JAIST

FINAL PROGRAM

JSSST International Symposium Series

2nd International Symposium on Object Technologies for Advanced Software

ISOTAS '96

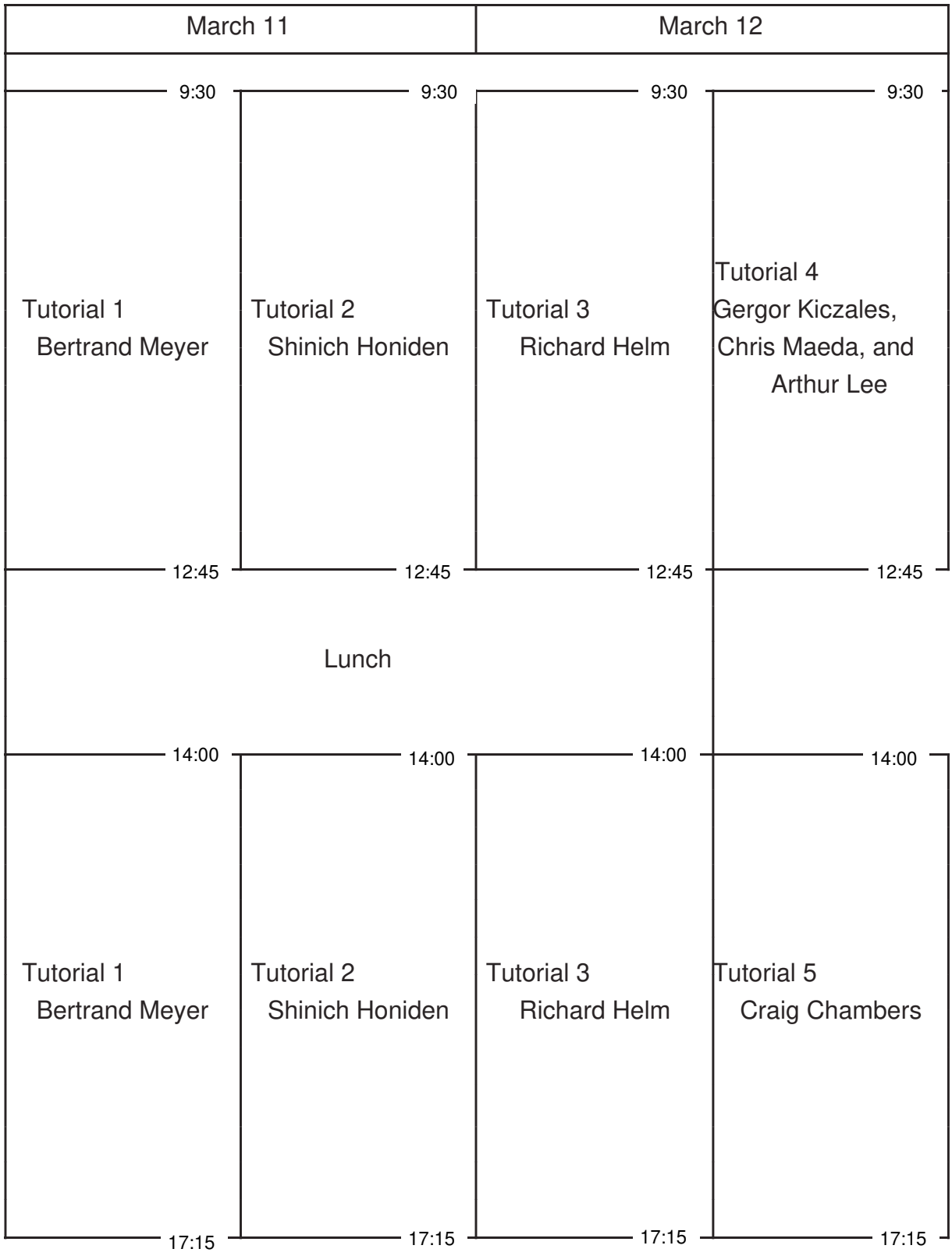
