

DOCTORAL SYMPOSIUM

Chair: Doug Lea, *SUNY Oswego*

Monday

Marriott Hotel — Meeting Room 10

OOPSLA 2001 will provide a selected group of doctoral students with a forum in which to present their work and to obtain guidance from mentors drawn from university faculty and industry laboratories. The goal of the symposium is to expose the student to external helpful but critical peers before their defense in order to improve upon their thesis work, as well as to give points of advice for job interviews. This year's mentors are Brent Hailpern (IBM), Doug Lea (SUNY Oswego), James Noble (Victoria University of Wellington, NZ), Mary Beth Rosson (Virginia Tech), Richard Gabriel (Sun Microsystems), and Ron Goldman (Stanford University).

As in the past, this year's selected participants were chosen on the basis of the mentors' evaluation of the students' 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.

The participating doctoral students and the titles of their research work are:

1 **Integration of Independently-Developed Object-Oriented Designs**

Adam Batenin, *University of Bath*

Programming technology should facilitate the independent development of object-oriented designs and their subsequent composition to create complete programs. We show that even in an environment where a limited independence is granted, relatively simple programs are difficult to integrate using present technologies. We propose techniques to address these issues.

2 **Design Support for Aspect-Oriented Software Development**

Christina von Flach G. Chavez and Carlos J.P. de Lucena,
Pontifical Catholic University of Rio de Janeiro

We propose a design model for aspect-oriented software development, which incorporates the main features of aspect-oriented programming and is language and process independent. A set of design principles to aspect-oriented modeling is under investigation. A representative case study and a development scenario will illustrate the suitability of the proposed ideas.

3 **Transmigration of Object Identity: The Programming Language GILGUL**

Pascal Costanza, *University of Bonn*

Gilgul is an extension of Java that strictly separates the notions of reference and comparison that are traditionally subsumed in the concept of object identity. This allows for the introduction of new operations that open up new degrees of flexibility by providing means for unanticipated, dynamic software adaptation.

4 **A Multithreaded Concurrent Generational Garbage Collector for Java**

Chia-Tien Dan Lo and J. Morris Chang,
Illinois Institute of Technology

A new multithreaded concurrent generational garbage collector based on mark-sweep with the assistance of reference counting is proposed and implemented. The scheme takes advantage of multiple processors in SMP systems and the merits of threads. Furthermore, it reduces garbage collection pauses and enhances garbage collection efficiency. Measurement results are studied.

5 An Object-Oriented Framework for Distributed Numerical Computations

Roxana Diaconescu, *Norwegian University of Science and Technology*

Writing concurrent distributed memory applications requires skills beyond the background of a researcher experimenting with scientific problems. We propose a component framework to capture the concurrency infrastructure for dynamically distributed numerical applications. We focus on the problem of concurrent Finite Element Method (FEM) solution of Partial Differential Equations (PDEs) for general (unstructured) meshes.

6 A Framework for Creating Aspect Weavers

Jeff Gray, *Vanderbilt University*

We present two objectives for extending Aspect-Oriented Programming (AOP) research. First, the concept of AOP is investigated at a higher level of abstraction, focusing on the application of aspect-oriented techniques to model-integrated computing. Second, we create a framework that aids in the construction of new aspect weavers. The framework utilizes domain-specific languages and generators to provide variability among weaver instances.

7 An OCL Query-Based Debugger for C++

Chanika Hobatr, *Clemson University*

Development using object technology can make debugging a daunting task due to the number of generated objects and the complexity of their interrelationships. We propose a debugging tool that permits the expression of complex relationships to be formulated easily and evaluated efficiently. OQBD is a query-based debugger for C++ programs using queries formulated in the Object Constraint Language.

8 Supporting the Use of Object-Oriented Frameworks

Daqing Hou, *University of Alberta*

Object-oriented frameworks are often hard to learn and use. The capability to automatically detect errors occurring at the boundary between frameworks and applications is crucial to mitigate the problem. We introduce the notion of framework constraints and a specification language, FCL (Framework Constraints Language), to formally specify them. Framework constraints are rules that frameworks impose on the code of framework-based applications.

9 Assessing the Quality of Object-Oriented Designs

Ralf Reißing, *University of Stuttgart*

Design quality is vital for reducing software cost. Unfortunately, it is unclear what design quality really is. This work wants to clarify the notion of design quality and make it measurable. A quality model is created that can be used for design assessment, comparing design alternatives, and design improvement.

10 Separation of Distribution Concerns in Distributed Java Programming

Michiaki Tatsubori, *University of Tsukuba*

We propose an aspect-oriented distributed programming tool, with which programmers can specify the distribution aspect of program simply and separately from non-distributed Java program, for enhancing the modularity of program. The aspect-weaver of this tool is a bytecode translator implemented as a customized class loader in Java.

11 Interdependence of Software Evolution and Development Process Evolution in Agile Methodologies

Christian Wege, *University of Tuebingen*

A software development process can become more successful by analyzing the produced artifacts and feeding the results back into the process. This research will produce techniques for analyzing the artifacts, show how to interpret them and list a collection of experiences made in real world projects.

STUDENT VOLUNTEERS

Chair: Dirk Siebert, *Leipzig University*, sv@oopsla.acm.org

The student volunteers program is an opportunity for students from around the world to associate with the top people in object-oriented technologies and software development. In return for about ten hours of their time, student volunteers receive complimentary registration and other benefits. In the past, job assignments have included assisting with tutorials, and panels, checking badges at doors, helping with traffic flow, and general go-for assistance to keep the conference running smoothly.

Student volunteers need to be enrolled in a full-time undergraduate, or graduate program at the time of the conference.

More detailed information is available on <http://oopsla.acm.org/~sv>.

BIRDS OF A FEATHER SESSIONS (BOFs)

Birds of a Feather Sessions are informal gatherings that provide a forum for discussion of a particular topic, tool, product, etc. Several rooms will be available for BOFs each evening from Sunday through Wednesday. Rooms may be reserved by any Conference attendee on a first-come, first-served basis by signing up at the Information Booth at the Conference. There will be no advance reservations for these rooms prior to the Conference.

BOF organizers are responsible for providing descriptive information for their session, which OOPSLA will post in the OOPSLA Courtyard. BOFs require no registration or submission for attendance. All OOPSLA attendees may attend any BOF session(s).