hernán Astudillo. “Open Issues in Industrial Use Case Modeling”. *The 7th International Conference on the Unified Modeling Language-UML'2004 Satellite Activities*. Lisbon, Portugal, October 11-15, 2004. *Lecture Notes in Computer Science 3297*, Springer 2004.
- [2] I International Workshop on Use Case Modeling, “Open Issues in Industrial Use Case Modeling”, <http://www.ie.inf.uc3m.es/uml2004-ws6/>.
- [3] II International Workshop on Use Case Modeling, “Use Cases in Model-Driven Software Engineering”, <http://www.ie.inf.uc3m.es/wuscam-05/>.



About the authors



Gonzalo Génova received in 2003 his PhD in Computer Science at the Carlos III University of Madrid, Spain, where he is currently an Associate Professor of Software Engineering. His main research subject is modeling and modeling languages in model-driven software engineering. He can be reached at ggenova@inf.uc3m.es.



Juan Llorens is a Professor of the Computer Science Department at the Carlos III University of Madrid, Spain, where he is the leader of the IE (Information Engineering) research group. He is also a Visiting Professor at Aland's Institute of Technology - ATL, Mariehamn, Finland. His current research involves the integration of Knowledge technologies and Software Engineering techniques towards Software and Information Reuse. He can be reached at llorens@inf.uc3m.es.



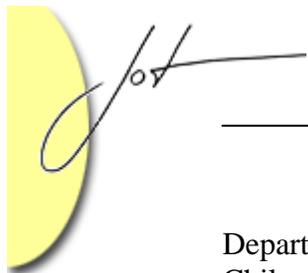
Pierre Metz received his PhD in Computer Science in 2005 from Cork Institute of Technology, Ireland and HETAC, a statutory board. Pierre is working in the SW industry in various domains such as defence, automotive, banking, and telecommunications as an ISO/IEC 15504-SPiCE assessor, CMMI and IEC EN DIN 61508 (functional safety) exponent, requirements engineering & management and object-oriented technology specialist. His research interests reflect the latter. He is also a part-time lecturer at Darmstadt University of Applied Sciences, Computing Dept. Pierre can be contacted at pmetz@cit.ie or pm@synspace.com.



Rubén Prieto-Díaz earned his PhD in information and computer science from University of California, Irvine, 1985. His areas of expertise include software reusability, reuse libraries, domain analysis, information modeling and software architecture. His current research interests include security requirements, Common Criteria evaluation process and trustworthy architectures. Dr. Prieto-Díaz is Associate Director for Information Assurance at the Institute for Infrastructure and Information Assurance (IIIA) and Associate Professor of Computer Science both at James Madison University in VA, USA. He can be reached at prietodiaz@cisat.jmu.edu.



Hernán Astudillo got his Ph.D. in Information and Computer Science from Georgia Institute of Technology in 1996. He worked several years as applications architect for systems integrators and consulting companies, and is currently full-time professor at the Informatics



Department of Universidad Técnica Federico Santa María (UTFSM), in Valparaíso, Chile. His research and practice interests include software architecture modeling, derivation and reuse, COTS, and middleware. He can be reached at hernan@acm.org.