11-15 March, 1996, Kanazawa, Japan,

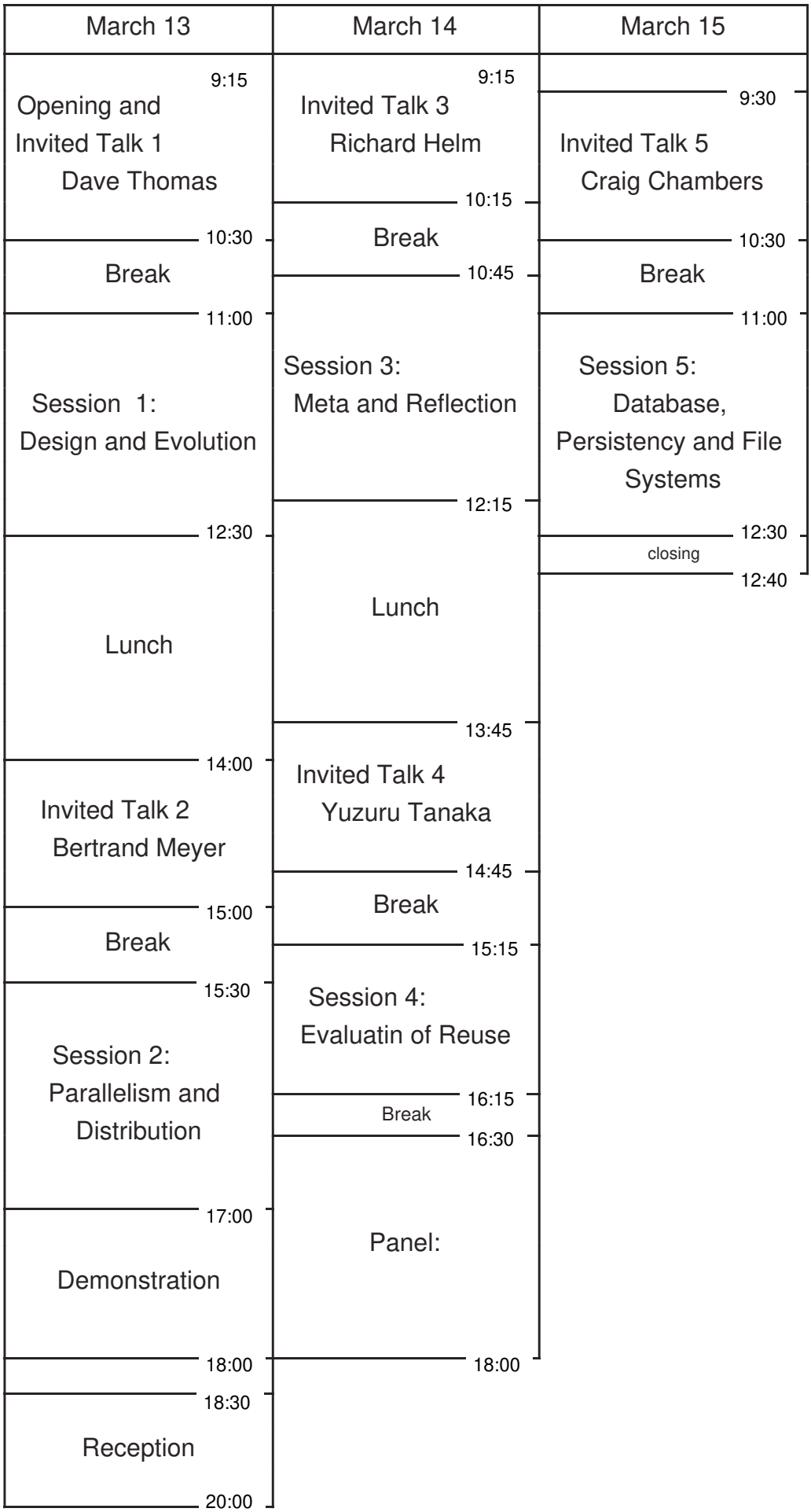
Sponsored by
Japan Advanced Institute of Science and Technology (JAIST),
Japan Society for Software Science and Technology (JSSST)

