

Contents

	Page
Editorial	5

COLUMNS

Guest Column

Refactoring as Meta Programming?	7
<i>By Dave Thomas</i>	

Refactoring is widely acknowledged as one of the best practices of OO programming. It has been widely practiced in the functional and procedural community in one form or other for many years. Refactoring is a process that takes an existing program and improves it by transforming the program into a new program that is an improved version of the initial program.

Classification Theory

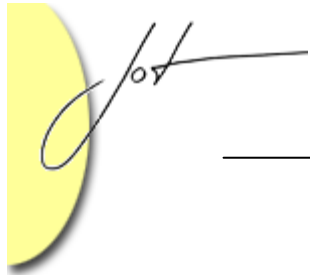
The Theory of Classification, Part 16: Rules of Extension and the Typing of Inheritance	13
<i>By Anthony J.H. Simons</i>	

Most object-oriented languages have restrictions on the types of overriding methods, to ensure that the resulting subclass is still type compatible with the superclass. This requires more precise rules about the typing of \oplus , the inheritance operator.

OO Requirements Engineering

Are Your Requirements Complete?	27
<i>By Donald Firesmith</i>	

What exactly does it mean for requirements to be complete? And what requirements work products are being referred to: individual requirements or requirements documents? And given that there is limited schedule and budget with which to perform requirements elicitation, analysis, specification, and management, just how complete should the requirements be?

**Java at Large****Resource Bundling for Distributed Computing** 45*By Douglas Lyon*

Wouldn't it be nice if a Java compiler could make sure that our resources were present, before run-time? In this way, we trade off a run-time error for a compile-time error. Some techniques are presented that allow resources to be integrated directly into the source code.

Strategic Software Engineering**Community** 59*By John D. McGregor*

Communities evolve over time. One way to evaluate the maturity of a technical community is to use the innovation adoption scale by Rogers : innovators, early adoptors, early majority, late majority, and laggards. A community is formed by a few innovators.

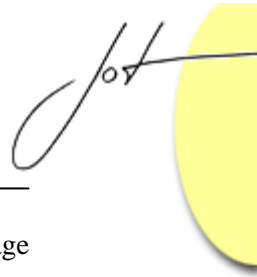
REFEREED ARTICLES

The Platform Based-Agents to Test and Evaluate Software Architecture 67*By Amar Ramdane-Cherif, Samir Benarif and Nicole Levy*

Architecture conception is a difficult and time consuming process, requiring advanced skills from the software architect. The tasks of an architect are alleviated if means can be provided to generate architectures that can be evaluated with respect to functional and non functional requirements. This paper discusses an easier approach for evaluating software architecture.

E-Bunny: A Dynamic Compiler for Embedded Java Virtual Machines 83*By Mourad Debbabi, Abdelouahed Gherbi, Lamia Ketari, Chamseddine Talhi, Hamdi Yahyaoui, Sami Zhioua, and Nadia Tawbi,*

The primary objective of our work is to come up with an efficient, lightweight and low-footprint accelerated embedded Java Virtual Machine. This is achieved by the means of integrating a selective dynamic compiler that we called E-Bunny into the J2ME/CLDC virtual machine KVM.



	Page
Ontology Modeling and MDA	109
<i>By Dragan Djuric, Dragan Gasevic and Vladan Devedzic</i>	
<p>The paper presents Ontology Definition Metamodel (ODM) that enables using Model Driven Architecture (MDA) standards in ontological engineering. Other similar metamodels are based on ontology representation languages, such as RDF(S), DAML+OIL, etc. However, none of these other solutions uses the recent W3C effort – The Web Ontology Language (OWL).</p>	
Foundations for MDA-based Forward Engineering	129
<i>By Liliana Favre</i>	
<p>Model Driven Architecture (MDA) is an emerging technology that is supposed to provide a technical framework for information integration and tools interoperability; many UML tools claim to be compliant with it. Model-to-model transformations are essential in MDA.</p>	
A Java Implementation of the Branch and Bound Algorithm: The Asymmetric Traveling Salesman Problem	155
<i>By Pawel Kalczynski</i>	
<p>This paper revisits some 20-year-old algorithms and contributes to object technology by offering a description of the model and implementation of a generic package for solving discrete minimization problems with branch and bound method. In particular, the package is implemented for and tested on the Traveling Salesman Problem with asymmetric cost matrices.</p>	
<hr/>	
A Unified Superstructure for UML	165
<i>By Andy Evans, Paul Sammut, James S. Willans, Alan Moore, and Girish Maskeri</i>	
<p>A key aspect of successfully using UML is understanding the semantics of the notations. UML 2 will increase the already substantial collection of notations supported by UML 1.x. At the same time, this will augment the difficulty users experience in understanding semantics. In this paper we propose that while the diverse notations may render concepts differently, the concepts can often be considered semantically equivalent.</p>	
<hr/>	
BOOK REVIEW	
<hr/>	
My best books of 2004	183
<i>An overview by Charles Ashbacher</i>	
<hr/>	
OUTLOOK	
<hr/>	
A brief outlook to the next issue	187
