Conference Programme

2004 ecoop
oslo, norway

18th European Conference on Object-Oriented Programming

University of Oslo
14–18 June 2004
Maps........................................................................................................................... .5
Conference Overview ..............................................................................................................6
Invited Talks ............................................................................................................................10
Functionality Object ..................................................................................................................10
Rich Interfaces for Software Modules ......................................................................................10
Banquet Speech..........................................................................................................................11
Facilitating the Evolution of our Collective IQ ........................................................................11
Technical Program ....................................................................................................................12
Programming Concepts I ...........................................................................................................12
  • Ownership Domains: Separating Aliasing Policy from Mechanism .................................12
  • Composable Encapsulation Policies ..................................................................................12
Program Analysis .....................................................................................................................12
  • Demand-Driven Type Inference with Subgoal Pruning; Trading Precision for Scalability ....12
  • Efficiently Verifiable Escape Analysis ..............................................................................12
  • Pointer Analysis in the Presence of Dynamic Class Loading ............................................12
Software Engineering .............................................................................................................12
  • The Expression Problem Revisited — Four new solutions using generics .......................12
  • Rewritable Reference Attributed Grammars .....................................................................12
  • Finding and Removing Performance Bottlenecks in Large Systems .................................12
Aspects .....................................................................................................................................12
  • Programming With Crosscutting Effective Views ................................................................12
  • AspectJ 2EE = AOP + J2EE — Towards an Aspect Based, Programmable and Extensible Middleware Framework ....................................................................................12
  • Use Case Level Pointcuts ................................................................................................12
Middleware ..............................................................................................................................13
  • Inheritance-Inspired Versioning for CORBA .....................................................................13
  • ModelFrame for Persistence and Querying of Java Objects ................................................13
  • Sequential Object Monitors .............................................................................................13
  • Increasing Concurrency in Databases using Program Analysis ........................................13
Programming Concepts II .....................................................................................................13
  • Semantic Casts: Structural Subtyping in a Nominal World ................................................13
  • LOOM: Weaving LOOM into Java ....................................................................................13
  • Modules with Interfaces for Dynamic Linking and Communication ...............................13
Verification ...............................................................................................................................13
  • Early identification of incompatibilities in multi-component upgrades .............................13
  • Typestates for Objects ......................................................................................................13
  • Object Invariants in Dynamic Contexts ............................................................................13
Systems ...................................................................................................................................13
  • Transactional Monitors for Concurrent Objects ................................................................13
  • Adaptive Tuning of Reserved Space in an Appel Collector ................................................13
  • Lock Reservation for Java Reconsidered ...........................................................................13
  • Customization of Java Library Classes using Type Constraints, and Profile Information ....13
Panel .........................................................................................................................................14
Object-oriented Programming and XML ..................................................................................14
Practitioners’ Reports ..............................................................................................................14
  • PR1 Keynote: Application Development Scenario - AD at the Crossroads ...........................14
  • PR2 A New Approach to Solutions Design and Systems Engineering ..............................14
  • PR3 Generating Service Descriptions and Code from Reusable Models ...........................14
  • PR4 Creating an Engine for Data-Centric Collaboration ...................................................14
  • PR5 FAST Data Search: Object-Oriented Programming in a High Performance Distributed Application ..................................................................................................................15
  • PR6 DocuLive – an Object-Oriented Patient Record System .............................................15
  • PR7 Using ISO 10303-11, Express, the Information Modeling Language, to create Advanced Solutions for Industrial Data Applications ........................................................................15
  • PR8 Experiences from introducing a component oriented UML-based Development Method .................................................................................................................................15
  • PR9 Testing framework for Platform Independent Models with TTCN-3 ................................15
  • PR10 MDA as a framework for supporting Enterprise standards ......................................15
Demonstrations .......................................................................................................................16
  • D1: Advanced Refactorings in Eclipse ...............................................................................16
  • D2: A Web Services Management Layer for the selection, integration and management of web services .................................................................16
  • D3: OptimalIJ - Model-Driven, Pattern-Based Development .................................................16
Posters .....................................................................................................................................17
  • Axiomatic Mathematics meets Object Oriented Programming .........................................17
  • SICSDA: an Adaptive Configurable Distributed Software Architecture Applied to Satellite Control Missions .................................................................17
  • A UML profile for service-oriented architectures ..................................................................17
  • Using Design Patterns to Abstract a Software Architecture for Natural Language Generation ...................................................................................................................17
  • UnderFrame: Understanding Object-Oriented Frameworks Using a Case-Based Teaching Approach .........................................................................................17
  • CUP: A CORBA-based Unified Process Adaptation ..........................................................17
  • Domain Modeling in Self Yields Warped Hierarchies .........................................................17
  • Learning object-oriented concepts mediated through collaboration and use of artifacts ..........17
<table>
<thead>
<tr>
<th>Tutorials:</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The AspectJ Middleware Platform</td>
<td>17</td>
</tr>
<tr>
<td>Developing Secure and Critical systems</td>
<td>18</td>
</tr>
<tr>
<td>TS1: Building secure software applications</td>
<td>18</td>
</tr>
<tr>
<td>TS2: Critical (and secure) systems development with UML: methods and tools</td>
<td>18</td>
</tr>
<tr>
<td>UML and Model Driven Architecture</td>
<td>18</td>
</tr>
<tr>
<td>TU1: Model-driven architecture in practice</td>
<td>18</td>
</tr>
<tr>
<td>TU2: MDA and metamodelling</td>
<td>19</td>
</tr>
<tr>
<td>Web Services and BPEL</td>
<td>19</td>
</tr>
<tr>
<td>TW1: Introduction to XML web services using .NET as an example</td>
<td>19</td>
</tr>
<tr>
<td>TW2: Programming internet-scale distributed applications in the 21st century: BPEL and beyond</td>
<td>19</td>
</tr>
<tr>
<td>Microsoft .Net.</td>
<td>19</td>
</tr>
<tr>
<td>TN1: Introduction to .NET using C#</td>
<td>19</td>
</tr>
<tr>
<td>TN2: An introduction to the Common Language Infrastructure (CLI)</td>
<td>20</td>
</tr>
<tr>
<td>Languages and Execution Environments Comparison</td>
<td>20</td>
</tr>
<tr>
<td>TL1: Foundations of object-oriented languages: types and language design</td>
<td>20</td>
</tr>
<tr>
<td>TL2: J2EE versus .NET</td>
<td>20</td>
</tr>
<tr>
<td>Java</td>
<td>20</td>
</tr>
<tr>
<td>TJ1: New features in J2SE 1.5</td>
<td>20</td>
</tr>
<tr>
<td>TJ2: Developing Java applications for mobile phones: approach and new Java standard interfaces for J2ME</td>
<td>21</td>
</tr>
<tr>
<td>Eclipse</td>
<td>21</td>
</tr>
<tr>
<td>TE1: Contributing to Eclipse: plug-ins and how to write your own</td>
<td>21</td>
</tr>
<tr>
<td>TE2: Contributing to Eclipse: advanced plug-in development</td>
<td>21</td>
</tr>
<tr>
<td>Design and Software Development</td>
<td>22</td>
</tr>
<tr>
<td>TD1: Design by contract and automatic verification for Java with JML and ESC/Java2</td>
<td>22</td>
</tr>
<tr>
<td>TD2: Adaptive object-model architecture: how to build systems that can dynamically adapt to changing requirements</td>
<td>22</td>
</tr>
<tr>
<td>Generative Programming and UML 2.0 Testing</td>
<td>22</td>
</tr>
<tr>
<td>TG1: Generative programming: methods, techniques, and applications</td>
<td>22</td>
</tr>
<tr>
<td>TG2: The UML 2.0 testing profile — the approach for test design and generation in UML-based development processes</td>
<td>22</td>
</tr>
<tr>
<td>Middleware and Distribution</td>
<td>23</td>
</tr>
<tr>
<td>TJ1: New features in J2SE 1.5</td>
<td>23</td>
</tr>
<tr>
<td>TJ2: Developing Java applications for mobile phones: approach and new Java standard interfaces for J2ME</td>
<td>21</td>
</tr>
<tr>
<td>Project Management</td>
<td>23</td>
</tr>
<tr>
<td>TP1: Seeing the forest and the trees — using system dynamics for high-level decision and project improvement</td>
<td>23</td>
</tr>
<tr>
<td>Workshops:</td>
<td>24</td>
</tr>
<tr>
<td>WS1: 2nd European Workshop on Web Services and Object Orientation (EOOWS 2004)</td>
<td>24</td>
</tr>
<tr>
<td>WS2: Practical Problems of Programming in the Large Workshop</td>
<td>24</td>
</tr>
<tr>
<td>WS3: 8th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2004)</td>
<td>24</td>
</tr>
<tr>
<td>WS4: Eighth Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts</td>
<td>24</td>
</tr>
<tr>
<td>WS5: 2nd Workshop on Object-Oriented Language Engineering for the Post-Java Era: Back to Dynamics</td>
<td>24</td>
</tr>
<tr>
<td>WS6: Philosophy, Ontology, and Information Systems</td>
<td>24</td>
</tr>
<tr>
<td>WS7: Communication Abstractions for Distributed Systems</td>
<td>24</td>
</tr>
<tr>
<td>WS8: Formal Techniques for Java-like Programs (FTJJP)</td>
<td>24</td>
</tr>
<tr>
<td>WS9: Component-oriented approaches to context-aware systems</td>
<td>24</td>
</tr>
<tr>
<td>WS10: The combined 14th Workshop for PhD Students in Object-Oriented Systems and Doctoral Symposium</td>
<td>24</td>
</tr>
<tr>
<td>WS11: Multiparadigm Programming with Object-Oriented Languages (MPOOL’04)</td>
<td>24</td>
</tr>
<tr>
<td>WS12: Mechanisms for Specialization, Generalization and Inheritance (MASPEGHI 2004)</td>
<td>24</td>
</tr>
<tr>
<td>WS13: Reflection, AOP and Meta-Data for Software Evolution (RAM-SE04)</td>
<td>25</td>
</tr>
<tr>
<td>WS14: First International Workshop on Coordination and Adaptation Techniques for Software Entities (WCAT’04)</td>
<td>25</td>
</tr>
<tr>
<td>WS15: Model Transformation and execution in the context of MDA</td>
<td>25</td>
</tr>
<tr>
<td>WS17: Interoperability of Enterprise systems</td>
<td>25</td>
</tr>
<tr>
<td>WS20: 10TH ECOOP Workshop on Mobile Object Systems</td>
<td>25</td>
</tr>
<tr>
<td>WS22: 5th International Workshop on Object-Oriented Reengineering (COR 2004)</td>
<td>25</td>
</tr>
<tr>
<td>WS23: Evolution and Reuse of Language Specifications for DSLs (ERLS)</td>
<td>25</td>
</tr>
<tr>
<td>Exhibition:</td>
<td>26</td>
</tr>
<tr>
<td>Social Program</td>
<td>27</td>
</tr>
<tr>
<td>Reception at the City Hall of Oslo</td>
<td>27</td>
</tr>
<tr>
<td>Boat Trip on the Oslo fjord</td>
<td>27</td>
</tr>
<tr>
<td>Banquet at Gamle Logen</td>
<td>27</td>
</tr>
<tr>
<td>Organizers of ECOOP 2004</td>
<td>28</td>
</tr>
</tbody>
</table>
Welcome to ECOOP 2004

