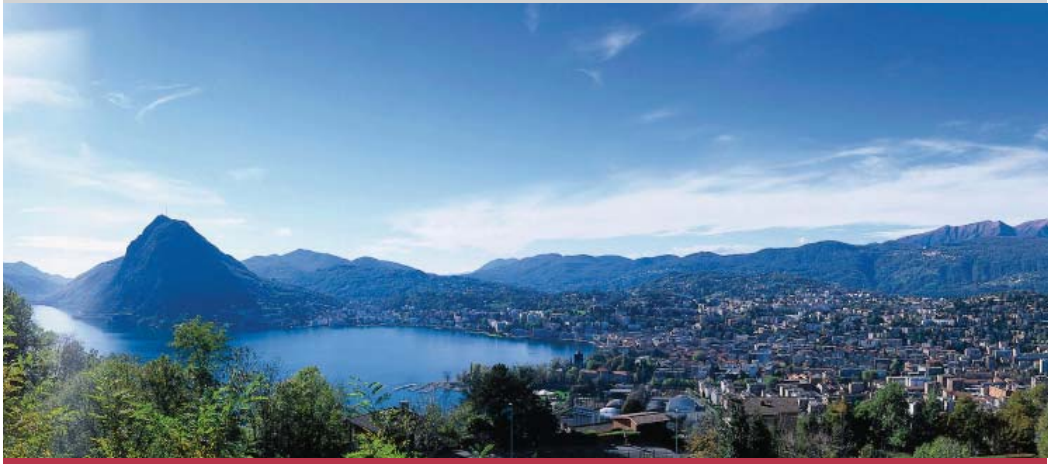
13th International Conference on

MODULARITY 14

13th International Conference on Modularity



Lugano, Switzerland
April 22-25, 2014

Program

Welcome to Modularity'14!

Modularity'14 addresses all aspects of modularity, abstraction, and separation of concerns as they pertain to software, including new forms, uses, and analysis of modularity, along with the costs and benefits, and tradeoffs involved in their application. The broadening in scope of the conference is also reflected in the change of its name: the International Conference on Aspect-Oriented Software Development (AOSD) has evolved to become the International Conference on Modularity.

Modularity'14 comprises two main parts: Research Results and Modularity Visions. Both parts invited full, scholarly papers of the highest quality on results and new ideas in areas that include but are not limited to complex systems, software design and engineering, programming languages, cyber-physical systems, and other areas across the whole system life cycle.

In addition, Modularity'14 hosts invited keynote talks, an ACM Student Research Competition (SRC), demonstrations, and the 13th Workshop on Foundations of Aspect-Oriented Languages (FOAL'14).

We hope that you will find this program inspiring and compelling, and that the conference will provide you with a valuable opportunity to share ideas with other researchers and practitioners from institutions around the world.

Walter Binder
General Chair
University of Lugano, Switzerland

Erik Ernst
Program Chair Research Results
Aarhus University, Denmark

Walter Cazzola
Demonstrations Chair
Università degli Studi di Milano, Italy

Michael Haupt
Workshops Chair
Oracle Labs, Germany

Achille Peternier
Organizing Chair
University of Lugano, Switzerland

Robert Hirschfeld
Program Chair Modularity Visions
Hasso-Plattner-Institut Potsdam, Germany

