Tracks
Technical Program
Tutorials
Workshops
DesignFest
Educators' Symposium
Doctoral Symposium
Demonstrations
Posters
Student Research Competition
Student Volunteers
Special Events
Exhibits
Housing Information
Registration Information
Transportation

Chair: James Noble
Victoria University of Wellington, New Zealand, kjx@mcs.vuw.ac.nz

The 2002 OOPSLA Doctoral Symposium will prepare the leaders of the future of object-oriented technology. In the symposium, a selected group of doctoral students present their work in a closed session and obtain guidance from university and industry mentors. The goal of the symposium is to expose students to helpful criticism before their thesis defense and to provide advice on future research and employment. This year's mentors are Prof. Alan Borning (The University of Washington), Prof. Mira Mezini (Darmstadt University of Technology), Dr. Dirk Riehle, and Dr. Mario Wolczko (Sun Microsystems Laboratories Inc.). To provide further opportunity for discussion and feedback, all doctoral symposium presenters have a poster on display during the conference and a two-page short paper published in the Conference Companion proceedings.

As in the past, this year's selected participants were chosen on the basis of the mentors' evaluation of the student's descriptions of their research, and whether the work was sufficiently advanced to have some preliminary results but with sufficient time remaining to benefit from the symposium experience.

Tuesday 8:30-17:30 – Sheraton Hotel - Cedar

A Framework for Performance Management of Component Based Distributed Applications

Adrian Mos
Dublin City University, mosa@eeng.dcu.ie
We propose the Component Performance Assurance Solutions (COMPAS) framework that can be used during as well as after development to identify performance problems, suggest corrections and predict performance in large-scale component-based distributed enterprise systems. COMPAS consists of three interrelated modules: monitoring, modelling and prediction.

Toward a Management Framework for Self-Adaptive Systems

Finbar McGurren
University of Limerick, finbar.mcgurren@ul.ie
Maintainability is becoming increasingly important, as systems will have to evolve and adapt to changing environments. We investigate dynamic software architectures as a means of providing facilities for developing dynamically adaptable applications.

The Design and Verification of Java's Memory Model

Jeremy Manson
University of Maryland, College Park, jmanson@umd.edu
Java's threading specification is fundamentally flawed. Some language features are under-specified while others are over-specified, and some have no specification at all. We attempt to provide a clear and concise definition of thread interaction to remedy these limitations.

The Use of Domain Level Semantics to Support Unanticipated System Adaptation

Damien Conroy
University of Limerick, damien.conroy@ul.ie
If interfaces were to be described in application domain terms then mappings could be provided to enable translations from application domain entities to syntactic interface entities. More specifically, this capability could enable automated adaptation to resolve syntactic discrepancies between semantically equivalent entities at runtime.

An Aspect-Oriented Infrastructure for a Typed, Stack-based, Intermediate Assembly Language

Douglas Dechow
Computer Science Department, Oregon State University, dechow@cs.orst.edu
We propose the creation of an aspect-oriented infrastructure to support a variety of software development tools. We investigate several new directions in aspect-orientation: aspects in system software, language independent aspects, aspect integration techniques, and opportunities for aspect reuse.

Separation of Concerns through Semantic Annotations

João Cachopo
Technical University of Lisbon, jcachopo@gia.ist.utl.pt
We propose the use of programmer extensible program annotations as a means to represent information about the domain. Using these program annotations we can specify join points by means of semantic properties of the programs, thereby improving the reusability and robustness of aspects.

Modular Programming with Aspectual Collaborations

Johan Ovlinger
Northeastern University, johan@ccs.neu.edu
We propose “Aspectual Collaborations”, modules tailored to capture multi-object behavioral interactions, both at the explicit and implicit levels. We explore static and dynamic properties of this construct, and provide a sample Java implementation

---

Encapsulating Concurrency with Early-Reply

Scott Pike
Ohio State University, pike@cis.ohio-state.edu
Early-Reply introduces runtime concurrency by forwarding invocation results to the caller as early as they are available. We propose to reformulate Early-Reply using local proof obligations that encapsulate concurrency behind a sequential abstraction of system execution.

A Measure of Design Readiness: Using Patterns to Facilitate Teaching Design Readiness

Tracy Lewis
Virginia Tech, tracyL@vt.edu
We propose a measure of assessing “design readiness” - an assessment of the cognitive state where one is able to understand design abstractly. We will then use programming and design patterns to assist in teaching critical design concepts.