

Contents

	Page
Editorial	5

COLUMNS

Java at Large

The JBoss Integration Plug-in for IntelliJ IDEA, Part 1	7
<i>By Douglas Lyon, Martin Fuhrer, Thomas Rowland</i>	

The IntelliJ IDEA is an Integrated Development Environment (IDE) used for Java development. It is known for its strong refactoring capabilities. It is a closed-source, proprietary product, which is used in both educational and industrial settings. IntelliJ functionality is extended by a set of open APIs that third-party developers can use to integrate their solutions by the development of plug-ins.

Strategic Software Engineering

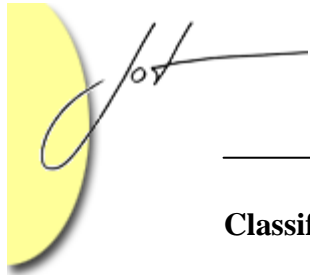
Customer Interface Management	19
<i>By John D. McGregor</i>	

One of the interfaces between the producing company and the consuming company is the formal review. Requirements reviews, design reviews, test reviews, and schedule reviews are all points at which the customer and producer come together. These need to be managed as explicit project activities. There are several different types of producer/consumer relationships and each has its own characteristics and its own interface at review time.

Cyber Databases

A Framework for the Integration of Multimedia Data	27
<i>By Won Kim, Seung Soo Park, Hyon Hee Kim</i>	

In January 2005, George Mason University found that hackers had gained access to a database containing information on 32,000 people including their social security numbers. In February 2005, ChoicePoint, a data collection service, announced that a security breach threatened the personal information of over 145,000 people. And the list could go on. We want the software that manages our personal and professional data to be secure.

**Classification Theory**

- The Theory of Classification Part 19:
The Proliferation of Parameters** 37
By Anthony J H Simons

We consider in more detail the kinds of manipulations performed upon polymorphic class-types. These are expressed using function-bounded type parameters of the form: $\tau <: F[\tau]$, where F is a type function, describing the shape of the interface that the type τ is expected to satisfy [2]

Guest Column

- Aligning IT with the Changes using the Goal-Driven Development
for UML and MDA** 49
By Birol Berkem

In order to assist organizations for aligning IT with their changing business environment, a Goal-Driven Development Process should be considered to accompany the OMG's MDA (2). Using such a methodology, MDA users could enable their organisations in propagating changes they capture through their business processes till their IT applications, thus synchronizing IT with their changing business rules, as a result to better capitalize on their business knowledge, independently of technological changes.

Business Objects

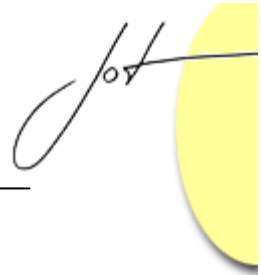
- Mirror, Mirror on the Wall, Whose SOA is the Best of them All?** 67
By Mahesh Dodani

Key to detailing the SOA reference architecture is to define the architecture of composite services that align with business processes. The relationship between services and components is that enterprise-scale components (large-grained enterprise or business line components) realize the services and are responsible for providing their functionality and maintaining their quality of service.

REFEREED ARTICLES

- Patterns of Interface-Based Programming** 75
By Friedrich Steimann and Philip Mayer

Modern software architectures heavily promote the use of interfaces. Originally conceived as a means to separate specification from implementation, popular programming languages today accommodate interfaces as special kinds of types that can be used – in place of classes – in variable declarations. While it is clear that these interfaces offer polymorphism independent of the inheritance hierarchy, little has been said



about the systematic use of interfaces, or how they are actually used in practice.

A Taxonomy of OO Classes to Support the Mapping of Testing Techniques to a Class 95

By Peter J. Clarke, Brian A. Malloy

The trend in the development of large scale object-oriented systems has shifted toward testable, robust models, with a focus on the prevention of faults and system failure. One process that supports the construction of robust software is testing. An advantage of software testing is the relative ease with which some of the testing activities can be performed, such as executing the program using a given set of inputs, or test cases, and then comparing the generated output to the expected output.

A Study of Test Coverage Adequacy In the Presence of Stubs 117

By Errol L. Lloyd, Brian A. Malloy

The purpose of implementation-based testing is to gain a measure of confidence in the correctness of the software by providing adequate coverage of the code. One unit of testing in object-oriented software is a class. However, classes use other classes and if class interactions form a cycle of dependencies then, to test a client class that uses an untested supplier class, stubs must be constructed to simulate the correct behavior of the untested supplier class.

An Implementation of the π -Calculus on the .NET 139

By liwu Li

An approach is presented to implement the π -calculus. In the approach, we realize a π -calculus process expression by decomposing the expression into a hierarchy, whose nodes manage, coordinate, and perform the basic actions that are specified in the process expression. When the process expression is executed, new nodes are added to the hierarchy to account for lower-level process expressions.

PRODUCT REVIEW

JBuilder 2005 Enterprise and Together Developer 2005 159

Reviewd by Dave Neuendorf and Richard Wiener

OUTLOOK

A brief outlook to the next issue 163