Christoph Bockisch
Student Events Chair
University of Twente, The Netherlands

Tuesday, April 22, 2014

14.00-15.00	**FOAL: Keynote /** Auditorium To be destructive or not to be, that is the question on modular extensions (Shigeru Chiba) - Chair: Gary T. Leavens		
15.00-15.30	**Coffee Break**		
15.30-17.30	**FOAL: Papers** **/** Auditorium Chair: Eric Bodden Event-based Modularization (Somayeh Malakuti, Mehmet Aksit) Static Verification of PtolemyRely Programs Using OpenJML (Jose Sanchez, Gary Leavens) Specification of Domain-Specific Languages Based on Concern Interfaces (Matthias Schöttle, Omar Alam, Gunter Mussbacher, Jörg Kienzle) Context Holders: Realizing Multiple Layer Activation Mechanisms in a Single Context-Oriented Language (Tomoyuki Aotani, Tetsuo Kamina, Hidehiko Masuhara) ECAlogic: Hardware-Parametric Energy-Consumption Analysis of Algorithms (Marc Schoolderman, Jascha Neutelings, Rody Kersten, Marko van Eekelen)	**Demos** **/** Room 351 Chair: Walter Cazzola TouchRAM: A Multi-touch-Enabled Software Design Tool Supporting Concern- Oriented Reuse (Matthias Schöttle, Omar Alam, Franz-Philippe Garcia, Gunter Mussbacher, Jörg Kienzle) Finding Bugs in Program Generators by Dynamic Analysis of Syntactic Language Constraints (Sebastian Erdweg, Vlad Vergu, Mira Mezini, Eelco Visser) Modularizing Crosscutting Contracts with AspectJML (Henrique Rebêlo, Gary T. Leavens, Mehdi Bagherzadeb, Hridesh Rajan, Ricardo Lima, Daniel M. Zimmerman, Márcio Cornélio, Thomas Thüm) iArch: An IDE for Supporting Fluid Abstraction (Di Ai, Naoyasu Ubayashi, Peiyuan Li, Daisuke Yamamoto, Yu Ning Li, Shintaro Hosoai, Yasutaka Kamei) Neverlang 2 - A Framework for Modular Language Implementation (Edoardo Vacchi, Diego Mathias Olivares, Albert Shaqiri, Walter Cazzola)	**Student Research Competition and Student Forum** **/** Room 321 Chair: Christoph Bockisch
18.00 onwards	**Reception with posters**		

Wednesday, April 23, 2014

8.50-9.00	**Opening /** Auditorium	
9.00-10.00	**Keynote /** Auditorium Separation of Concerns in Language Definition (Eelco Visser) - Chair: Hidehiko Masuhara	
10.00-10.30	**Coffee Break**	
10.30-12.30	**Session: Language Mechanisms I /** Auditorium - Chair: Julia Lawall Delegation Proxies: The Power of Propagation (Erwann Wernli, Oscar Nierstrasz, Camille Teruel, Stéphane Ducasse) Composable User-Defined Operators That Can Express User-Defined Literals (Kazuhiro Ichikawa, Shigeru Chiba) REScala: Bridging Between Object-oriented and Functional Style in Reactive Applications (Guido Salvaneschi, Gerold Hintz, Mira Mezini) FlowR: Aspect Oriented Programming for Information Flow Control in Ruby (Thomas F. J.-M. Pasquier, Jean Bacon, Brian Shand)	
12.30-14.30	**Lunch**	
14.30-16.30	**Session: Software Evolution** **/** Auditorium Chair: Christoph Bockisch Assessing Modularity using Co-Change Clusters (Luciana Lourdes Silva, Marco Tulio Valente, Marcelo de A. Maia) Blending and Reusing Rules for Architectural Degradation Prevention (Alessandro Gurgel, Isela Macia, Alessandro Garcia, Arndt von Staa, Mira Mezini, Michael Eichberg, Ralf Mitschke) Automated Software Remodularization Based on Move Refactoring - A Complex Systems Approach (Marcelo Serrano Zanetti, Claudio Juan Tessone, Ingo Scholtes, Frank Schweitzer) **Session: Modularity Visions** **/** Auditorium Context-Oriented Software Engineering: A Modularity Vision (Tetsuo Kamina, Tomoyuki Aotani, Hidehiko Masuhara, Tetsuo Tamai)	**Demos** **/** Room 351 - Chair: Walter Cazzola TouchRAM: A Multitouch-Enabled Software Design Tool Supporting Concern-Oriented Reuse (Matthias Schöttle, Omar Alam, Franz-Philippe Garcia, Gunter Mussbacher, Jörg Kienzle) Finding Bugs in Program Generators by Dynamic Analysis of Syntactic Language Constraints (Sebastian Erdweg, Vlad Vergu, Mira Mezini, Eelco Visser) Modularizing Crosscutting Contracts with AspectJML (Henrique Rebêlo, Gary T. Leavens, Mehdi Bagherzadeb, Hridesh Rajan, Ricardo Lima, Daniel M. Zimmerman, Márcio Cornélio, Thomas Thüm) iArch: An IDE for Supporting Fluid Abstraction (Di Ai, Naoyasu Ubayashi, Peiyuan Li, Daisuke Yamamoto, Yu Ning Li, Shintaro Hosoai, Yasutaka Kamei) Neverlang 2 - A Framework for Modular Language Implementation (Edoardo Vacchi, Diego Mathias Olivares, Albert Shaqiri, Walter Cazzola)