In cooperation with
ACM SIGPLAN/SIGSOFT, IEEE Computer Society-TCSE, IEICE, IPSJ

JSSST JAIST 96

ISOTAS'96 at a GLANCE




March 13
March 14
March 15
9:15
Opening and
Invited Talk 1
Dave Thomas
10:30
Break
11:00
Session 1:
Design and Evolution
12:30
Lunch
14:00
Invited Talk 2
Bertrand Meyer
15:00
Break
15:30
Session 2:
Parallelism and
Distribution
17:00
Demonstration
18:00
18:30
Reception
20:00
9:15
Invited Talk 3
Richard Helm
10:15
Break
10:45
Session 3:
Meta and Reflection
12:15
Lunch
13:45
Invited Talk 4
Yuzuru Tanaka
14:45
Break
15:15
Session 4:
Evaluatin of Reuse
16:15
Break
16:30
Panel:
18:00
9:30
Invited Talk 5
Craig Chambers
10:30
Break
11:00
Session 5:
Database,
Persistency and File
Systems
12:30
closing
12:40

SYMPOSIUM COMMITTEE

Symposium Chair:	Takuya KATAYAMA (JAIST)
Symposium Series Committee Chair:	Mario TOKORO (Keio U./Sony CSL)
Program Committee Co-Chairs:	Kokichi FUTATSUGI (JAIST), Satoshi MATSUOKA (U. of Tokyo)
Workshop Chair:	Koichiro OCHIMIZU (JAIST)
Tutorial Chair:	Jiro TANAKA (U. of Tsukuba)
Finance Chair:	Naoki YONEZAKI (Tokyo Inst. of Tech.)
Executive Chair:	Yoichi SHINODA (JAIST)
Local Arrangement Chair:	Tatsuo NAKAJIMA (JAIST)
Publicity Chair:	Takuo WATANABE (JAIST)
Exhibition Chair:	Masato SUZUKI (JAIST)

PROGRAM COMMITTEE MEMBERS

Tsuneo AJISAKA	Kyoto U.	Shojiro NISHIO	Osaka U., ISOTAS'93 PC Co-Chair
Mehmet AKSIT	Twente U.	David NOTKIN	Washington U.
Craig CHAMBERS	U. of Washington	Atsushi OHORI	Kyoto U.
Andrew CHIEN	U. of Illinois	Rikio ONAI	NTT Lab.
Pierre COINTE	U. of Nantes	Tamiya ONODERA	IBM
Steven FEINER	Columbia U.	Beng Chin OOI	N. U. of Singapore
Bjorn FREEMAN-BENSON	Carleton U.	Lee OSTERWEIL	U. of Mass.
Dennis GANNON	Indiana U.	Jens PALSBERG	Aarhus U.
Shinichi HONIDEN	Toshiba	Motoshi SAEKI	Tokyo Inst. of Tech.
Hiroshi ISHIKAWA	Fujitsu Lab.	Etsuya SHIBAYAMA	Tokyo Inst. of Tech.
Yutaka ISHIKAWA	RWC Lab.	Yoichi SHINODA	JAIST
Ralph JOHNSON	U. of Illinois	Jiro TANAKA	U. of Tsukuba
Kazuhiko KATO	U. of Tsukuba	Dave THOMAS	Object Tech. Intl.
Gregor KICZALES	Xerox PARC	Takuo WATANABE	JAIST
Roger KING	U. of Colorado	Grant E. WEDDELL	U. of Waterloo
Karl LIEBERHERR	Northeastern U.	Kazumasa YOKOTA	Kyoto U.
Jose MESEGUER	SRI	Yasuhiko YOKOTE	Sony CSL
Eliot MOSS	U. of Mass.	Akinori YONEZAWA	U. of Tokyo, ISOTAS'93 PC Co-Chair
Shin NAKAJIMA	NEC	Masatoshi YOSHIKAWA	NAIST
Tatsuo NAKAJIMA	JAIST	Stanley B. ZDONIK	Brown U.
K. NARAYANASWAMY	USC-ISI	Roberto ZICARI	J.W. Goethe U.
Oscar NIERSTRASZ	U. of Bern		

AIMS and GOALS

Object technologies are attracting ever increasing attentions in various areas of research and development for advanced software:

・Object-Oriented Programming (OOP) reduces the complexity of large-scale software development through inheritance/encapsulation/class libraries/framework. OOP also serves as a basis of new generations of programming languages in concurrent and reflective computing, etc. There is also strong affinity to so-called autonomous `agent-oriented programming'.

