

## Contents

	Page
<b>Editorial</b>	5

---

### COLUMNS

---

#### Guest Column: OOA

<b>Use Cases and Aspects—Working Seamlessly Together</b>	7
<i>By Ivar Jacobson</i>	

Aspect oriented programming (AOP) is “the missing link” to allow you slice a system, use case by use case, over “all” lifecycle models. This will dramatically change the way complex systems are understood, how new features are added to systems, and how systems are implemented and tested. AOP will also add a new dimension of reuse to software development. And it is here to be harvested—now.

#### Guest Column: OODBMS

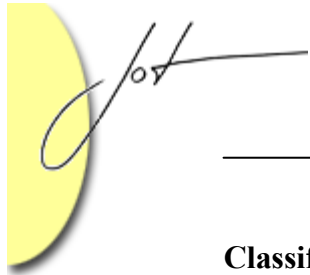
<b>Achievements and Weaknesses of Object-Oriented Databases</b>	29
<i>By Sikha Bagui</i>	

This column serves as an overview on the achievements of object-oriented database technology so far, and also discusses the weaknesses that have to be yet resolved by the object-oriented database community before object-oriented database technology can become as widespread as relational databases.

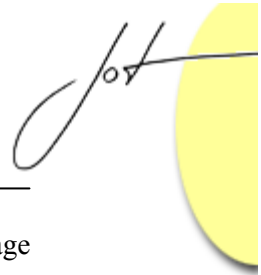
#### UML

<b>UML 2 Activity and Action Models</b>	43
<i>By Conrad Bock</i>	

This column covers motivation and architecture for the new models, basic aspects of UML 2 activities and actions, and introduces the general notion of behavior in UML 2.

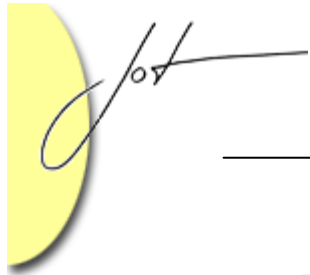


	Page
<b>Classification Theory</b>	
<b>The Theory of Classification, Part 8: Classification and Inheritance</b>	55
<i>By Anthony J.H. Simons</i>	
<p>We explore the differences between classes and types, developing the alternative formal model of classification and inheritance, which is quite different from subtyping.</p>	
<b>Business Objects</b>	
<b>The Best Practice Promise and Myth</b>	65
<i>By Mahesh H. Dodani</i>	
<p>The quest and use of best practices has always been an integral part of any software engineering practice. Best practices promise quality, consistency, efficiency and flexibility in engineering software systems. On the other hand, best practices can also be abused or misused – once a best practice has been established, it can be used by its proponents to force people to follow it blindly.</p>	
<b>Cyber Databases</b>	
<b>Towards Quantifying Data Quality Costs</b>	69
<i>By Won Kim</i>	
<p>A comparison of the costs of low quality data and ensuring high quality data can be a simple and compelling basis for an organization to determine the extent of the efforts it must expend to ensure high quality of its operational data.</p>	
<b>OO Requirements Engineering</b>	
<b>Specifying Good Requirements</b>	77
<i>By Donald Firesmith</i>	
<p>Many of the characteristics of properly specified requirements have been well known for many years, at least among professional requirements engineers. Yet most requirements specifications seen today in industry still include many poor-quality requirements. Far too many requirements are ambiguous, incomplete, inconsistent, incorrect, infeasible, unusable, and/or not verifiable (e.g., not testable). To combat this sad state of affairs, this column provides a questionnaire that can be used when specifying and technically evaluating requirements.</p>	



---

	Page
<b>Educator's Corner</b>	
<b>Branch and Bound Implementations for the Traveling Salesperson Problem - Part 3: Multi-threaded solution with many inexpensive nodes</b> <i>By Richard Wiener</i>	89
<p>The multi-threaded implementation presented in this column sets the stage for the distributed processing implementation to be presented in the next column. In the previous column a best-first branch and bound algorithm was introduced and implemented. This algorithm forms the basis for this current column.</p>	
<hr/> <b>REFEREED ARTICLES</b> <hr/>	
<b>Object-Oriented Intelligent Mechanism - Vital for the Success of E-Commerce</b> <i>By Myron Sheu and Xin (James) He</i>	101
<p>Motivated by widespread applications of e-commerce, this paper addresses the unique challenges of e-commerce by introducing object-oriented intelligence to the user interfaces of e-commerce rather than by utilizing traditional expert systems. Specifically, this research examines the intelligent mechanism settings that are aimed largely at improving knowledge representation of online transactions.</p>	
<b>Dynamic Caching Design Proto-Pattern for J2EE Web Component Development</b> <i>By Serestina Viriri</i>	113
<p>This paper discusses how to extend the scope of caching to un-cacheable content. It shows how to optimize the performance of J2EE Web applications by caching some of the dynamically-generated content.</p>	
<b>A Taxonomy of Components</b> <i>By Markus Voelter</i>	119
<p>The notion of component is not well defined. This paper presents a taxonomy of components.</p>	



	Page
<b>Formal Specification of Design Patterns - A Balanced Approach</b> <i>By Toufik Taibi and David Chek Ling Ngo</i>	127

The main problem of existing formal specification languages for design patterns is lack of completeness. This is mainly due either because they were not originally conceived to specify design patterns and have been adapted to do so, or they tend to focus on specifying either the structural or behavioral aspect of design patterns but not both of them. We propose a simple yet balanced pattern specification language that is aimed to achieve equilibrium by specifying both aspects of design patterns.

---

## OUTLOOK

---

<b>A brief outlook to the next issue</b>	141
--	-----