Thursday, April 24, 2014

8.50-9.00	Most Influential Paper Award **/** Auditorium Chair: David H. Lorenz
9.00-10.00	**Keynote /** Auditorium Graal and Truffle: Modularity and Separation of Concerns as Cornerstones for Building a Multipurpose Runtime (Thomas Würthinger) - Chair: Shigeru Chiba
10.00-10.30	**Coffee break**
10.30-12.00	**Session: Understanding Programmers /** Auditorium Chair: Guido Salvaneschi Type Names without Static Type Checking already Improve the Usability of APIs (As Long as the Type Names are Correct) - An Empirical Study (Samuel Spiza, Stefan Hanenberg) How Do Programmers Use Optional Typing? An Empirical Study (Carlos Souza, Eduardo Figueiredo) An Empirical Study on How Developers Reason about Module Cohesion (Bruno C. da Silva, Claudio N. Sant'Anna, Christina von F. G. Chavez)
12.00-13.30	**Lunch**
13.30-15.00	**Session: The Meaning of Programs /** Auditorium Chair: Eric Bodden Compositional Reasoning About Aspect Interference (Ismael Figueroa, Tom Schrijvers, Nicolas Tabareau, Éric Tanter) Reusable Components of Semantic Specifications (Martin Churchill, Peter D. Mosses, Paolo Torrini) AspectJML: Modular Specification and Runtime Checking for Crosscutting Contracts (Henrique Rebêlo, Gary T. Leavens, Mehdi Bagherzadeh, Hridesh Rajan, Ricardo Lima, Daniel M. Zimmerman, Márcio Cornélio, Thomas Thüm)
15.00 onwards	**Excursion + banquet**

Friday, April 25, 2014

8.50-9.00	Best Paper Award and Winner of the Student Research Competition **/** Auditorium Chairs: Erik Ernst and Christoph Bockisch
9.00-10.00	**Keynote /** Auditorium Coccinelle: Reducing the Barriers to Modularization in a Large C Code Base (Julia Lawall) - Chair: Naoyasu Ubayashi
10.00-10.30	**Coffee Break**
10.30-12.30	**Session: Software Product Lines /** Auditorium Chair: Stefan Hanenberg Probabilistic Model Checking for Energy Analysis in Software Product Lines (Clemens Dubslaff, Sascha Klüppelholz, Christel Baier) Systematic Derivation of Static Analyses for Software Product Lines (Jan Midtgaard, Claus Brabrand, Andrzej Wasowski) **Session: Concurrency /** Auditorium Chair: Gary T. Leavens Aspectual Session Types (Nicolas Tabareau, Mario Südholt, Éric Tanter) JEScala: Modular Coordination with Declarative Events and Joins (Jurgen M. Van Ham, Guido Salvaneschi, Mira Mezini, Jacques Noyé)
12.30-14.30	**Lunch**
14.30-16.00	**Session: Language Mechanisms II /** Auditorium Chair: Walter Cazzola Designing Information Hiding Modularity for Model Transformation Languages (Andreas Rentschler, Dominik Werle, Qais Noorshams, Lucia Happe, Ralf Reussner) JavaScript Module System: Exploring the Design Space (Junhee Cho, Sukyoung Ryu) Modular Specification and Dynamic Enforcement of Syntactic Language Constraints when Generating Code (Sebastian Erdweg, Vlad Vergu, Mira Mezini, Eelco Visser)
16.00-16.10	**Closing /** Auditorium

FOAL'14 Keynote

Shigeru Chiba
The University of Tokyo, Japan

To be destructive or not to be, that is the question on modular extensions

Tuesday, April 22
14.00-15.00

Abstract
Inheritance is a classic mechanism for extending an existing module. Since it preserves the original module, programmers can use both the original module and the extended one in the same program. So inheritance is a non-destructive mechanism. On the other hand, there are some extension mechanisms that directly modify an existing module and thus only the extended module is available in a program. These mechanisms such as aspects in AspectJ and revisers in our language GluonJ should be categorized into destructive mechanisms. Both destructive extension mechanisms and non-destructive ones are useful but in different scenarios.

