A Report on the First International Workshop on Best Practices of UML (BP-UML'05)

Juan Trujillo
Dept. of Language and Information Systems
University of Alicante, Spain
Apto. Correos 99. E-03080
jtrujillo@dlsi.ua.es

1. Introduction

The Unified Modeling Language (UML) has been widely accepted as the standard object-oriented (OO) modeling language for modeling various aspects of software and information systems. The UML is an extensible language, in the sense that it provides mechanisms to introduce new elements for specific domains if necessary, such as web applications, database applications, business modeling, software development processes, data warehouses and so on. Furthermore, the latest work of the Object Management Group (OMG) on UML [1] resulted in a larger and more complicated specification, with even more diagrams for some good reasons. Although providing different diagrams for modeling specific parts of a software system, not all of them need to be applied in most cases. Therefore, heuristics, design guidelines, and lessons learned from experiences are extremely important for the effective use of UML and to avoid unnecessary complication.

This report focuses on the First International Workshop on Best Practices of UML (BP-UML'05) held in conjunction with the 24th International Conference on Conceptual Modeling (ER'05) in Klagenfurt, Austria, on October 24th-28th, 2005. A summary of the accepted papers is given.

In the call for papers, papers focused on the application of the UML in new domains were especially encouraged. In response to the call for papers, the workshop received 25 submissions and only 9 papers were selected by the Program Committee, making an acceptance rate of 36%.

The accepted papers were organized in three different sessions: (i) Experience reports and new applications, (ii) Model evaluation and requirement's modeling, and (iii) Metamodeling and Model Driven Development. In the first one, two papers present valuable experience reports and another one describes how to apply UML for multidimedia modeling. In the second one, one paper is focused on evaluating the cardinality interpretation by users in a UML class diagram, and the other two papers are focused on the Use case diagrams of the UML. Finally, in the third session, while one paper presents how to analyze the consistency of a UML diagram, the other two are focused on the Model Driven Architecture (MDA) and metamodeling. The workshop proceedings are published in [11].

2. Experience reports and new applications

B. Dobing and J. Parsons [2] argue that although many research papers and text books have been published on the different aspects of UML, works on the practical use of UML are absolutely missing. In this paper, authors report results of a survey of UML use by practitioners. Authors developed the survey through the Web and based on a literature review and preliminary interviews with about a dozen practitioners. The Object Management Group (OMG) supported this survey and most practitioners were associated with the OMG. Results indicate varying levels of use, and perceived usefulness, of different UML diagrams such as Use Case, Activity, Sequence, Class, Collaboration and Statechart Diagrams. The reported survey received 299 usable responses, which either contained data on UML component usage (182) or reasons why the UML was not being used (117). Of the 182 analysts using UML components, most (171) were using the UML while 11 were using several UML components as part of another methodology. Another conclusive aspect that the authors argue is that UML is also used by non-IT professional, and therefore, UML diagrams should be more readable and easy to understand.

J.A. Cruz et al. [3] present an interesting approach on the understandability of UML statechart diagrams, and in particular, on how the use of Composite states affects the understandability of these diagrams. To this aim, authors define a new metric named the Nesting Level in Composite States (NLCS) which indicates the maximum number of nested composite states in a UML statechart diagram. Then, the authors focus on describing the experimental process accomplished in order to check the empirical validation of the proposed metric. Unfortunately, the obtained results were not highly conclusive and the authors have not been able to find an optimal use of nesting within UML statechart diagrams and they can only partially conclude that a flat nesting level (0 or 1) within a relatively simple UML statechart diagram makes it more

understandable. Obviously, further empirical research is needed, considering more complex UML statechart diagrams.

T. Ignatova and I. Bruder [4] propose a UML framework to derive applications-specific multimedia database models. The authors mainly focus on describing their framework, which allows us to define the core elements of a multimedia database model, such as mediatype - and application- independent structure, content, relationships and operations. Then, the authors also discuss the advantages of using UML for representing multimedia data as well as shortcomings of this approach that should be covered in the future. Also, they describe the utilization of their UML framework for the instantiation of a model for an image database of scanned handwritten music scores. Finally, the authors showed the advantages of the framework, such as the facilitated design and maintenance of the application, and the seamless integration with other applications.

