

Contents

	Page
Guest Editorial	5

INTRODUCTION	Page
--------------	------

Report about the Workshop on Use Case Modeling at UML-2004 “Open Issues in Industrial Use Case Modeling”	7
<i>By Gonzalo Génova, Juan Llorens, Pierre Metz, Rubén Prieto-Díaz, and Hernán Astudillo</i>	

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 semantics of concepts. Consequently, use case models are dangerously ambiguous, and there is an unnecessary divergence of practice.

ARTICLES	Page
----------	------

Formalism, technique and rigour in Use Case Modelling	15
<i>By Bruce Anderson</i>	

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.

Traceability Management through Use Cases when Developing Distributed Object Applications 29

By Nelly Bencomo, and 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.

Toward Engineered, Useful Use Cases 45

By 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.

Accommodating Informality with Necessary Precision in Use Case Scenarios 59

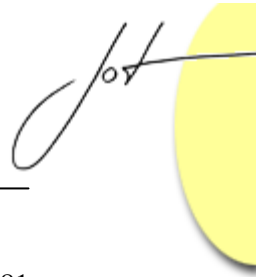
By Michał Śmiałek

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.

On UML2.0's Abandonment of the Actors-Call-Use-Cases Conjecture 69

By 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.

The Emperor's New Use Case

81

By Gonzalo Génova, and 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.

Use Case Concepts using a Clear, Consistent, Concise Ontology

95

By Guy Genilloud, and 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.

Use Case from the ODP Viewpoint

109

By 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.

OUTLOOK

Page

A brief outlook to the next issue

117