This talk presents that the primary difference between destructive mechanisms and non-destructive ones is the scope of where the extensions are effective and visible in a program. Then this talk shows the third approach in the middle between the two extreme ones, destructive and non-destructive. The third approach allows programmers to control the scope of the extensions in a modular fashion. The talk presents a few language mechanisms of this approach, including our method shells, and also remaining issues in the contexts of feature-oriented programming.

About the Speaker
Shigeru Chiba is Professor at The University of Tokyo. Before starting his current position, he was Assistant, Associate, and later Full Professor at Tokyo Institute of Technology from 2001 to 2012. Before 2001, he was Assistant Professor at University of Tsukuba. He received his PhD in computer science in 1996 from the University of Tokyo. His research interests are in programming language design, in particular, of object-oriented and/or aspect-oriented programming languages. He is also interested in various kinds of system software including operating systems, distributed systems, and web application frameworks. He has been serving as a program committee member or organizer of a number of prestigious conferences and workshops. He is also a primary developer of Javassist, which is a Java bytecode engineering toolkit widely used in industry and academia.

Modularity'14 Keynotes

Eelco Visser
TU Delft, The Netherlands

Separation of Concerns in Language Definition

Wednesday, April 23
9.00-10.00

Abstract
Effectively applying linguistic abstraction to emerging domains of computation requires the ability to rapidly develop software languages. However, a software language is a complex software system in its own right and can take significant effort to design and implement. We are currently investigating a radical separation of concerns in language definition by designing high-level declarative meta-languages specialized to the various concerns of language definition that can be used as the single source of production quality (incremental) semantic operations and as a model for reasoning about language properties. In this talk I report about our progress in this direction by demonstrating language definition using the Spoofax Language Workbench.

About the Speaker
Eelco Visser is Antoni van Leeuwenhoek Professor of Computer Science at Delft University of Technology. He received a master's and doctorate in computer science from the University of Amsterdam in 1993 and 1997, respectively. Previously, he served as postdoc at the Oregon Graduate Institute, as Assistant Professor at Utrecht University, and as Associate Professor at TU Delft. In 2013 he received the prestigious NWO VICI grant for research into verification of language definitions. His research interests include software language engineering, domain-specific programming languages, model-driven engineering, program transformation, software deployment, and interaction design. With his students he has designed and implemented the Spoofax language workbench, as well as many domain-specific languages, including DSLs for syntax definition (SDF), program transformation (Stratego), software deployment (Nix), web application development (WebDSL), and mobile phone applications (mobl). He is the lead developer of the researchr bibliography management system and the WebLab learning management system.

Thomas Würthinger
Oracle Labs, Austria

Graal and Truffle: Modularity and Separation of Concerns as Cornerstones for Building a Multipurpose Runtime

Thursday, April 24
9.00 - 10.00

Abstract
Multi-language runtimes providing simultaneously high performance for several programming languages still remain an illusion. Industrial-strength managed language runtimes are built with a focus on one language (e.g., Java or C#). Other languages may compile to the bytecode formats of those managed language runtimes. However, the performance characteristics of the bytecode generation approach are often lagging behind compared to language runtimes specialized for a specific language. The performance of JavaScript is for example still orders of magnitude better on specialized runtimes (e.g., V8 or SpiderMonkey).

We present a solution to this problem by providing guest languages with a new way of interfacing with the host runtime. The semantics of the guest language is communicated to the host runtime not via generating bytecodes, but via an interpreter written in the host language. This gives guest languages a simple way to express the semantics of their operations including language-specific mechanisms for collecting profiling feedback. The efficient machine code is derived from the interpreter via automatic partial evaluation. The main components reused from the underlying runtime are the compiler and the garbage collector. They are both agnostic to the executed guest languages.

The host compiler derives the optimized machine code for hot parts of the guest language application via partial evaluation of the guest language interpreter. The interpreter definition can guide the host compiler to generate deoptimization points, i.e., exits from the compiled code. This allows guest language operations to use speculations: An operation could for example speculate that the type of an incoming parameter is constant. Furthermore, the guest language interpreter can use global assumptions about the system state that are registered with the compiled code. Finally, part of the interpreter's code can be excluded from the partial evaluation and remain shared across the system. This is useful for avoiding code explosion and appropriate for infrequently executed paths of an operation. These basic mechanisms are provided by the underlying language-agnostic host runtime and allow separation of concerns between guest and host runtime.

Guest language objects are stored on the host heap and managed by the host garbage collector. Multiple executing languages can share the same heap. This enables language interoperability without marshaling at the language boundary, because the field value of one language can point to an object of another language. Reusing the allocation system of the host runtime is another major simplification for guest language runtimes.