The organizing committee welcomes you to the 18th European Conference on Object-Oriented Programming. From its start in 1987 (Paris), the ECOOP series of conferences has proved to be the European conference that is the obvious choice for presentation of new ideas, discussing even newer ideas and present tutorials on state-of-the-art technologies. The second ECOOP (88) was held in Oslo, with Kristen Nygaard as Program Chair and Ole-Johan Dahl as invited speaker. They will not be with us this time, but the conference will have an exhibition in memory these two pioneers, in addition to Edsger W. Dijkstra. They were of rather different nature, but one thing they had in common: the liked good and lively discussions. We hope that you will also enjoy good and lively discussions during this week.

ECOOP 2004 is organized by the Department of Informatics, University of Oslo, Norwegian Computing Center, and SINTEF.

under the auspices of AITO: Association Internationale pour les Technologies Objets and in cooperation with ACM Sigplan.

Sponsors
Maps: City Center → Conference Site (Blindern)

How to get to the Conference Site

1. Take westbound T-bane (Metro) line 3 (direction Sognsvann) or 5 (direction Storo) from city center (Nationaltheatret or Stortinget stations) or from Majorstuen.
2. Get off at Blindern.
3. Follow the signs.

Conference Site

The conference is in two buildings on the campus:

- Georg Sverdrup house (GS) - large black building with banner
- Vilhelm Bjerknes house (VB) – left of Georg Sverdrups house

Registration

Registration is in the main hall of Georg Sverdrups house and is open

Monday – Wednesday: 8:00 – 17:00
Thursday – Friday: 8:30 – 12:30
# Conference Overview

## Tutorial and Workshop Program

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th></th>
<th>Tuesday</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tutorials</td>
<td>Workshops</td>
<td>Tutorials</td>
<td>Workshops</td>
</tr>
<tr>
<td>9:00</td>
<td>TN1 VB-C309</td>
<td>WS1 VB-C313</td>
<td>TE1 VB-Aud.4</td>
<td>WS3 VB-C313</td>
</tr>
<tr>
<td></td>
<td>TS1 VB-Aud.4</td>
<td>WS2 VB-C315</td>
<td>TG1 VB-Aud.2</td>
<td>WS6 GS-11</td>
</tr>
<tr>
<td></td>
<td>TU1 VB-Aud.2</td>
<td>WS4 GS-8</td>
<td>TJ1 VB-C309</td>
<td>WS8 GS-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WS5 GS-1</td>
<td>TL1 VB-C311</td>
<td>WS10 GS-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WS7 GS-2</td>
<td>TW1 VB-C305</td>
<td>WS11 VB-C315</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WS9 GS-5</td>
<td>TM1 VB-C305</td>
<td>WS12 GS-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WS10 GS-3</td>
<td>TP1 VB-C303</td>
<td>WS13 GS-5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WS14 VB-C317</td>
<td></td>
<td>WS16 VB-C317</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WS20 GS-11</td>
<td></td>
<td>WS18 GS-8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WS23 VB-C319</td>
<td></td>
<td>WS21 VB-Aud.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WS22 VB-C319</td>
</tr>
<tr>
<td>10:30</td>
<td>Coffee Break</td>
<td></td>
<td>Coffee Break</td>
<td></td>
</tr>
<tr>
<td>11:00</td>
<td>Tutorials cntd.</td>
<td>Workshops cntd.</td>
<td>Tutorials cntd.</td>
<td>Workshops cntd.</td>
</tr>
<tr>
<td>12:30</td>
<td>Lunch</td>
<td></td>
<td>Lunch</td>
<td></td>
</tr>
<tr>
<td>13:30</td>
<td>TN2 VB-C305</td>
<td>Workshops cntd.</td>
<td>TE2 VB-C311</td>
<td>Workshops cntd.</td>
</tr>
<tr>
<td></td>
<td>TS2 VB-Aud.4</td>
<td></td>
<td>TG2 VB-Aud.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TU2 VB-C309</td>
<td></td>
<td>TJ2 VB-C305</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TW2 VB-C311</td>
<td></td>
<td>TL2 VB-C309</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TD1 VB-Aud.2</td>
<td></td>
<td>TD2 VB-Aud.4</td>
<td></td>
</tr>
<tr>
<td>15:00</td>
<td>Coffee Break</td>
<td></td>
<td>Coffee Break</td>
<td></td>
</tr>
<tr>
<td>17:30</td>
<td>Welcome Reception</td>
<td></td>
<td>Eclipse Technology eXchange Reception hosted by IBM</td>
<td></td>
</tr>
<tr>
<td>20:00</td>
<td>Exhibition Opening</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday</td>
<td>Thursday</td>
<td>Friday</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------</td>
<td>---------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>09:00</strong></td>
<td>Invited talk</td>
<td>Invited talk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opening</td>
<td>Functional Objects</td>
<td>Rich Interfaces for Software Modules</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trygve Reenskaug</td>
<td>Matthias Felleisen</td>
<td>Thomas A. Henzinger</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:30</td>
<td>Invited talk</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:30</td>
<td>Functional Objects</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:00</td>
<td>Programming Concepts I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PR1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:00</td>
<td>Programming Concepts II</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PR2 D1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:00</td>
<td>Program Analysis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PR2 D1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:00</td>
<td>Lunch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13:30</td>
<td>Software Engineering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PR3 D2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14:00</td>
<td>Coffee Break</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14:30</td>
<td>Coffee Break</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:30</td>
<td>Coffee Break/Poster Session</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:00</td>
<td>Aspects</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PR4 D3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:30</td>
<td>Verification</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PR8 D3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17:00</td>
<td>Panel: OOP and XML</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18:00</td>
<td>Reception</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>City Hall of Oslo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19:00</td>
<td>Fjord cruise on the Oslo Fjord</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19:30 – 21:15</td>
<td>Banquet at Gamle Logen</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Banquet Speaker</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Douglas C. Engelbart</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The opening, the invited talks, the paper sessions and the panel are in **GS-Aud.1**: the main auditorium in the main building.

Practitioners’ Reports are in **GS-Aud.2**, the auditorium to the right of the main auditorium.

Demonstrations are in **GS-1**, at the third floor of the main building.
Conference Site

Georg Sverdrup (GS) – ground floor

Georg Sverdrup (GS) – 3rd floor

Tutorial and Workshop rooms
Invited Talks

Functional Objects
Matthias Felleisen, Northeastern University, Boston, Massachusetts
Thursday, 17 June 9:00 – 10:00
At first glance, object-oriented programming has little or nothing in common with functional programming. One grew out of the procedural tradition, providing means for representing real-world objects and mechanisms for encapsulating state. Computing proceeds via method calls. The other is a radical departure from conventional programming. It emphasizes a(n almost) purely mathematical approach. Programmers design systems of algebraic datatypes and functions, and a computation is the evaluation of an expression. Still, nobody can overlook the similarities of the two approaches considering the development of design patterns and notions of effective object-oriented programming practices.

In my talk, I will compare and contrast the two ideas of programming and programming language design. I will present and defend the thesis that good object-oriented programming heavily "borrows" from functional programming and that the future of object-oriented programming is to study functional programming and language design even more.