・Object-Oriented Databases (OODB) have been proposed as a model for next generation database systems, which are expected to overcome the limitations of conventional data models. Currently OODB are used in a wide variety of important application areas such as engineering and multimedia information systems, where the fusion of OOP and OODB shows a great promise in providing much more effective architecture for advanced database systems.

・Object-Oriented Software Engineering (OOSE) promises to present solutions to numerous difficulties the software engineers face today. Examples include: OO requirement analysis, OO formal specifications, OO design, OO software databases, etc.

The diversity of different areas suggest that there are underlying, fundamental principles common to a wide range of software development activities. Following up the success of the first ISOTAS, the second ISOTAS focuses again on the object technologies to shed light on such principles. By bringing together the leading researchers in the areas of OOP, OODB, and OOSE, we hope to deepen the understanding of object technologies and establish a solid framework for future advanced software development.

INVITED TALKS

Invited Speakers

March 13, 1996

Dave Thomas (OTI)
Object Utopia

Bertrand Meyer (ISE Inc.)
Beyond Design by Contract: Towards Formally Provable Object-Oriented Software

March 14, 1996

Richard Helm (IBM Consulting Group/ISSC Australia)
Patterns in Practice

Yuzuru Tanaka (Hokkaido Univ.)
A Meme Media Architecture for Fine-Grain Computer Software

March 15, 1996

Craig Chambers (Univ. of Washington)
Synergies between Object-Oriented Language Design and Implementation Research

TECHNICAL PROGRAM

Wednesday, March 13, 1996

Opening and Invited Talk 1 (9:15 - 10:30)
Opening
Opening Welcome Speech
Takuya Katayama (JAIST, Organizing Chair)

Report from the Program Committee
Kokichi Futatsugi (JAIST, Program Committee Co-Chair)
Satoshi Matsuoka (Univ. of Tokyo, Program Committee Co-Chair)

Invited Talk 1
Session Chair:
Mario Tokoro (Keio University/Sony CSL)
Object Utopia
Dave Thomas (OTI)

Break (10:30 - 11:00)

Session 1: Design and Evolution (11:00 - 12:30)
Session Chair:
Naoki Yonezaki (Tokyo Institute of Technology)

Automating the Evolution of Object-Oriented Systems
Walter L. Hursch & Linda M. Seiter (Northeastern Univ.)

Using C++ Templates to Implement Role-Based Designs
Michael Van Hilst & David Notkin (Univ. of Washington)

Explicit Metaclasses as a Tool for Improving the Design of Class Libraries
T. Ledoux & P. Cointe (Ecole des Mines de Nantes)

Break(Lunch) (12:30 - 14:00)

Invited Talk 2 (14:00 - 15:00)
Session Chair:
Kokichi Futatsugi (JAIST)

Beyond Design by Contract: Towards Formally Provable Object-Oriented Software
Bertrand Meyer (ISE Inc.)

Break (15:00 - 15:30)

Session 2: Parallelism and Distribution (15:30 - 17:00)
Session Chair:
Yoichi Shinoda (JAIST)

ICC++ - A C++ Dialect for High Performance Parallel Computing
Andrew Chien, U. Reddy, J. Plevyak & J. Dolby (Univ. of Illinois)

Hierarchical Collections: an Efficient Scheme to Build an Object-Oriented Distributed Class Library for Massively Parallel Computation
Naohito Sato, Satoshi Matsuoka & Akinori Yonezawa
(Univ. of Tokyo)
Adaptive Parameter Passing
Cristina Lopes (Northeastern Univ.)

