

Contents

	Page
Editorial	5

COLUMNS

Guest Column	7
---------------------	---

Using Active Objects for Structuring Service Oriented Architectures

Anthropomorphic Programming with Actors

By Dave Thomas and Brian Barry

Service Oriented Architectures (SOAs), which were previously beneficial in legacy OLTP and Telecom systems, are once again popular, this time for use with web services. SOAs offer language and technology independence, including the important ability to not require every useful program to be in the latest language and/or on the latest platform. SOAs use self described wire formats such as SOAP, which make it easy to communicate between different technologies.

Classification Theory

The Theory of Classification	15
-------------------------------------	----

Part 12: Template Classes and Genericity

By Anthony J.H. Simons

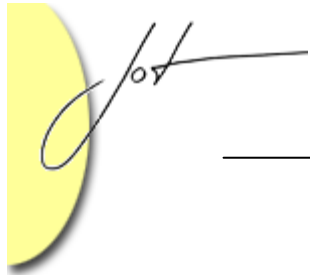
In this article, Simons explores the consequences of adding generic classes to the Theory of Classification. After looking at some historical notions of polymorphism and type parameters, he examines how to incorporate these into the type-level of the theory, and then looks at how introducing or instantiating type parameters can be combined with the process of deriving subclasses by inheritance.

Cyber Databases

On Supporting Structure-Agnostic Queries for XML	27
---	----

By Won Kim, Wol Young Lee and Hwan Seung Yong

This navigational access to XML documents is a natural consequence of the hierarchical structure of XML. However, it is also desirable to allow the users to formulate structure-agnostic queries against XML documents, to complement the current navigation-based queries.



UML

- UML 2 Activity and Action Models** 37
Part 5: Partitions
By Conrad Bock

This column describes partitions, which are a way of grouping actions that have some characteristic in common. In particular, they can relate actions to classes that are responsible for them, and highlight the abstraction that activities provide for interaction diagrams and state machines.

Java at Large

- The Imperion Threading System** 57
By Douglas Lyon

The Imperion threading system is a more reliable threading system than the normal *java.lang.Thread*. Imperion removes dangerous methods and guards' inputs in order to avoid exceptions.

Strategic Software Engineering

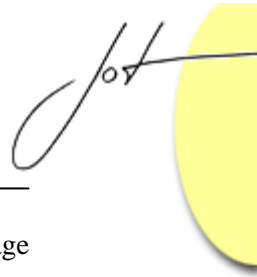
- Domain *** 71
By John D. McGregor

Taking a domain-based approach provides a context for software development that creates synergy with other business activities of the company producing strategically significant results.

REFEREED ARTICLES

- UML Associations: A Structural and Contextual View** 83
By Gonzalo Génova, Juan Llorens and José M. Fuentes

The different kinds of communication links that can exist in an interaction among objects pose the question of whether every link is or is not an instance of an association, and whether an association must exist whenever there is a communication path between objects. The distinction between static and dynamic associations is not adequate to solve this problem, since in object-orientation every association has static and dynamic features, so that these two aspects do not serve to define two disjoint subtypes of association.



	Page
Extending the Java Language with Dynamic Classification <i>By Liwu Li</i>	101

A dynamic classification feature of an object-oriented programming language allows an object to change its class membership without changing its identity at runtime. The new membership of the object can be signified with a role, which is taken on by the object and which can be implemented as an object of the target class.

Online Upgrade of Object-Oriented Middleware <i>By Apostolos Zarras</i>	121
---	-----

A system that relies on object-oriented middleware comprises computational objects that are specific to the system's purpose and middleware objects used for the transparent integration of the former. The efficient maintenance of such a system involves the dynamic upgrade of the aforementioned entities. So far, there have been various approaches dealing with the online upgrade of computational objects. This paper examines the second part of the problem: the online upgrade of middleware.

OUTLOOK

A brief outlook to the next issue	141
--	-----