Matthias Felleisen’s career consists of two parts. For the first ten years (1984–1994), he focused on the development of a new form of operational semantics and used this semantics to study design issues in mostly functional programming languages. His form of operational semantics, often dubbed evaluation context semantics, has become the standard tool for studying the well-definedness of programming languages (aka type soundness theorem). His work on continuation-based control constructs and calculi of control has spawned small areas of investigation in both control constructs and logic.

In 1994, Felleisen and his research group (PLT) began to work on the development of a programming environment for novice programmers (DrScheme). They use this software (and a curriculum they developed in parallel) to inject true design principles into the introductory programming curriculum. They use the software development project to study problems in programming languages, software engineering, and operating systems. Over the past ten years, Felleisen and his collaborators have published numerous papers on object-oriented design patterns, the nature of classes and mixins, the interaction between classes and modules, extensibility in functional and OO programs, and other matters of objects.

Felleisen spent most of his career at Rice University, with short sabbaticals at Carnegie Mellon University (Pittsburgh) and Ecole Normale Superieure (Paris). He is now a Trustee Professor at Northeastern University, Boston.

Rich Interfaces for Software Modules
Thomas A. Henzinger, University of California, Berkeley and EPFL
Friday, 18 June 9:00 – 10:00
Interfaces play a central role in the modular design of systems. Good interface design is based on two principles. First, an interface should expose enough information about a module so to make it possible to predict if two or more modules work together properly, by looking only at their interfaces. Second, an interface should not expose more information about a module than is required by the first principle. The technical realization of these two principles depends, of course, on the precise interpretation of what it means for two or more modules to “work together properly.” A simple interpretation is offered by typed programming languages: a module that implements a function and a module that calls that function are compatible if the function definition and the function call agree on the number, order, and types of the parameters.

We present richer notions of interfaces, which expose in addition to type information, also temporal information about software modules. For example, the interface of a file server with the two methods open file and read file may stipulate that read file must not be called before open file has been called. Symmetrically, then, the interface of a client must specify the possible sequences of open file and read file calls during its execution, so that a compiler can check if the server and the client fit together. Such behavioral interfaces, which expose temporal information about a module and at the same time impose temporal requirements on the environment of the module, can be specified naturally using an automaton-based language [1, 2]. In other situations, the appropriate notion of compatibility between software modules, as suggested by the first principle of interface design, is richer still and may require, for example, the exposure of assertional, real-time, and resource-use information. This leads, in turn, to push-down, timed, and resource interfaces [3, 4, 5]. For instance, resource interfaces can be used to ensure that no two modules simultaneously access a unique resource.

We formally capture the requirements on interfaces by axiomatizing interface theories [6]. For example, the axiom of “independent implementability” of interfaces guarantees that if A and B are compatible interfaces, and A0 is a module that conforms to interface A, and B0 is a module that conforms to interface B, then the composition A0||B0 of the two modules conforms to the composite interface A||B. For selected interface
formalisms, such as behavioral, push-down, timed, and resource interfaces, we show that they satisfy the
axioms of interface theories, and we discuss the following three algorithmic problems:
1. Compatibility: given two interfaces, are they compatible?
2. Conformance: given an interface and a software module, does the module conform to the interface?
3. Interface extraction: given an interface theory and a software module, what is the interface of the module
with respect to the theory?
In particular, we show that the compatibility checking of interfaces amounts to solving a game in which the
interfaces and the unknown environment represent players. Furthermore, we show that the conformance
relationship between a module and its interface must be a contravariant one, which as in subtyping, treats
inputs and outputs differently. This distinguishes interface conformance from many formal methods for
stepwise refinement.
The work reported here is joint with Arindam Chakrabarti, Luca de Alfaro, Marcin Jurdiński, Freddy Mang,
and Marielle Stoelinga.

Thomas A. Henzinger is a Professor of Electrical Engineering and Computer Sciences at the
University of California, Berkeley. He holds a Dipl.-Ing. degree in Computer Science from Kepler
University in Linz, Austria, an M.S. degree in Computer and Information Sciences from the
University of Delaware, and a Ph.D. degree in Computer Science from Stanford University
(1991). He was an Assistant Professor of Computer Science at Cornell University (1992–95),
and a Director of the Max-Planck Institute for Computer Science in Saarbruecken, Germany
(1999).
His research focuses on modern systems theory, especially formalisms and tools for the
component-based and hierarchical design, implementation, and verification of embedded, real-
time, and hybrid systems. His HyTech tool was the first model checker for mixed discrete-
continuous systems.

Banquet Speech

Facilitating the Evolution of our Collective IQ
Douglas C. Engelbart, PhD, Director, Bootstrap Institute
Thursday, 17 June  20:00

Mankind’s significant problems are growing larger and more complex at an accelerating rate. Technologies
that offer aid in their solutions are emerging, also at an accelerating rate -- and they themselves are
introducing further complexity into our world.
And note: complex, urgent problems have to be dealt with collectively!
I happen to believe that if we don't increase our collective capabilities SIGNIFICANTLY -- for recognizing,
understanding, and coping with complex, urgent problems -- much of our civilization will be at risk of crashing and burning.
The scale of this challenge is so large that it can only be pursued directly and effectively by following an
appropriately scaled improvement strategy.
That’s what I’ll be describing -- my proposed strategy. Call it "Bootstrapped, Facilitated Evolution".
And then I’ll ask why the OOP people, who seem to have developed such a superior way to deal with
information objects, haven’t already solved this. Hmm, perhaps they’ll join the pursuit?

On December 9, 1968, Douglas C Engelbart and a team of software developers gave the first
public demonstration of a computer with a windows interface, videoconferencing, black on white
text, context-sensitive help, and a mouse. They delivered this demo to 4000 stunned spectators
at the Falls Joint Computer Conference in San Francisco. Although the demo system linked to a
remote mainframe computer, it sparked research that led to the development of personal
computers, the graphical user interface, and more-advanced networks. It launched a disruptive
revolution in the way people work, communicate and produce. If not for Douglas Engelbart, many
of the technical innovations we consider vital to the personal computer revolution would not exist.
Technical Program

Opening
Trygve Reenskaug, *University of Oslo*
Wednesday, 16 June  9:00 – 09:30  GS-Aud.1

Programming Concepts I
*Chair: Gilad Bracha*
Wednesday, 16 June  9:30 – 10:30  GS-Aud.1

- **Ownership Domains: Separating Aliasing Policy from Mechanism**

- **Composable Encapsulation Policies**
  Nathanael Schärli, Stéphane Ducasse, and Oscar Nierstrasz, *University of Bern*, and Roel Wuyts, *Université Libre de Bruxelles*

Program Analysis
*Chair: Martin Odersky*
Wednesday, 16 June  11:00 – 12:30  GS-Aud.1

- **Demand-Driven Type Inference with Subgoal Pruning: Trading Precision for Scalability**
  S. Alexander Spoon and Olin Shivers, *Georgia Institute of Technology*

- **Efficiently Verifiable Escape Analysis**
  Matthew Q. Beers, Christian H. Stork and Michael Franz, *University of California Irvine*

- **Pointer Analysis in the Presence of Dynamic Class Loading**
  Martin Hirzel and Amer Diwan, *University of Colorado*, and Michael Hind, *IBM Watson Research Center*

Software Engineering
*Chair: Erik Ernst*
Wednesday, 16 June  13:30 – 15:00  GS-Aud.1

- **The Expression Problem Revisited — Four new solutions using generics**
  Mads Torgersen, *University of Aarhus*

- **Rewritable Reference Attributed Grammars**
  Torbjörn Ekman and Görel Hedin, *Lund University*

- **Finding and Removing Performance Bottlenecks in Large Systems**
  Glenn Ammons, Jong-Deok Choi, and Manish Gupta, *IBM*, and Nikhil Swamy, *University of Maryland*

Aspects
*Chair: Mira Mezini*
Wednesday, 16 June  15:30 – 17:00  GS-Aud.1

- **Programming With Crosscutting Effective Views**
  Doug Janzen and Kris De Volder, *University of British Columbia*

- **AspectJ2EE = AOP + J2EE — Towards an Aspect Based, Programmable and Extensible Middleware Framework**
  Tal Cohen and Joseph (Yossi) Gil, *Israel Institute of Technology*

- **Use Case Level Pointcuts**
  Jonathan Sillito, Christopher Dutchyn, Andrew David Eisenberg and Kris De Volder, *University of British Columbia*
Middleware
Chair: Peter Dickman
Thursday, 17 June 10:30 – 12:30  GS-Aud.1

- Inheritance-Inspired Versioning for CORBA
  Skef Iterum, Shoreline, and Ralph Campbell, Sun Microsystems

- A Middleware Framework for Persistence and Querying of Java Objects
  Mourad Alia, Sébastien Chassande-Barrioz, Pascal Déchamboux, Catherine Hamon, and Alexandre Lefebvre, France Télécom R&D

- Sequential Object Monitors
  Denis Caromel, Université de Nice, Luis Mateu, University of Chile, and Eric Tanter, Ecole des Mines de Nantes

- Increasing Concurrency in Databases using Program Analysis
  Roman Vitenberg, Kristian Kvilekval and Ambuj K. Singh, UCSB Santa Barbara