Demonstration (17:00 - 18:00)
Session Chair:
Masato Suzuki/Tatsuo Nakajima (JAIST)

An Overview of RWC Massively Parallel Software
Yutaka Ishikawa, Atsushi Hori, Hiroshi Tezuka, Hiroki Konaka, Munenori Maeda, Jorg Nolte, Nobuyuki Ichiyoshi, Toshihiro Nishioka, Motohiko Matsuda & Kazuto Kubota
(Real World Computing Partnership)

Schematic: A Concurrent Object-Oriented Extension to Scheme
Kenjiro Taura, Toshio Endo, Yoshihiro Oyama, & Ryoichi Kakimoto (University of Tokyo)

Mobile and Multimedia support on Real-Time Mach
Akihiro Hokimoto, Jun Noritake, & Tatsuo Nakajima
(JAIST)

CafeOBJ an O-O algebraic specification language
Toshimi Sawada & Kokichi Futatsugi (JAIST)

Reception (18:30 - 20:00)

Thursday, March 14, 1996

Invited Talk 3 (9:15 - 10:15)
Session Chair:
Akinori Yonezawa (University of Tokyo)

Patterns in Practice
Richard Helm (IBM Consulting Group/ISSC Australia)

Break (10:15 - 10:45)

Session 3: Meta and Reflection (10:45 - 12:15)
Session Chair:
Takuo Watanabe (JAIST)

Designing a Meta Object Protocol to Wrap a Standard Graphical Toolkit
Erik Gallesio (Univ. de Nice - Sophia Antipolis)

Avoiding Confusion in Metacircularity: The Meta-Helix
Shigeru Chiba, Gregor Kiczales & John Lamping (Xerox PARC)

An Algebraic Semantics of Reflective Objects
Yasuyuki Tahara, Fumihiro Kumeno, Akihiko Ohsuga & Shinichi Honiden (Information-Technology Promotion Agency)

Break(Lunch) (12:15 - 13:45)

Invited Talk 4 (13:45 - 14:45)
Session Chair:
Jiro Tanaka (University of Tsukuba)

A Meme Media Architecture for Fine-Grain Computer Software
Yuzuru Tanaka (Hokkaido Univ.)

Break (14:45 - 15:15)

Session 4: Evaluation of Reuse (15:15 - 16:15)
Session Chair:
Koichiro Ochimizu (JAIST)

A Metric for Evaluating Effectiveness of Object-Oriented Interface for Promoting Software Reuse
Koki Nakanishi & Takashi Arano (NTT Software Labs.)

Experience in Classification and Reuse of Synchronization Schemes (cancelled)
Jean-Pierre Briot (Univ. of Tokyo)

Break (16:15 - 16:30)

Panel (16:30 - 18:00)
Towards object paradise: How do we get there?
Panel Moderator:
Yutaka Ishikawa (Real-World Computing Partnership)
Panelists:
Dave Thomas
Richard Helm
Tsutomu Kamimura
(other panelists to be announced)

Friday, March 15, 1996

Invited Talk 5 (9:30 - 10:30)
Session Chair:
Satoshi Matsuoka (University of Tokyo)

Synergies between Object-Oriented Language Design and Implementation Research
Craig Chambers (Univ. of Washington)

Break (10:30 - 11:00)

Session 5: Database, Persistency and File Systems (11:00 - 12:30)
Session Chair:
Kazuhiko Kato (Univeristy of Tsukuba)

Ocore and VAST - Integrating Persistence into a Parallel Object-Oriented Language
Jorg Nolte, Hiroki Konaka, Atsushi Hori, Yutaka Ishikawa, Munenori Maeda & Takashi Tomokiyo (RWC)

A Metaobject Protocol for Accessing File Systems
Chris Maeda (Univ. of Washington)

Subtyping by Constraints in Object-Oriented Databases
Wolfgang Kowarschick, Gerhard Kostler & Werner Kiessling (Technische Universitaet Muenchen)

Closing (12:30 - 12:40)
Kokichi Futatsugi (JAIST, Program Committee Co-Chair)
Satoshi Matsuoka (Univ. of Tokyo, Program Committee Co-Chair)

TUTORIALS

Monday, March 11, 1996

T1: *An In-Depth Look at Reusability*
(all day: 9:30 - 17:15, 2 units)

Bertrand Meyer (ISE Inc.)

