

Contents

Page

Editorial

5

COLUMNS

Guest Column

UML - Unified or Universal Modeling Language? UML2, OCL, MOF, EDOC - The Emperor Has Too Many Clothes

7

Dave Thomas

We have no successful example in which language design by committee results in a simple clean language with a clear semantic account. We do have many examples of committee approaches to language design and standardization which produce complex, bloated languages that reflect the compromise of committee lobbying to reach consensus. UML2, OCL and Action semantics coupled with the MOF seem destined to be sucked into yet another committee attempt to unify the world in a single grand language – the vain quest for a “computer Esperanto”.

Classification Theory

The Theory of Classification, Part 5: Axioms, Assertions and Subtyping

13

Anthony J.H. Simons

We consider the exact specification of a type's *behaviour* and also the relationship between algebraic specification and sub-typing, which will allow us to prove when one object *behaves in a subtype-conformant way* to another.

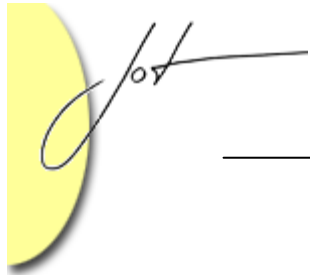
Business Objects

Hello World! Goodbye Skills!

23

Mahesh H. Dodani

Building skills is difficult. Building “do level” skills that can be applied on complex application development projects are even more difficult. Building skills that establishes the foundation for continual change and enhancement is extremely difficult.

**Cyber Databases****Knowledge Management: A Careful Look** 29*Won Kim and Seung-Soo Park*

The term “knowledge management” is vaguely understood. One reason is that the term “knowledge” itself has been both overused and loosely used. To understand the technical issues and challenges in knowledge management, one must first understand the term knowledge better.

Objects and Agents**The Role of Roles** 39*James Odell, H. Van Dyke Parunak, and Mitchell Fleischer*

The notion of *role* is fundamentally a thespian concept, and attention to how it functions in the theater can reinforce our intuitions and provide useful metaphors for application to multiagent systems.

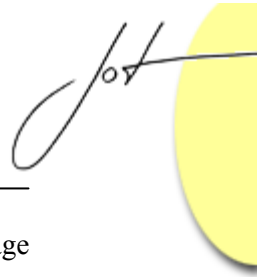
OO Requirements Engineering**Engineering Security Requirements** 53*Donald Firesmith*

Most requirements engineers are poorly trained to elicit, analyze, and specify security requirements, often confusing them with the architectural security mechanisms that are traditionally used to fulfill them. They thus end up specifying architecture and design constraints rather than true security requirements.

REFEREED ARTICLES

How You Could Use NEPTUNE in the Modelling Process 69*Agusti Canals, Yannick Cassaing, Antoine Jammes, Laurent Pomiès, and Etienne Roblet*

The main objective of the European NEPTUNE project is to develop both a method and tools supporting the use of the UML notation. NEPTUNE will apply to a variety of application fields, including software engineering, business process and knowledge management.



| | Page |
|---|------|
| An Object-Oriented Framework for Building Software Agents <i>José Alberto Rodrigues Pereira Sardinha, Paula Clark Ribeiro, Ruy Luiz Milidiú, and Carlos José Pereira de Lucena</i> | 85 |
| Agent technology is a new approach of Distributed Artificial Intelligence to implement autonomous entities driven by beliefs, goals, capabilities and plans, and other agency properties such as adaptation, interaction, and mobility. | |
| Sending Messages in UML <i>Gonzalo Génova, Juan Llorens and Vicente Palacios</i> | 99 |
| During the last decade there has been an intense controversy since James Rumbaugh introduced a strong concept of association derived from entity-relationship models. In this approach, associations should be regarded as first-class semantic constructs of similar weight to classes and generalizations, because classes and associations abstract jointly, and in a natural way, not only the high-level static structure of the system, but also the overall structure of interactions between objects. | |
| <hr/> BOOK REVIEWS <hr/> | |
| Design by Contract, by Example , by Richard Mitchell and Jim McKim. Reviewed by Charles Ashbacher | 117 |
| Facts and Fallacies of Software Engineering , by Robert L. Glass Reviewed by Charles Ashbacher | 119 |
| Peer Reviews in Software: A Practical Guide , by Karl E. Wiegers Reviewed by Charles Ashbacher | 121 |
| My Best Books of the Year 2002 An overview by Charles Ashbacher | 123 |
| <hr/> OUTLOOK <hr/> | |
| A brief outlook to the next issue | 127 |