Programming Concepts II
Chair: Doug Lea
Thursday, 17 June 13:30 – 15:00  GS-Aud.1

- Semantic Casts: Structural Subtyping in a Nominal World
  Robert Bruce Findler, University of Chicago, Matthew Flatt, University of Utah, and Matthias Felleisen, Northeastern University, Boston

- LOOJ: Weaving LOOM into Java
  Kim B. Bruce, Williams College, and Nathan Foster, University of Pennsylvania

- Modules with Interfaces for Dynamic Linking and Communication
  Yu David Liu and Scott F. Smith, The Johns Hopkins University

Verification
Chair: Clemens Szyperski
Thursday, 17 June 15:30 – 17:00  GS-Aud.1

- Early identification of incompatibilities in multi-component upgrades
  Stephen McCamant and Michael D. Ernst, Massachusetts Institute of Technology

- Typestates for Objects
  Robert DeLine and Manuel Fähndrich, Microsoft Research

- Object Invariants in Dynamic Contexts
  K. Rustan M. Leino, Microsoft Research, and Peter Müller, ETH Zurich

Systems
Chair: Urs Hoelzle
Friday, 17 June 10:30 – 12:30  GS-Aud.1

- Transactional Monitors for Concurrent Objects
  Adam Welc, Suresh Jagannathan and Antony L. Hosking, Purdue University

- Adaptive Tuning of Reserved Space in an Appel Collector
  José Manuel Velasco, Antonio Ortiz, Katzalin Olcoz, and Francisco Tirado, Complutense University

- Lock Reservation for Java Reconsidered
  Tamiya Onodera, Kiyokuni Kawachiya and Akira Koseki, IBM Research, Tokyo Research Lab

- Customization of Java Library Classes using Type Constraints, and Profile Information
  Bjorn De Sutter, Ghent University, and Frank Tip and Julian Dolby, IBM T.J. Watson Research Center
Panel

Object-oriented Programming and XML
Thursday, 17 June 17:00 – 18:00  GS-Aud.1
Panel Chair: Clemens Szyperski, Microsoft.
Panelists:  
Erik Meijer, Microsoft, USA  
Alain Frisch, École Normale Supérieure, France  
Vivek Sarkar, IBM, USA

The panel focuses on recent work, performed concurrently in multiple projects, to integrate XML concepts into OO languages. Examples of such integration efforts include CDuce, Xacr, Xduce, Xen, XJ, and XML/Scala. The key challenges faced by these projects range from appropriate type systems, to the proper integration of query semantics, to the meeting of high performance bars.

The panelists will address questions such as: Where is OOP-XML convergence headed - at the level of languages, type systems, semantics? What are the biggest open issues / research problems? What are interesting practical applications of combined OOP-XML approaches?

Practitioners' Reports

- **PR1 Keynote: Application Development Scenario - AD at the Crossroads**  
Peter Hidas, Gartner Norway  
Wednesday, 16 June  9:30 – 10:30  GS-Aud.2

  From 2003 to 2006, we see little innovation making its way into the enterprise's portfolio of application software. Why? In 2003, new application software approaches such as services-based applications and business process fusion were still in their infancy - few utilities or standards were in place to enable the infrastructure to support these approaches, and, thus, applications must wait on infrastructure. Application software vendors are among the enterprises that depend on reliable infrastructure to build and deploy their applications - most vendors will rearchitect their platform and applications during this period. By 2007, activity will begin in earnest to implement the next generation of applications.

- **PR2 A New Approach to Solutions Design and Systems Engineering**  
Frank Lillehagen, Computas  
Wednesday, 16 June  11:00 – 12:00  GS-Aud.2

  Enterprise and solutions design, and applications supporting the creative phases of project work has been a theme for IT providers since the mid-eighties. Many promises of providing solutions have been made, such as CAD, PDM and ERP systems, but so far industry has been left with disappointment and unsatisfactory solutions. This paper will revise the industrial challenges, present the Active Knowledge Modeling Approach to Enterprise Design and Development (ED&D), and present a way ahead to enable Model Designed Solutions.

- **PR3 Generating Service Descriptions and Code from Reusable Models**  
Jens M. Glattetre, SuperOffice ASA, Marek Vokac, Simula Research Laboratory  
Wednesday, 16 June  13:30 – 14:30  GS-Aud.2

  Models tend to change relatively slowly, and they have a longer lifetime than the storage platform or technology they are implemented on at any given time. Hence it is desirable to generate platform- and technology specific data definitions and persistence layers from model descriptions as technology changes over time. We have implemented a code generator in C# that combines plain-text templates and model data to achieve a high level of reuse, by concentrating on reusing descriptions instead of concrete, platform-dependent implementations. We have successfully generated a C# persistence layer for a legacy database, and also applied this principle to descriptions of high-level functional services. Service access technology, e.g. COM, CORBA or Web Services varies on a shorter timescale than do the services themselves. Our code generator has been used to generate service descriptions, WSDL, COM IDL and C# implementation shells, based on platform-independent definitions in UML. We have also successfully used it as the basis for multiple commercial products within the same product family/line.

- **PR4 Creating an Engine for Data-Centric Collaboration**  
Stian Bakken, Xymphonic Systems, Ole J. Anfindsen, EPM Technology AS  
Wednesday, 16 June  15:30 – 16:30  GS-Aud.2

  Xymphonic Systems AS has a patented transactional technology which is very attractive for data-centric collaboration. During the last half of 2003, a new version of our server-side program, Xymphonic Engine, was planned and implemented. In essence, Xymphonic Engine is responsible for providing transactional control on
interactions between project participants on shared data. The presentation will focus on how a theoretical model for long duration transactions was turned into an object-oriented implementation.

- **PR5 FAST Data Search: Object-Oriented Programming in a High Performance Distributed Application**
  Eirik Knudsen, Fast Search & Transfer ASA
  Thursday, 17 June  10:30 – 11:30  GS-Aud.2

- **PR6 DocuLive – an Object-Oriented Patient Record System**
  Gunvald Harket, Jo Piene, Sigbjørn Skjervold, Lars Tafjord, Siemens Health Services
  Thursday, 17 June  11:30 – 12:30  GS-Aud.2
  Crucial requirements for a patient record system are security, availability, efficiency, version-control, adaptability and ease-of-use. A typical DocuLive installation has 5000 work stations, 100 GigaByte data, 100 000 patient records, 10 000 users, 1000 simultaneous users, and all with a single database server. The development of DocuLive led to a generic object-oriented design. The project team's background from Simula, combined with an understanding of the domain, led to a design where objects are defined and handled by a run-time-system, and a Class Dictionary, that are all implemented in a relational database. A generic client handles objects in a similar way as Web-browsers handle Web-pages. DocuLive objects comprise structured tagged information (like XML), a default presentation (like HTML), multi-user editing with transaction control, versioning and read/write access control.

- **PR7 Using ISO 10303-11, Express, the Information Modeling Language, to create Advanced Solutions for Industrial Data Applications**
  Kjell Bengtsson, EPM Technology
  Thursday, 17 June  13:30 – 14:30  GS-Aud.2
  The presentation will give highlights from projects in Aerospace, Defence and the Building sector, where requirements, granularity and rules are important to manage complex systems and projects. One demo from the Building Control Authority in Singapore will be given, where an Automated Building Plan Check System uses ISO object definitions.

- **PR8 Experiences from introducing a component oriented UML-based Development Method**
  Kai Hansen, ABB Corporate Research Center, Bente Anda, Simula Research Laboratory, Ingolf Gullesen, ABB Corporate Research Center, and Hanne Kristin Thorsen, Department of Informatics, University of Oslo.
  Thursday, 17 June  15:30 – 16:30  GS-Aud.2
  The experience from introducing a UML based component oriented design methodology into a large project which develops software and hardware for the industrial automation market. Keywords: UML, safety system, software development, hardware development

- **PR9 Testing framework for Platform Independent Models with TTCN-3**
  Gabor Batori, Domonkos Asztalos, Software Engineering Group, Ericsson Hungary Ltd.
  Friday, 18 June  10:30 – 11:30  GS-Aud.2
  MDA supporting tools provide only limited means for describing complex model level test procedures. Hence a framework for model testing is indispensable. This presentation considers how to assist the analysis level test development with TTCN-3 (the Testing and Test Control Notation version 3). With the help of a model compiler we can ease and partly automate the test development process.

- **PR10 MDA as a framework for supporting Enterprise standards**
  Carl-Olav Løvold, Chief consultant, Mesan, Norway
  Friday, 18 June  11:30 – 12:30  GS-Aud.2
  MDA provides an approach for developing and maintaining applications with a basis in models. In addition to benefits with respect to productivity, changeability and management of technology complexity, this is also a useful tool for supporting use of enterprise standards for design, architecture and quality. The presentation focuses on experiences from using MDA, and how a software architect can make use of this, exemplified by the use of the Compuware OptimalJ tool. The presented project provides experiences on the use of MDA transformation technology to support the use of SWING and design patterns.
Demonstrations

**D1: Advanced Refactorings in Eclipse**

Robert Fuhrer, Frank Tip, *IBM T.J. Watson Research Center*, Adam Kiezun, *MIT*

**Wednesday, 16 June 11:00 – 12:00 GS-1**

**Thursday, 17 June 10:30 – 11:30 GS-1**

We propose a demonstration of several advanced refactorings for Java implemented for the Eclipse IDE (www.eclipse.org). These refactorings are semantics-preserving program transformations typical of the transformations object-oriented programmers perform manually to improve the structure of existing programs, and promote reuse, clarity, and extensibility.