T2: *Object-Oriented Software Engineering*
(all day: 9:30 - 17:15, 2 units)

Shinichi Honiden (Toshiba)

Tuesday, March 12, 1996

T3: *Using Design Patterns: Elements of Reusable Architectures*
(all day: 9:30 - 17:15, 2 units)

Richard Helm (IBM Consulting Group/ISSC)

T4: *Open Implementation Analysis and Design [TM]*
(How to Make Black Boxes Easier to Reuse)
(morning: 9:30 - 12:45, 1 unit)

Gregor Kiczales, Chris Maeda (Xerox PARC), and
Arthur Lee (Korea Univ.)

T5: *Efficient Implementation of Object-Oriented Programming Languages*
(afternoon: 14:00 - 17:15, 1 unit)

Craig Chambers (U. of Washington)

ABSTRACTS of TUTORIALS

T1: *An In-Depth Look at Reusability*

Bertrand Meyer (ISE Inc.)

Language: **English**
Level: **Intermediate**

Drawing on the experience of thousands of widely reused classes, this presentation will explain the issues, both managerial and technical, that must be addressed for a successful reuse policy. It will review what managers must do to promote reuse in their organization, and describe the technical tools that are necessary to produce large numbers of high-quality reusable components. The technical part of the presentation will be based on the Eiffel approach to reusability and reliability.

Among the topics surveyed:

- Real and imagined obstacles to reuse, and how to overcome them.
- Degrees of reusability.
- Producers and consumers. Where to concentrate efforts.
- Technical issues in reuse. What is the contribution of object technology?
- High-level design principles to ensure software reusability: active data structures, command-query separation principle, uniform access, options vs. parameters, etc.
- Using inheritance properly.
- A Linnaean approach to the cataloging of fundamental reusable structure.
- Design by Contract and its influence on library design and use.
- How to make a library easy to use.
- Style rules: choice of names, argument passing, class organization, feature clauses, comments.
- Metrics of reuse and quality. Measurements and statistics from existing libraries.
- Documenting reusable components.
- Library organization.
- The role of consistency.
- Lessons from successful reusable libraries.
- Specific needs of application libraries. The role of the application domain.
- The future of reusability.

Attendees will be presented with a number of examples of good (and bad) reusable library design.

Bertrand Meyer is president of ISE Inc. (Santa Barbara), editor of two book series (Prentice Hall's Object-Oriented Series and Addison-Wesley's ``Eiffel in Practice''), chairman of the TOOLS conferences (Technology of Object-Oriented Languages and Systems) and associate member of the applications section of the French Academy of Sciences. He is the author of many books including two available in Japanese (``Object-Oriented Software Construction'', ``Introduction to the Theory of Programming Languages''), ``Object Success'' (a presentation of object technology for managers), ``Eiffel: The Language'', and ``Reusable Software'' (on the tutorial's topics). He has directed successful projects of more than half a million lines of reusable object-oriented reusable software.

T2: *Object-Oriented Software Engineering*

Shin'ichi Honiden (Toshiba)

Language: **Japanese**
Level: **Beginning/Intermediate**

The tutorial has two major parts. Part one focuses on object-oriented analysis and design method – the basic concept of methodology, the several methodology examples, an overview of state-of-the-art methodologies, and the applications to the practical system. It also discusses some success and failure stories that shed light on the lessons to be learned.

Part two focuses on reusable software components, such as design pattern, framework, and componentware. It gives an overview of approaches and how to apply reusable software components in application domains.

Shinichi Honiden received the B.E., M.E., and Dr. Eng. degrees in electrical engineering from Waseda University, Tokyo, Japan, in 1976, 1978, and 1986, respectively. Since 1978, he has been with Toshiba Corporation, Tokyo. His research interests include software engineering and artificial intelligence, with particular reference to object-oriented analysis and design, agent-oriented model, formal specification languages and methodologies, and analysis and validation techniques for real-time and concurrent systems. He has published more than 70 papers in journals and international conferences of the fields, and 6 books on object-oriented software engineering.

