

Contents

	Page
Editorial	5

COLUMNS

Strategic Software Engineering

Natures and Perspectives	7
<i>By John McGregor</i>	

The nature of core asset building in a software product line organization spans the extremes reflected in the pure proactive and reactive approaches to asset base development with a continuum of intermediate flavors available. The proactive approach builds a complete, or nearly complete, set of assets prior to the majority of product development. The reactive approach builds assets, or modifies existing assets, just in time for use in a product.

The OOP Scene

Programming with Models – Modeling with Code	15
<i>The Role of Models in Software Development</i>	
<i>By Dave Thomas</i>	

Models provide a representation of partial truths about the software system, but only the actual code contains the whole truth. The art of programming is about managing the half-truths at higher levels of abstraction in a way such that they are just sufficiently correct to make progress, and sufficiently vague that they can be refined at the next level. Good software evolves like a book outline starting with an initial set of key ideas which are ruthlessly refactored into the final form. Model refinements and compositions can at best be captured as a set of annotated hyperlinks which describe the evolution of the software as a literate program.

OO Requirements Engineering

- Requirements Engineering Tasks** 21
By Don Firesmith

Having well-engineered requirements is critical. This is not just due to their major positive impact on project costs (both development and life-cycle) and schedule, which are largely due to the extreme costs of fixing requirements defects once the system is built and fielded. Also critical is having well-engineered functional and quality requirements because of their positive impact on system acceptability by its many stakeholders.

Java at Large

- The Intium RJS Screensaver: Part 4, Automatic Deployment** 31
By Douglas Lyon and Francisco Castellanos

The Intium RJS System makes use of screensavers to perform CPU scavenging for grid computing. A Java Web Start application has been created to complete the installation of the screensaver for Windows and Unix/Linux platform. This application identifies the operating system and proceeds to download, install, and configure the screensaver files.

Business Objects

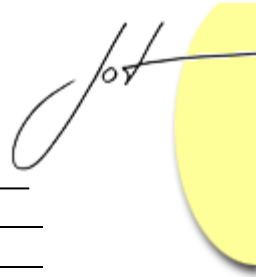
- SOA 2006: State Of The Art** 41
By Mahesh Dodani

As we have progressed through 2006, SOA has dug its heels into all aspects of software engineering and become entrenched as the main enabler of the holy grail of aligning IT to business, along with the ability to facilitate business innovation through a flexible and agile IT.

Guest Column

- Encrypting the Java Serialized Object** 49
By Alan Ward

The pros and cons of encrypting Java objects are examined, not in terms of Class files that can be signed, transmitted through secure channels (such as SSL) and stored in protected volumes on disk, but rather as serialized objects that can do all that – and maintain their state through transmission as well.



REFEREED ARTICLES

Structural Design Patterns and .NET Framework 2.0

59

By Dhamayanthi N and Thangavel P.

This paper investigates applications of design patterns in .NET Framework. Among the seven structural patterns described by Gang of Four (GOF), we have discussed five patterns from .NET perspective. As there is no official documentation available on the patterns used in the latest version of .NET Framework (version 2.0) currently, this study would be beneficial for .NET designers in understanding patterns behind the framework, which in turn would help them in applying relevant patterns while developing their applications in .NET Framework.

Matching Objects Without Language Extension

81

By Joost Visser

We propose an approach to support pattern matching in mainstream object-oriented languages without language extension. In this approach, a pattern is a first-class entity, which can be created, be passed as argument, and receive method invocations, just like any other object. We demonstrate how our approach can be used in conjunction with existing parser generators to perform pattern matching on various kinds of abstract syntax representation.

Framework Evolution Tool

101

By Mariela Cortés, Marcus Fontoura and Carlos Lucena

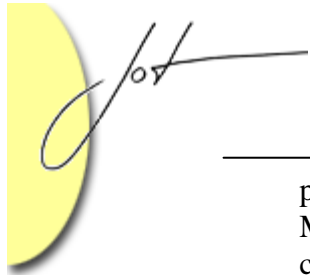
An intrinsic property of software in a real-world environment is its need to evolve. In particular, framework technology constitutes an ever-evolving representation of our knowledge of the domain in terms of variations and commonalties. A key point in framework development is that design work should not start by trying to model its variability and flexibility at once.

The Tao of Modeling Spaces

125

By Dragan Djurić, Dragan Gašević and Vladan Devedžić

What are models and metamodels, and why do we need them; What does it mean to transform a model into a programming language. Unlike current research efforts that answer to those questions in rather



partial ways, we define a formal encompassing framework (i.e. Modeling spaces) for studying many modeling problems in a more comprehensive way. We illustrate the benefits of that framework for explaining present dilemmas practitioners have regarding models, metamodels, and model transformations.

Empirical Study of Object-Oriented Metrics

149

By K.K.Aggarwal, Yogesh Singh, Arvinder Kaur and Ruchika Malhotra

Unlike other engineering disciplines, software engineering is not grounded in the basic quantitative laws of physics. Absolute measurements, such as voltage, mass, velocity or temperature, are uncommon in the software world. Instead, we attempt to derive a set of indirect measures that lead to metrics that provide an indication of the quality of some representation of software.

OUTLOOK

A brief outlook to the next issue

175