This demonstration consists of three parts: (i) a review of Eclipse’s implementations of several common refactorings, including Extract Method, Introduce Factory and Change Method Signature, described in Fowler’s seminal book on refactoring, (ii) a demonstration of generalization-related refactorings such as Extract Interface, which extracts an interface from a class to reduce dependencies on implementation particulars, and Generalize Type, for remediating over-specific declarations, and (iii) a demonstration of Genericize Container Uses, a refactoring in development for migrating uses of standard container classes (e.g. ArrayList or HashMap) to their generic counterparts in Java 1.5 (e.g., ArrayList<String>). This determines the element types that flow into containers, modifies declarations and allocations where possible, and removes casts rendered redundant.

The latter two categories require non-trivial static analysis, and present challenging software engineering usability issues, reflecting complications that programmers face in performing such transformations manually. Therefore, each demonstration will include a brief overview of the technical challenges that the refactoring presents, and highlights of the approach we took in addressing them. In particular, we briefly describe a new piece of Eclipse infrastructure designed to help implement various kinds of context-sensitive and context-insensitive type and pointer analyses for Java.

**D2: A Web Services Management Layer for the selection, integration and management of web services**

María Agustina Cibrán, Bart Verheecke, Davy Suvée, Wim Vanderperren, *Vrije Universiteit Brussel*

**Wednesday, 16 June 13:30 – 14:30 GS-1**

**Thursday, 17 June 13:30 – 14:30 GS-1**

The goal of this demonstration is to illustrate the Web Services Management Layer (WSML), an intermediate layer in between the client applications and the web services. The WSML aims at decoupling client applications from specific web services allowing the just in time integrations of services: new services that were not foreseen at development time can be dynamically integrated into applications. By weakening the link between the client applications and the services, hot-swapping functionality between services can be installed to dynamically swap to other semantically equivalent services in case of unavailability. In addition, this mechanism is extended to consider non-functional properties of services in the selection. The WSML also deals with several client-side management concerns such as caching, billing, accounting, security, etc. We observe that service management concerns crosscut the core applications. At the same time, a technology that can cope with the volatile world of the web services environment is required.

Aspect Oriented Programming (AOP) aims to cleanly modularize crosscutting concerns of a system. In this demo we show how dynamic AOP is ideal to implement the core functionality of the WSML and achieve a clean encapsulation of the service management concerns at the same time as achieving the desired flexibility and adaptability. For the implementation of the WSML the dynamic aspect oriented language JAsCo is adopted.

**D3: OptimalJ - Model-Driven, Pattern-Based Development**

Bastiaan Schönhage, *Compuware Europe B.V.*

**Wednesday, 16 June 15:30 – 16:30 GS-1**

**Thursday, 17 June 15:30 – 16:30 GS-1**

OptimalJ is a software development tool that supports Model-Driven, Pattern-Based development. It is based on OMG’s MDA (Model Driven Architecture).

In OptimalJ, the software developer starts with a high-level platform independent model that defines the application from a conceptual perspective. Based on these high-level models, OptimalJ can automatically generate J2EE models, including EJBs and JSPs. At this level, developers can add platform specific knowledge and model the integration with databases or external entities such as webservice or CORBA applications. Finally, based on the platform specific application models, Java code is generated that is used as the basis for the working J2EE application. Since OptimalJ supports incremental development, changes in the
models are incorporated in dependent models as well as in the generated code while, of course, leaving the code added by the developer untouched.

In this demonstration, the OptimalJ product itself will be demonstrated to illustrate how OMG's MDA can be implemented in a development tool. Some of the underlying technology and concepts will be discussed. Additionally, we will see how the architecture of OptimalJ allows developers to write their own "implementation patterns" to plug in new technologies based on the available models.

**Posters**

**Thursday 15.00-15.30**

- **Axiomatic Mathematics meets Object Oriented Programming**  
  Marc Conrad, Tim French, Carsten Maple, *University of Luton*

- **SICSDA: an Adaptive Configurable Distributed Software Architecture Applied to Satellite Control Missions**  
  Adriana Cursino Thomé, *Positivo Universitarian Center (UNICENP)*  
  Mauricio G. V. Ferreira, *National Institute for Space Research (INPE)*  
  João Bosco S. Cunha, *School of Engineering of Itajubá (UNIFEI)*

- **A UML profile for service-oriented architectures**  
  Amir Zeid, Rafik Amire, *The American University in Cairo*

- **Using Design Patterns to Abstract a Software Architecture for Natural Language Generation**  
  Raquel Hervás, Pablo Gervás, *Universidad Complutense de Madrid*

- **UnderFrame: Understanding Object-Oriented Frameworks Using a Case-Based Teaching Approach**  
  Guillermo Jimenez-Diaz, Mercedes Gomez-Albarran, Pedro Gonzalez-Calero, *Departamento de Sistemas Informaticos y Programacion. Universidad Complutense de Madrid*

- **CUP: A CORBA-based Unified Process Adaptation**  
  Nelly Bencomo, *Lancaster University*  
  Alfredo Matteo, *Universidad Central de Venezuela*

- **Domain Modeling in Self Yields Warped Hierarchies**  
  Ellen Van Paesschen, Wolfgang De Meuter, Theo D'Hondt, *Vrije Universiteit Brussel - Programming Technology Lab*

- **Learning object-oriented concepts mediated through collaboration and use of artifacts**  
  Jan Dolonen, Annita Fjuk, Ola Berge, *InterMedia, University of Oslo, Norway*  
  Jens Bennedsen, *IT University West, Denmark*

- **The AspectIX Middleware Platform**  
  Franz J. Hauck, *University of Ulm*  
  Rüdiger Kapitza, *Univ. of Erlangen-Nürnberg*  
  Hans P. Reiser, *Univ. of Erlangen-Nürnberg*  
  Andreas I. Schmied, *University of Ulm*. 
Tutorials

Developing Secure and Critical systems

- **TS1: Building secure software applications**,  
  Frank Piessens, Wouter Joosen, Bart De Win, *University of Leuven*  
  Monday, 14 June 09:00 – 12:30  VB-Aud.4  
  Most software developed today runs in an Internet-connected, and hence possibly hostile environment. As a consequence, it is generally recognized that it is important to pay attention to security aspects during the development of software.  
  However, secure software development is hard, for a number of reasons. It is hard to correctly analyze and predict potential risks for a given software product, the technologies used to counter the risks are complex, and the correct and consistent integration of these technologies in a software product is difficult.  
  This tutorial will discuss these problems and possible solutions from a software development perspective.  
  Practitioners will learn about the state of the art of secure software development. Researchers will get a deep insight in one of the most challenging non-functional requirements of today’s distributed software applications.

- **TS2: Critical (and secure) systems development with UML: methods and tools**  
  Jan Jürjens, *TU Munich*  
  Monday, 14 June 13:30 – 17:00  VB-Aud.4  
  The high quality development of critical systems (be it dependable, security-critical, real-time, or performance-critical systems) is difficult. Many critical systems are developed, deployed, and used that do not satisfy their criticality requirements, sometimes with spectacular failures.  
  Part of the difficulty of critical systems development is that correctness is often in conflict with cost. Where thorough methods of system design pose high cost through personnel training and use, they are all too often avoided.  
  UML offers an unprecedented opportunity for high-quality critical systems development that is feasible in an industrial context, because a large number of developers is trained in UML, UML is relatively precisely defined, and a number of tools are developed to assist its use.  
  The tutorial aims to give background knowledge on using UML for critical systems development and to contribute to overcoming some challenges in this context including:  
  - Adaptation to critical system application domains.  
  - Providing advanced tool-support for critical systems development with UML.  
  It includes a demo of a tool supporting critical systems analysis with UML.  
  The content is identical to WICSA tutorial TA2 (Sunday).

UML and Model Driven Architecture

- **TU1: Model-driven architecture in practice**  
  Petter Graff, *Inferdata Corporation* and Krzysztof Czarnecki, *University of Waterloo*  
  Monday, 14 June 09:00 – 12:30  VB-Aud.2  
  Today, application development still remains a laborious process with relatively little reuse and automation. Application programmers have to manually map their high-level analysis models to target platform architectures such as J2EE and .NET and eventually to code. Rather than focusing on the problem domain, they have to deal with the complex details of the target platforms. The analysis and design models – being just additional documentation artifacts - are often not properly maintained in the face of approaching deadlines. This makes the applications hard to evolve later. Retargeting an application to a new platform is almost as difficult as writing it from scratch. Model-Driven Architecture (MDA) is a framework for model-based development being standardized by the Object Management Group (OMG) that aims at solving the above-mentioned problems. In MDA, models are the primary source of an application. All other artifacts such as code, tests, and documentation are (mostly) automatically derived from models. In this tutorial, we will take a critical look at the promises made by MDA and clearly distinguish what is possible today from the visions of tomorrow.  
  After explaining basic MDA concepts such as metamodeling and model transformations, we'll discuss tool requirements and review some existing MDA tools. We'll round up the tutorial with a demo of generating a complete J2EE and .NET application from the same high-level UML model.  
  The tutorial will explain the basic technologies involved in MDA, and give examples on practical industrial usage of these.
**TU2: MDA and metamodeling**

Brian Henderson-Sellers, *University of Technology, Sydney*

Monday, 14 June 13:30 – 17:00 VB-C309

Metamodeling underpins the UML both directly and in terms of the UML Profile Mechanism. This tutorial starts by explaining why metamodeling is important in the UML context and then evaluates the various aspects of metamodeling that are known to cause both practitioners and researchers the most problems. We explore the pros and cons of strict versus loose metamodeling, the notions of physical versus logical instance-of relationships and the various flavors of the "is-a" relationship (which include instantiation and generalization). These ideas are then applied to the UML context directly in an evaluation of the extension mechanisms of the UML, particularly stereotypes, and illustrated directly in terms of the metamodels currently underpinning not only the UML but also, as further examples, the OMG SPEM model for process metamodeling. This information will then permit developers of UML profiles to more successfully utilise the metamodeling aspects and know on which occasions it is better not to use metamodeling, in addition to providing end users with a greater capability to utilize the UML extension mechanisms.