T3: *Using Design Patterns: Elements of Reusable Architectures*

Richard Helm (IBM Consulting Group/ISSC Australia)

Language: **English**
Level: **Advanced**

Designing object-oriented software is hard, and designing reusable object-oriented software is even harder. Experience shows that any object-oriented systems exhibit recurring structures or "design patterns" of communicating and collaborating objects that promote extensibility, flexibility, and reusability. This course describes a set of fundamental design patterns and, through a design scenario, demonstrates how to build reusable object-oriented software based on them. Participants will learn a valuable set of design patterns that they can apply to the design of their own object-oriented systems, thereby making them more effective designers. The course covers the roles design patterns play in the object-oriented development process: how they provide a common vocabulary, reduce system complexity, and how they act as reusable architectural elements that contribute to an overall system architecture. This tutorial is intended for architects, system designers, and programmers who design object-oriented software. Attendees should have experience in object-oriented design and should understand object-oriented concepts such as polymorphism and type versus interface inheritance.

Richard Helm is a consultant with the object technology practice with IBM Consulting Group/ ISSC Australia in Sydney Australia. There he is actively applying patterns to the design of commercial systems. Prior to IBM, Richard was with DMR Group based in Montreal, Quebec, and prior to that he was a research staff member with IBM at the T.J. Watson Research Center in New York. Richard has numerous international publications, is a frequent speaker at international conferences, and is one of the four co-authors of the award-winning book Design Patterns: Elements of Reusable Object-Oriented Software. Richard has a Ph.D. from the University of Melbourne Australia.

T4: *Open Implementation Analysis and Design [TM] (How to Make Black Boxes Easier to Reuse)*

Gregor Kiczales, Chris Maeda (Xerox PARC), and
Arthur Lee (Korea University)

Language: **English**
Level: **Intermediate**

Open Implementation Analysis and Design is a new methodology for designing Open Implementations of substrate software. The designer of any performance-critical reusable software faces a difficult challenge: there are many implementation decisions that will invariably bias the system's performance towards one kind of client use and away from others. For example: a file system implementation can be designed to penalize or favor the use of many small files; a virtual memory system can be designed to penalize or favor its use in implementing a database; a set abstraction can penalize or favor frequent delete operations etc. We call such decisions implementation strategy dilemmas, to reflect the fact that the designer seems forced to choose between making some clients happy vs. making other clients happy. Open Implementation is an architectural solution to this problem that works by allowing clients of a module principled control over the module's implementation strategy decisions. So, for example, one client of a file system could choose the block size for their files and in doing so ensure that the file system was biased towards their needs. Open Implementation Analysis and Design is a methodology that allows designers to decide what aspects of a module's implementation a client should control, and how best to provide that control. This methodology is synergistic with OOA/D methodologies, but places more of an emphasis on how to make software that is tailorable by clients.

Gregor Kiczales is the leader of the Open Implementation project at Xerox PARC. He is the inventor of the concept of Open Implementation, and one of the developers of OIA/D. He has done extensive work in object-oriented programming languages and techniques, and in the area of metaobject protocols.

Chris Maeda is a member of the Open Implementation project at PARC. He is one of the developers of OIA/D.

Arthur Lee is a professor at Korea University, where he heads a new project on Open Implementation. He is one of the developers of OIA/D.

T5: *Efficient Implementation of Object-Oriented Programming Languages*

Craig Chambers (U. of Washington)

Language: **English**
Level: **Advanced**

How are object-oriented languages implemented? What features of object-oriented languages are expensive? What optimizations have been developed to make object-oriented languages more efficient? How important is compiler optimization for supporting high-level object-oriented languages? What are important considerations when assessing the effectiveness of compiler optimization? This tutorial aims to provide some answers to these questions. The tutorial will begin by presenting the important language design issues, identifying the features of object-oriented languages that are difficult to implement efficiently. In the main part of the tutorial, three classes of implementation techniques will be presented. First, run-time system techniques such as virtual function dispatch tables and inline caches will be described. Second, the tutorial will describe several levels of static analyses which seek to identify at compile-time the possible classes of message receivers in order to reduce or eliminate the overhead of dynamic binding. Third, the tutorial will discuss ways in which dynamic execution profiles can be exploited to complement static analysis techniques. The tutorial will present empirical measurements of the effectiveness of these various techniques for different kinds of programs and languages.