3. Model evaluation and requirement's modeling

G. Poels et al. [5] present an empirical study on the many-to-many relationships with attributes in Class diagrams. Firstly, the authors provide related work and discussions on the pros and cons of objectifying *many-to-many* relationships in Class diagrams. Then, the authors describe an experiment in order to check if the representation chosen for a relationship with attributes affects the ability of model users to understand the information conveyed by a UML class diagram. The authors employed two pairs of class diagrams representing two structural models including one *many-to-many* relationship with attributes. The results presented in the paper indicate that, controlling for cardinality knowledge, business users can better interpret the information that a UML class diagram conveys about a *many-to-many* relationship with attributes if this relationship is represented as an association class instead of an object class. Finally, the authors argue that the implication for establishing 'best practices' in UML modeling is that modelers should refrain from objectifying such relationships if the goal is an effective communication of domain semantics to business users, who are not UML or modeling experts.

J. Goldman and I-Y. Song's paper [6] argues for the need of a structure or framework for organizing a large number of use cases that may be required for modeling complex systems. Thus, the authors start by analyzing five existing use case classification schemas from existing literature. Then, they propose a new additional classification schema based on the system functionalities for classifying and organizing use cases. The authors also propose a straightforward methodology, resting on sequentially answering some simple questions, to determine use case categories to aid analyzers in real-world projects. In order to illustrate the proposed method, the authors present an exercise conducted in a classroom with 31 graduate students in an introductory UML course. Finally, the authors discuss how we can effectively understand categorized use cases in terms of project priority and personnel skills to achieve the best possible allocation of project resources to use case-driven development efforts.

M. Hilsbos et al. [6] present a comparative analysis of the use case relationships discussed in eleven literatures, including the UML 2.0 specification. First of all, and due to the different terms used in the referred literature, the authors provide a common terminology in order to correctly compare and analyze the related work. Then, the authors provide an extensive literature review and present the agreed usages and different proposed view points of the use case relationships, and argue for a logical resolution for each proposal. As a coherent approach for applying use case relationships, the authors proposed three rules derived from the review of the literature and their own experience and illustrate the rules with examples. Their rules are based on the analysis of preconditions, postconditions of use cases, and characteristics of the behaviors being separated. Finally, from the provided analysis, the authors mainly conclude that practitioners should be aware of the nuances of appropriate application of each use case relationship, apply the relationships sparingly, and, when in doubt, develop several alternative models for complex problems.

4. Metamodeling and Model Driven Development

S. Meliá and J. Gomez [7] propose a generic approach called WebSA (Web System Architecture), based on the Model Driven Architecture (MDA) paradigm to design Web applications. Authors start by providing an overview of the WebSA development process and the modeling notation. Their approach is made up of a set of UML architectural models and QVT (Query/View/Transformations) transformations as mechanisms to integrate the functional aspects of the current methodologies with the architectural aspects. In order to illustrate their proposal, the authors use their WebSA approach to tackle the design of the well known J2EE Petstore specification, showing how to integrate functional and architectural aspects in the design of Web applications. Then, the authors explain the QVT transformations showing how traditional Web functional models and the Configuration model can be merged into an Integration model. Finally, the authors provide an overview of this Integration model.

F.J. Lucas and A. Toval [8] present a rigorous approach to improve the consistency analysis between UML diagrams. To start with, the authors provide a summary of the algebraic formalization of part of the UML metamodel on which their rigorous approach is based. The authors

argue that their framework helps to guarantee the consistency of models because all the specifications are integrated within the same formalism. Then, the authors show the applicability of their approach by verifying the consistency between Class Diagrams and Communication Diagrams. Finally, the authors focus on verifying two properties: (i) a syntactic verification through associations, and (ii) a type consistency of the parameters in the calls of methods.