**Web Services and BPEL**

**TW1: Introduction to XML web services using .NET as an example**

(Combined with TM1 and moved to Tuesday morning)

Michael Stal, *Siemens Corporate Technology*

Tuesday, 15 June 09:00 – 12:30 VB-C305

XML Web services represent a kind of object-oriented meta middleware used for integrating heterogeneous platforms across the Internet. This technology is considered as an important integration technology by major vendors and open source communities. First organizations have started to use XML Web services for their commercial applications. The most relevant implementation platforms for XML Web services today are J2EE and .NET. As with all middleware technologies XML Web services are only trivial at the beginning. They turn out to be really complicated when developers want to build mission critical applications. The tutorial will illustrate all concepts using .NET as example. The tutorial will present all new developments in this area such as the standards specified by WS-I. A great focus will be on best practice patterns that show how to build high quality Web services applications.

**TW2: Programming internet-scale distributed applications in the 21st century: BPEL and beyond**

John Field and Vivek Sarkar, *IBM T.J. Watson Research center*

Monday, 14 June 13:30 – 17:00 VB-C311

Increasingly, commercial software is built as loosely-coupled collections of distributed components, glued together by middleware such as databases and messaging systems. Moreover, spurred in part by the advent of web services protocols (e.g., SOAP), internet-scale distributed applications are becoming more common. Such applications have a number of distinguishing characteristics, including:

- lack of a single organizational locus of control for application development
- use of asynchronous (rather than synchronous) inter-component communication
- prevalence of "long-running" transactions
- greater likelihood of system or network failure than in local area distributed applications

Until recently, there has been little work on programming models directly supporting heterogeneous, internet-scale distributed applications. In an effort to address this need, IBM, Microsoft, and BEA announced the Business Process Execution Language for Web Services specification ("BPEL4WS", or "BPEL") in mid 2002. The name is somewhat misleading—BPEL is essentially a high-level programming language, one which addresses some, but not all of the issues relevant to developing internet-scale distributed applications.

The tutorial will first discuss general issues that arise in building internet-scale applications. Next, we will give a detailed overview of BPEL with examples, showing how BPEL addresses many of these issues. The overview will cover such topics as processes, service links, activities, containers, scopes, fault handlers, and exception handlers. Finally, we will cover a number of other technologies and standards related to programming internet-scale distributed applications, and discuss their relationship to BPEL.

**Microsoft .NET**

**TN1: Introduction to .NET using C#**

Michael Stal, *Siemens Corporate Technology*

Monday, 14 June 09:00 – 12:30 VB-C309

Microsoft .NET has been developed as the successor of technologies such as COM, COM+, ADO, ASP. Microsoft .NET is based on a virtual execution engine called Common Language Runtime (CLR) which provides functionality such as garbage collection, thread synchronization, or security. .NET programs may be
written in various programming languages. They are compiled to an intermediate language, and dynamically converted to native code by the CLR. To support the development of different kinds of applications such as Web-based applications, Web-services, or fat clients the .NET Framework provides several thousand classes covering all types of functionality. While today, .NET is primarily focused on the Windows OS, implementations are currently developed by different parties to run on other operating systems as well. Thus, Microsoft .NET is increasingly competing with the Java platform and is also becoming important for software developers.

- TN2: An introduction to the Common Language Infrastructure (CLI)
  Damien Watkins, Microsoft Research
  Monday, 14 June  13:30 – 17:00  VB-C305
  A number of Component-Based Architectures (CBAs) have been developed to facilitate the interoperability of software components that are written in different programming languages. The most widely known and used of these is the OMG's Common Object Request Broker Architecture (CORBA) and Microsoft's Component Object Model (COM). Although not specifically designed to provide language interoperability, many Java-based technologies, such as Enterprise Java Beans (EJB), also adhere to the principles of a CBA. The Common Language Infrastructure (CLI) is the latest CBA to be developed and it builds on many of the lessons learned from previous architectures.

  In this tutorial, we will investigate the major components of the CLI. The CLI is an ECMA standard on which Microsoft has built three implementations: the .NET Framework, the Compact Framework and the Shard Source CLI. The CLI is also the specification on which non-Microsoft implementations, such as Ximian's Mono, are based.

Languages and Execution Environments Comparison

- TL1: Foundations of object-oriented languages: types and language design
  Kim Bruce, Williams College
  Tuesday, 15 June  09:00 – 12:30  VB-C311
  Static typing aids in earlier error detection, supports compiler optimizations, and provides information to programmers on the intended use of constructs. However, simple static-typing disciplines for object-oriented languages like C++ and Java are so restrictive that programmers are forced to bypass the type system with type casts. Other languages allow more freedom, but require run-time checking to pick up the type errors that their more permissive systems missed.

  After surveying problems with existing type systems, we explain contravariance and covariance in type systems, and suggest ways of improving the expressiveness of these systems while retaining static type safety. Constructs introduced include "MyType," "matching," and "F-bounded polymorphism." We include a brief discussion on how the type system and semantics ensure type safety. We compare the strengths and weaknesses of proposals to extend Java to support genericity based on F-bounded polymorphism, "where" clauses, match-bounded polymorphism, and virtual types, with special focus on the forthcoming extensions to Java (taken from GJ) to support parametric polymorphism.

- TL2: J2EE versus .NET
  Michael Stal, Siemens Corporate Technology
  Tuesday, 15 June  13:30 – 17:00  VB-C309
  Microsoft .NET and J2EE both offer complete platforms to build sophisticated enterprise applications without being strongly tied to the underlying operating system or hardware. They provide a complete range of APIs to build different kinds of applications such as Web-based applications, GUIs, or distributed software systems. While J2EE mainly addresses platform neutrality, the main focus of .NET currently is programming language independence. As competition drives the market, many experts anticipate J2EE and .NET to become competitors each of them gaining a significant share of the software development market. Needless to say that it is important for the programmer to understand the commonalities and differences of these platforms.

  The tutorial will introduce the .NET technology and contrast its core concepts with their Java counterparts.

Java

- TJ1: New features in J2SE 1.5
  Angelika Langer, Training & Consulting, Munich
  Tuesday, 15 June  09:00 – 12:30  VB-C309
  The upcoming release 1.5 of the Java Standard Edition (J2SE) is scheduled for summer 2004 and comes with two major additions, Concurrency Utilities and Generic Types and Methods, plus a number of minor additions to the language.

  Concurrency Utilities
The Java platform provides basic, low-level primitives for writing concurrent programs, but they are difficult to use properly. Many programs become clearer, shorter, faster, easier to write, and more reliable if higher-level synchronization constructs are used.

A comprehensive library of concurrency utilities has been developed for inclusion in the 1.5 release of J2SE. This library, known as the util.concurrent package, contains thread pools, queues, concurrent collections, special-purpose locks, barriers, and related utilities. The addition to the Java platform is substantial and will change the way we will implement concurrent Java applications. This tutorial gives an overview of the new library.

Generic Types and Methods
The addition of generic types and methods to the Java programming language has been explored since the first release of Java. Eventually, with the 1.5 release of J2SE, support for Java generics will be incorporated in the language.

A common misconception regarding generics is its comparison to C++ templates. Many people believe that Java generics are similar to C++ templates, while in fact both features have barely anything in common: C++ templates is a Turing complete language – Java generics is syntactic sugar that elides some casting. Java generics are a pure compile-time feature that is translated to non-generic Java bytecode, for reasons of compatibility with previous versions of Java.

In this tutorial, we will take a look into the proposed language feature, how it will help us improve our programs, how it is supposed to be used and where its limitations are.

More specifically, we will discuss the additions made to the language itself (parameterized types and methods as well as covariant return types), changes to the platform libraries (generic collections and extended reflection) and various exciting details of generics such as the properties of type parameters (their bounds and their scope), generic methods and type parameters inference, and the translation of generics into bytecode.

**TJ2: Developing Java applications for mobile phones: approach and new Java standard interfaces for J2ME**

Ekaterina Chtcherbina, Marquart Franz, Siemens AG
Tuesday, 15 June 13:30 – 17:00 VB-C305

The world of Java Applications for mobile phones is growing fast. The evolution has started from Java games, the topic, which is booming now. Due to the fast progress in this area, mobile phone is becoming a powerful platform that allows you to create the whole range of entertaining and business applications – games are only the beginning.