Craig Chambers has been researching object-oriented language design and implementation since 1987, with publications in OOPSLA, ECOOP, and PLDI on the topic. He developed the first efficient implementation of the Self language, and he is the designer of the Cecil language and heads the Vortex optimizing compiler project. Chambers is currently an Assistant Professor of Computer Science & Engineering at the University of Washington.

Research Demonstrations

Exhibition Hours
March 13: 12:00 - 19:30
March 14: 9:00 - 16:00
Location
Meeting Room C1,C2

1. An Overview of RWC Massively Parallel Software
Yutaka Ishikawa, Atsushi Hori, Hiroshi Tezuka, Hiroki Konaka, Munenori Maeda, Jorg Nolte, Nobuyuki Ichiyoshi, Toshihiro Nishioka, Motohiko Matsuda and Kazuto Kubota
(Real World Computing Partnership)

We have been designing and developing a parallel object-oriented software system which includes two programming languages, MPC++ and OCore, an operating system called SCore, a persistent object environment called VAST, and a programming environment. We will demonstrate i) OCore language feature and runtime environment by Java and a video tape, and ii) MPC++/SCorer runtime and programming environment on the RWCP workstation cluster using the internet. For more information, please see http://www.rwcp.or.jp/people/mpslab.

2. Schematic: A Concurrent Object-Oriented Extension to Scheme
Kenjiro Taura, Toshio Endo, Yoshihiro Oyama, and Ryoichi Kakimoto (University of Tokyo)

We present Schematic, a concurrent object-oriented language based on Scheme. The particular implementation we demonstrate runs on networks of workstations which talk TCP/IP, and features a Tk-like concurrent object-oriented GUI toolkit. Application programs include a simple concurrent GUI application (Mandelbrot), a classic two-players game (Table Tennis), and a distributed application (Interactive Questionnaire Server). We explain how languages like Schematic are useful for developing these applications. We emphasize that concurrent languages such as Schematic are not only for running intensive applications on parallel machines, but also for making development of such fun programs (GUI and distributed applications) easier!

3. Mobile and Multimedia support on Real-Time Mach
Akihiro Hokimoto, Jun Noritake, and Tatsuo Nakajima (JAIST)

Multimedia and mobile computing(MMMC) group in JAIST is working on building a new software system architecture that can be adapted when operating environment is changed. We are currently building continuous media systems such as a continuous media toolkit and a storage server based on resource reservation and dynamic QOS control. Also, we are working on a toolkit for mobile computing environment that enables the structures of applications to be changed when a PC Card is inserted or removed. If you like to get more information, please check
http://mmmc.jaist.ac.jp:8000".

4. CafeOBJ an O-O algebraic specification language
Toshimi Sawada and Kokichi Futatsugi (JAIST)

CafeOBJ is an algebraic specification language which has rewriting logic as its logical basis.
It has some object oriented facilities, such as classes, inheritance, concurrent OO computation, and so on. In our demonstration, some examples of axiomatic descriptions (specifications) of objects and their prototyping executions by CafeOBJ interpreter will be presented. The computation just corresponds to the deduction of rewriting logic, and is performed by rewriting modulo structural axioms of ACI (associative commutative and identity) that capture abstractly the essential aspects of communication in a distributed object-oriented configuration made up of concurrent objects and messages.

ObjectWeek at a GLANCE

	Morning	Afternoon
March 8		**WOOC (JSSST Workshop on Object-Oriented Computing)**
March 9	**WOOC**	
March 10	**WOOC**	**IWPC++ (International Workshop on Parallel C++)**
March 11	**WINGS (Workshop on Design and Integration of New Generation Software Systems)** **IWPC++** **Tutorial 1: Meyer** 9:30 17:15 **Tutorial 2: Honiden**	
March 12	**WINGS** **IWPC++** **Tutorial 3: Helm** 9:30 12:45 **Tutorial 4: Kiczales**	14:00 17:15 **Tutorial 5: Chambers**
March 13	**ISOTAS'96 Technical Sessions**	
March 14	**ISOTAS'96 Technical Sessions**	
March 15	**ISOTAS'96 Technical Sessions**	