B. List and B. Korherr [9] propose a UML 2 Profile for Business Process Modelling (BPM). The authors start by discussing the main requirements that a Business Process model should capture. Then, they describe the meta-model as the basis for the proposed UML 2 profile. This meta-model allows designers to consider two complementary perspectives: (i) the business perspective and (ii) the sequence perspective. The sequence perspective refines the business perspective and describes the detailed flow of the process. The business perspective presents the business process from a wide angle by integrating aspects like goals, customers, deliverables, process types etc. Then, the authors provide the specification of the proposed UML 2 profile by defining the required stereotypes, tagged values and constraints. Finally, in order to demonstrate the practical applicability of the business perspective of the UML 2 profile for BPM, the authors apply their profile to *Processing of Claims* business process of an insurance company.

5. Conclusions / Summary

BP-UML'05 was organized on the basis that although UML provides different diagrams for modeling specific parts of a software system, not all of them need to be applied in most cases. Furthermore, due to the considerable number of different diagrams that can be used for modeling the different aspects of a software system, many inconsistencies may appear between the different used UML diagrams. In this workshop, some experimental works were presented in order to help us understand where and when to use the different UML diagrams. Other works showed us how to correctly use the Use Case diagrams and to avoid the inconsistency between different UML diagrams. Finally, another group of papers showed how to apply and extend UML to new applications such as Multimedia, Web or Business Process modeling. In other words, the workshop was a valuable forum where UML researchers will find interesting papers in order to improve the way we can apply UML to real world projects.

Thanks to the number of submissions of this first edition (25) together with the high quality of the accepted papers and the low acceptance rate (36%), it is my pleasure to announce that the second edition of BP-UML is held together with ER2006. My intention is to keep organizing this workshop several years as there is a wide agreement in the UML research community that we still need more and best practices of UML in order to correctly use and apply UML.

6. Acknowledgments

I would like to express my gratitude to the Program Committee (PC) members and the additional external referees for their hard work in reviewing papers. In order to keep the high quality of former workshops held in conjunction with ER, a strong International PC was organized with extensive experience in the UML and their relevant scientific production in the area. I also thank all the authors for submitting their papers and the ER2005 organizing committee for all their support. Another gratitude is for our technician Miguel A. Varo, who developed the web site and the review system (http://gplsi.dlsi.ua.es/congresos/bpuml05/). This workshop was organized within the framework of the following projects: MESSENGER (PCC-03-003-2), METASIGN (TIN2004-00779), DADASMECA (GV05/ 220) and DADS (PBC-05-012-2).

7. References

[1] OMG. UML 2.0 Superstructure and Infrastructure Specifications. http://www.uml.org/#UML2.0. 2005

[2] B. Dobing, J. Parsons. "Current Practics in the Use of UML", in [11].

[3] J. A. Cruz-Lemus et al. "An Empirical Study of the Nesting Level of Composite States within UML Statechart Diagrams", in [11].

[4] T. Ignatova. "Utilizing a Multimedia UML Framework for an Image Database Application", in [11].

[5] G. Poels et al. "Object Class or Association Class? Testing the User Effect on Cardinality Interpretation", in [11].

[6] J. L. Goldman, I-Y. Song. "Organizing and Managing Use Cases", in [11].

[7] M. Hilsbos et al. "A Comparative Analysis of Use Case Relationships", in [11].

[8] S. Melia, J. Gomez. "Applying transformations to Model Driven Development of Web applications", in [11].

[9] F. Lucas, A. Toval. "A Precise Approach for the Analysis of the UML Models Consistency", in [11].

[10] B. List, B. Korherr. "A UML 2 Profile for Business Process Modelling", in [11].

[11] J. Akoka, et al. (eds.). "ER Workshops 2005", Lectures Notes in Computer Science, vol. 3770, pp. 1-96, 2005, Springer-Verlag, Berlin Heidelberg 2005.