Creating Java Application for mobile phones is a challenging task: things like footprint, performance, and memory usage are critical. The tutorial will show how to design, develop, debug, optimize, and run on the platform your java applications. It will cover of course the most challenging topics. Also recommendations regarding tools will be given. We would like to discuss new Java Standard Interfaces (APIs) as they define which functionality you can access in the phone: mainly we will discuss how to use APIs, but also how you can influence the upcoming APIs. And the last but not the least, what are the current activities on the market in the area of java applications for mobile phones: related standards and key trends will be discussed.

**TE1: Contributing to Eclipse: plug-ins and how to write your own**

Kai-Uwe Mätzel, IBM Research, OTI Labs, Zurich
Tuesday, 15 June 09:00 – 12:30 VB-Aud.4

Eclipse was designed from the ground-up with extensibility in mind. A scalable plug-in mechanism and a rich set of APIs enables developers to develop and explore new tools quickly, without having to start from scratch. This tutorial illustrates the full plug-in development cycle by way of an example. You'll explore the Eclipse architecture and become familiar with the basic plug-in mechanism. With this knowledge, you'll write and debug your first plug-in using Eclipse's Plug-in and Java Development Environments. You'll add extension points to your plug-in to enable others to extend it. You will then package your extensible plug-in as a Feature and publish it with the built-in Eclipse Update Mechanism. Finally, you'll set up and manage an Eclipse Update Site, a place for other Eclipse users to explore new features as well as finding upgrades. During all these steps, you will learn the underlying Eclipse concepts and design ideas. In addition, you will receive many interesting insights on design challenges in large scale plug-in architectures. Tutorial participants are invited to bring their laptops and deepen their understanding by implementing practical exercises.

Part I: Contributing to Eclipse: Plug-ins and how to write your own. (Part II continues in TE2).

**TE2: Contributing to Eclipse: advanced plug-in development**

Kai-Uwe Mätzel, IBM Research, OTI Labs, Zurich
Tuesday, 15 June 13:30 – 17:00 VB-C311
Part II adds advanced plug-in development to the knowledge provided by TE1, with focus on advanced plug-in development, covering how to enable others to contribute to your plug-in and turning your plug-in into a stand-alone application.

Tutorial participants are invited to bring their laptops and deepen their understanding by implementing practical exercises.

Design and Software Development

- **TD1: Design by contract and automatic verification for Java with JML and ESC/Java2**
  Joseph Kiniry and Erik Poll, *University of Nijmegen* and David Cok, *Eastman Kodak Company*
  **Monday, 14 June  13:30 – 17:00  VB-Aud.2**
  This tutorial introduces the new ESC/Java version 2 and the JML annotation language.
  The Java Modeling Language (JML) is a behavioral interface specification language that can be used to specify the behavior of Java modules. It combines the Design by Contract approach of Eiffel and the model-based specification approach of the Larch family of interface specification languages, with some elements of the refinement calculus. JML has a Java-based syntax and semantics, thus is easy to learn for Java programmers.
  ESC/Java2 is a tool that checks that a program is consistent with its annotation. It also detects, at compile time, common programming errors that ordinarily are not detected until run time, and sometimes not even then; for example, null dereference errors, array bounds errors, type cast errors, and race conditions. While ESC/Java uses a theorem prover, it feels to a programmer more like a powerful type checker.
  Because JML is familiar to Java programmers, and ESC/Java2 just feels like a typechecker, we believe that they are an excellent way to gently introduce programmers to formal methods.

- **TD2: Adaptive object-model architecture: how to build systems that can dynamically adapt to changing requirements**
  Joseph Yoder, *The Refactory, Inc.*
  **Tuesday, 15 June  13:30 – 17:00  VB-Aud.4**
  Architectures that can dynamically adapt to changing requirement have been called “reflective” or “meta” architectures. We call a particular kind of reflective architecture an “Adaptive Object-Model (AOM)” architecture. An AOM is a system that represents classes, attributes, relationships, and behavior as metadata. AOMs store their Object-Model in XML files or in a database and interprets it. Consequently, the object model is adaptive; when the descriptive information for the object model is changed, the system immediately reflects those changes. Many architects of a system with Adaptive Object-Models often claim this is the best system they have ever created primarily due to their flexibility, power, and eloquence. These systems are related to Model-Driven Architectures and Generative Programming Techniques. This tutorial will give a description of the Adaptive Object-Model architectural style and will make it easier for developers to understand and build systems that need to adapt to changing requirements.

Generative Programming and UML 2.0 Testing

- **TG1: Generative programming: methods, techniques, and applications**
  Krzysztof Czarnecki, *University of Waterloo*
  **Tuesday, 15 June  09:00 – 12:30  VB-Aud.2**
  System family engineering seeks to exploit the commonalities among systems from a given problem domain while managing the variabilities among them in a systematic way. In system family engineering, new system variants can be rapidly created based on a set of reusable assets (such as a common architecture, components, models, etc.).
  Generative programming aims at modeling and implementing system families in such a way that a given system can be automatically generated from a specification written in a domain-specific language. In this tutorial, participants will learn how to perform domain analysis (i.e., identify and document the commonalities and variabilities within a system family using feature modeling) and domain design (i.e., develop a common architecture for a system family), and implement program generators using a number of different technologies, such as template-based code generation, C++ template metaprogramming, and model transformations.

- **TG2: The UML 2.0 testing profile — the approach for test design and generation in UML-based development processes**
  Ina Schieferdecker, *Technical University Berlin/Fraunhofer, FOKUS*, Øystein Haugen, *UiO*
  **Tuesday, 15 June  13:30 – 17:00  VB-Aud.2**
  Testing often accounts for more than 50% of the required effort during system development. However, testing is frequently not well integrated with other development phases. One reason for this is that designers, developers and testers all use different languages and tools, making it difficult to communicate and exchange
documents. The UML Testing Profile bridges the gap between designers and testers by providing a means for using UML for test specification and modeling. This allows the reuse of UML design documents for testing and enables test development in an early system development phase. The testing profile provides support for UML based model-driven testing.

Middleware and Distribution

- **TM1: Distributed .NET**
  (Combined with TW1, e.g. includes now also material on XML Web Services)
  Michael Stal, *Siemens Corporate Technology*
  **Tuesday, 15 June 09:00 – 12:30 VB-C305**
  Almost all software systems today are distributed, either leveraging middleware technologies or Internet protocols for communication. As a consequence, platforms such as .NET spend a significant bunch of functionality for the development of distributed and Web-based systems. However, there is not such a thing like a common distributed component-based software architecture. Different distributed applications must balance different forces with respect to non-functional requirements such as scalability, communication paradigms, security, bandwidth, to name just a few. This is the reason why .NET offers a whole range of different approaches for communication and distribution. Examples will be illustrated using the programming language C#. Topics covered include Web-services, Sockets, .NET Remoting, COM+, ADO.NET, Indigo.

Project Management

- **TP1: Seeing the forest and the trees — using system dynamics for high-level decision and project improvement**
  Martine Devos, *Avayalabs, Avaya Inc.*
  **Tuesday, 15 June 09:00 – 12:30 VB-C303**
  Despite demonstrated benefits of object technology, agile methods, software product line engineering and quality improvement projects, many improvement programs end in failure. Companies do not manage to sustain effort in even initially successful projects; practitioners lack techniques to build strong cases for their superior technology. Often improvement programs -- even successful programs -- worsen the business performance and lead to lay-offs and degeneration of morale. System dynamics and systems thinking can help us understand this "improvement paradox". We use two cases of improvement efforts (product line engineering and agile development) to look at the dynamic process that is core in project success. Failure to account for the dynamics and feedback among tightly coupled decisions and activities lead to unexpected side effects. We learn and USE techniques to visualize dynamics that will help managers and developers build sustainable project environments and improvement programs.
Workshops

- **WS1: 2nd European Workshop on Web Services and Object Orientation (EOOWS 2004)**
  Monday 14 June 09:00-17:00 VB-C313

- **WS2: Practical Problems of Programming in the Large Workshop**
  Ralf Reussner, Wolfgang Weck.
  Monday 14 June 09:00-17:00 VB-C315

- **WS3: 8th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2004)**
  Coral Calero, Fernando Brito e Abreu, Geert Poels, Houari A. Sahraoui.
  Tuesday 15 June 09:00-17:00 VB-C313

- **WS4: Eighth Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts**
  Jürgen Börstler, Isabel Michiels, Annita Fjuk, Jens Kaasbøll.
  Monday 14 June 09:00-17:00 GS-8

- **WS5: 2nd Workshop on Object-Oriented Language Engineering for the Post-Java Era: Back to Dynamicity**
  Wolfgang De Meuter, Pascal Costanza, Stéphane Ducasse, Richard Gabriel, Theo D'Hondt.
  Monday 14 June 09:00-17:00 GS-1

- **WS6: Philosophy, Ontology, and Information Systems**
  Tuesday 15 June 09:00-17:00 GS-11

- **WS7: Communication Abstractions for Distributed Systems**
  Antoine Beugnard, Eric Jul, Laurence Duchien, Ludger Fiege, Robert Filman, Salah Sadou.
  Monday 14 June 09:00-17:00 GS-2

- **WS8: Formal Techniques for Java-like Programs (FTfJP)**
  Sophia Drossopoulou, Gary T. Leavens, Peter Müller, Arnd Poetzsch-Heffter, Erik Poll.
  Tuesday 15 June 09:00-17:00 GS-1

- **WS9: Component-oriented approaches to context-aware systems**
  Monday 14 June 09:00-17:00 GS-5

- **WS10: The combined 14th Workshop for PhD Students in Object-Oriented Systems and Doctoral Symposium**
  Susanne Jucknath, Eric Jul.
  Monday 14 and Tuesday 15 June 09:00-17:00 GS-3

- **WS11: Multiparadigm Programming with Object-Oriented Languages (MPOOL'04)**
  Kei Davis, Jörg Striegnitz.
  Tuesday 15 June 09:00-17:00 VB-C315

- **WS12: Mechanisms for Specialization, Generalization and Inheritance (MASPEGHI 2004)**
  Philippe Lahire, Gabriela Arévalo, Hernán Astudillo, Andrew P. Black, Erik Ernst, Marianne Huchar, Markku Sakkinen, Petko Valtchev.
  Tuesday 15 June 09:00-17:00 GS-2
- **WS13: Reflection, AOP and Meta-Data for Software Evolution (RAM-SE04)**  
  Walter Cazzola, Shigeru Chiba, Gunter Saake.  
  Tuesday 15 June 09:00-17:00 GS-5

- **WS14: First International Workshop on Coordination and Adaptation Techniques for Software Entities (WCAT'04)**  
  Carlos Canal, Juan Manuel Murillo, Pascal Poizat.  
  Monday 14 June 09:00-17:00 VB-C317

- **WS16: Model Transformation and execution in the context of MDA**  
  Jan Aagedal, Jean Bézivin, Marc Born, Philippe Desfray, Tracy Gardner, Alan Hartman, Sune Jakobson, Peter Linington, Laurent Rioux, Pete Rivett.  
  Tuesday 15 June 09:00-17:00 VB-C317

- **WS17: Interoperability of Enterprise systems**  
  Arne J. Berre, Kim Elms, Klaus Fischer, Rainer Ruggaber, Tracy Gardner, Kurt Geihs, Maarten Steen, Lea Kutvonen.  
  Monday 14 June 09:00-17:00 VB-Aud.3

- **WS18: Ninth International Workshop on Component-Oriented Programming (WCOP 2004)**  
  Jan Bosch, Clemens Szyperski, Wolfgang Weck.  
  Tuesday 15 June 09:00-17:00 GS-8

- **WS20: 10TH ECOOP Workshop on Mobile Object Systems**  
  Ciarán Bryce, Grzegorz Czajkowski, Doug Lea, Jan Vitek.  
  Monday 14 June 09:00-17:00 GS-11

- **WS21: ECOOP Workshop on Programming Languages and Operating Systems (ECOOP-PLOS 2004)**  
  Olaf Spinczyk, Andreas Gal, Michael Schoettner.  
  Tuesday 15 June 09:00-17:00 VB-Aud.3

- **WS22: 5th International Workshop on Object-Oriented Reengineering (OOR 2004)**  
  Serge Demeyer, Stéphane Ducasse, Kim Mens, Roel Wuyts.  
  Tuesday 15 June 09:00-17:00 VB-C319

- **WS23: Evolution and Reuse of Language Specifications for DSLs (ERLS)**  
  Thomas Cleenewerck, Krzysztof Czarnecki, Jörg Striegnitz, Markus Völter.  
  Monday 14 June 09:00-17:00 VB-C319
Exhibition

Opening: Monday, 14 June 17:30 Main building (Georg Sverdrups house)


Three well-known and important European computing scientists died within a very short time during the summer of 2002: Ole-Johan Dahl, Edsger W. Dijkstra, and Kristen Nygaard. These three men were among the brightest stars of early informatics. All three were honoured with the ACM A. M. Turing Award, by many considered the Nobel Prize for Computing, for their work.

Dijkstra received the Turing Award as early as 1972 for being "a principal contributor in the late 1950's to the development of the ALGOL, a high level programming language which has become a model of clarity and mathematical rigor". Dahl and Nygaard received the Award in 2001 "for ideas fundamental to the emergence of object oriented programming, through their design of the programming languages Simula I and Simula 67". Dahl and Nygaard were co-recipients of the 2002 IEEE John von Neumann Medal "for the introduction of the concepts underlying object-oriented programming through the design and implementation of SIMULA 67."

Edsger W. Dijkstra
11 May 1930–6 Aug 2002

Kristen Nygaard
27 Aug 1926–10 Aug 2002

Ole-Johan Dahl
12 Oct 1931–29 Jun 2002

To honour Dahl, Dijkstra, and Nygaard, the University of Klagenfurt, Austria, established a memorial exhibition in connection with two conferences (Euro-Par and JMLC) held in Klagenfurt at the end of August 2003. The exhibition, which was most fittingly named "People behind Informatics", showed the evolution of the field of information and communication technology during the lifetime of Dahl, Dijkstra, and Nygaard, as well as their main accomplishments and place in the history.

The University of Klagenfurt has generously donated the exhibition to the Institute of Informatics at the University of Oslo. The exhibition will be shown at the conference site during ECOOP 2004

A virtual exhibition, accessible to anyone, can be found on the web at http://cs-exhibitions.uni-klu.ac.at/.
Social Program

Reception at the City Hall of Oslo
Wednesday, 16 June 18:00
The Oslo City Hall ("Rådhuset") is situated in the city center within walking distance of the Royal Palace and the Parliament building.

Located on the waterfront, overlooking the bay of the Oslo fjord, the City Hall reflects the historic role of Oslo as the capital of a seafaring nation.
Inaugurated in 1950, it is decorated by the foremost Norwegian artists from the period 1900-1950 with motifs from Norwegian history, culture and working life.
The City Hall is within walking distance from all city center hotels.

Fjord cruise on the Oslo fjord
Wednesday, 16 June 19:30 – 21:15
After the reception at the City Hall you are invited to a fjord cruise on the Oslo Fjord. The boat will leave at the harbor just beside the City Hall.
The cruise will include snacks and drinks, but not a full meal. Restaurants will be open for dinner after the cruise.

Banquet at Gamle Logen
Thursday, 17 June 20:00

The banquet will be at Gamle Logen, one of Oslo’s oldest cultural building. It was originally built as a Masonic Lodge and official banquet hall for the City of Christiania - Oslo’s previous name - in 1846. The Norwegian Parliament celebrated its 50 years Union with Sweden, and King Haakon VII his 25th anniversary as King of Norway in these premises. Gamle Logen’s main hall, decorated in late-empire style, and world famous for the quality of its acoustics, was officially opened in 1844.
Gamle Logen is within walking distance from all city center hotels. From other hotels, get to one of the city center stations and walk from there.
Address: Grev Wedelsplass 2
Organizers of ECOOP 2004

Conference Chair
Birger Møller-Pedersen, University of Oslo, Norway.

Organizing Chair
Arne Maus, University of Oslo, Norway.

Program Chair
Martin Odersky, EPFL - Ecole Polytechnique Federale de Lausanne, Switzerland.

Workshop Chairs
Jacques Malenfant, Université Pierre et Marie Curie, France.
Bjarte M. Østvold, Norwegian Computing Center, Oslo, Norway.

PhD workshop/doctoral symposium Chairs
Susanne Jucknath, Technische Universität Berlin, Germany.
Eric Jul, University of Copenhagen, Denmark.

Tutorial Chairs
Arne-Jørgen Berre, Sintef, Norway.
Hanspeter Mössenböck, Johannes Kepler Universität, Linz, Germany.

Poster, Exhibition and Demo Chair
Ole Smørødal, University of Oslo, Norway.

Practitioners’ Reports Chair
Egil P Andersen, DNV Software, Norway.

Room Logistics
Stein Krogdahl, University of Oslo, Norway.

Registration and Web
Dag Langmyhr, University of Oslo, Norway.

Program Committee
Uwe Assmann, Linköpings Universitet, Sweden.
Don Batory, University of Texas at Austin, USA.
Gilad Bracha, Sun Microsystems, USA.
Luca Cardelli, Microsoft Research Cambridge, UK.
Charles Consel, LaBRI/INRIA, France.
Giuseppe Castagna, Ecole Normale Superieure, France.
Peter Dickman, University of Glasgow, UK.
Sophia Drossopoulou, Imperial College, UK.
Erik Ernst, University of Aarhus, Denmark.
Manuel Faehndrich, Microsoft Research, USA.
Giovanna Guerrini, University of Pisa, Italy.
Urs Hoelzle, Google, USA.
Mehdi Jazayeri, Technical University of Vienna, Austria.
Shriram Krishnamurthi, Brown University, USA.
Doug Lea, SUNY Oswego, USA.
Mira Mezini, Darmstadt University of Technology, Germany.
Oscar Nierstrasz, University of Bern, Switzerland.
Martin Odersky, EPFL, Switzerland.
Atushi Ohori, JAIST, Japan.
Douglas Schmidt, Vanderbilt University, USA.
Luis Rodrigues, Universidade de Lisboa, Portugal.
Clemens Szyperski, Microsoft Research, USA.
Philip Wadler, University of Edinburgh, UK.