We implemented Truffle, the guest language runtime framework, on top of the Graal compiler and the HotSpot virtual machine. So far, there are prototypes for C, J, Python, JavaScript, R, Ruby, and Smalltalk running on top of the Truffle framework. The prototypes are still incomplete with respect to language semantics. However, most of them can run non-trivial benchmarks to demonstrate the core promise of the Truffle system: Multiple languages within one runtime system at competitive performance.

About the Speaker
Thomas Würthinger is a Senior Research Manager at Oracle Labs. He is the lead of the Graal compiler OpenJDK project and the architect of the Truffle self-optimizing runtime system. Previously, he worked on the Crankshaft optimizing compiler of V8 at Google, and the Maxine research virtual machine at Sun Microsystems. He received his PhD degree from Johannes Kepler University Linz in Austria for his thesis about dynamic code evolution for Java.

Modularity'14 Keynotes

Julia Lawall
Inria, France

Coccinelle: Reducing the Barriers to Modularization in a Large C Code Base

Friday, April 25
9.00 - 10.00

Abstract
Coccinelle is a program matching and transformation tool for C code that has been extensively used for bug finding and evolutions in the Linux kernel. In this talk, we show how Coccinelle can be used in maintaining and improving the modularity of software, taking as a case study the introduction of an API providing a form of managed resources into the Linux kernel.

About the Speaker
Julia Lawall is a Senior Researcher at Inria, and was previously an Associate Professor at the University of Copenhagen. Her research is at the intersection of programming languages, software engineering and operating systems. She has had over 1100 patches accepted into the Linux kernel based on tools developed in her research.

Sponsor & Organizing Committee

Sponsors

Università della Svizzera italiana — Facoltà di scienze informatiche

ORACLE LABS

Città di Lugano

Organizing Committee

General chair
Walter Binder
University of Lugano (USI), Switzerland

Organizing chair
Achille Peternier
University of Lugano (USI), Switzerland

Research Results chair
Erik Ernst
Aarhus University, Denmark

Modularity Visions chair
Robert Hirschfeld
Hasso-Plattner-Institut Potsdam, Germany

Workshops chair
Michael Haupt
Oracle Labs, Germany

Demonstrations chair
Walter Cazzola
Università degli Studi di Milano (Unimi), Italy

Student Events chair
Christoph Bockisch
University of Twente, The Netherlands

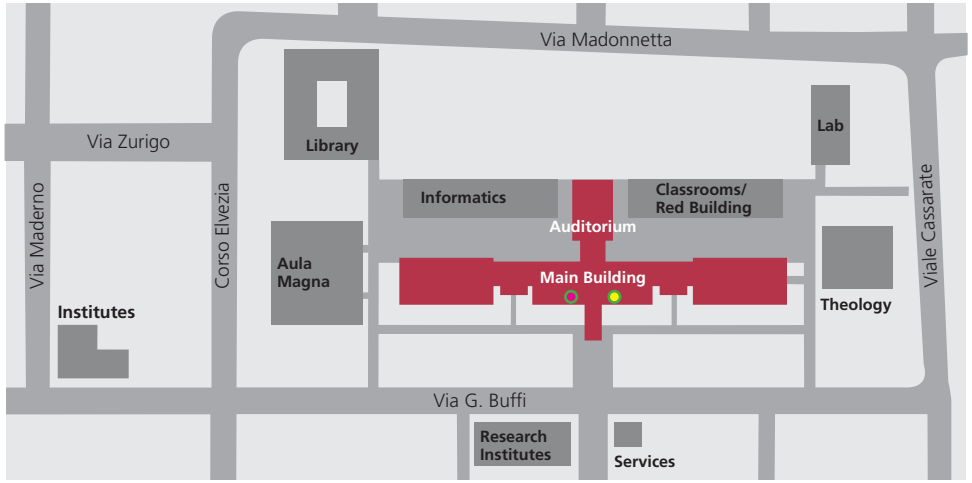
Publicity chair
Danilo Ansaloni
University of Lugano (USI), Switzerland

Web chair
Giacomo Toffetti Carughi
University of Lugano (USI), Switzerland

Location

Modularity'14 will be hosted by the Faculty of Informatics, University of Lugano (USI), Switzerland.

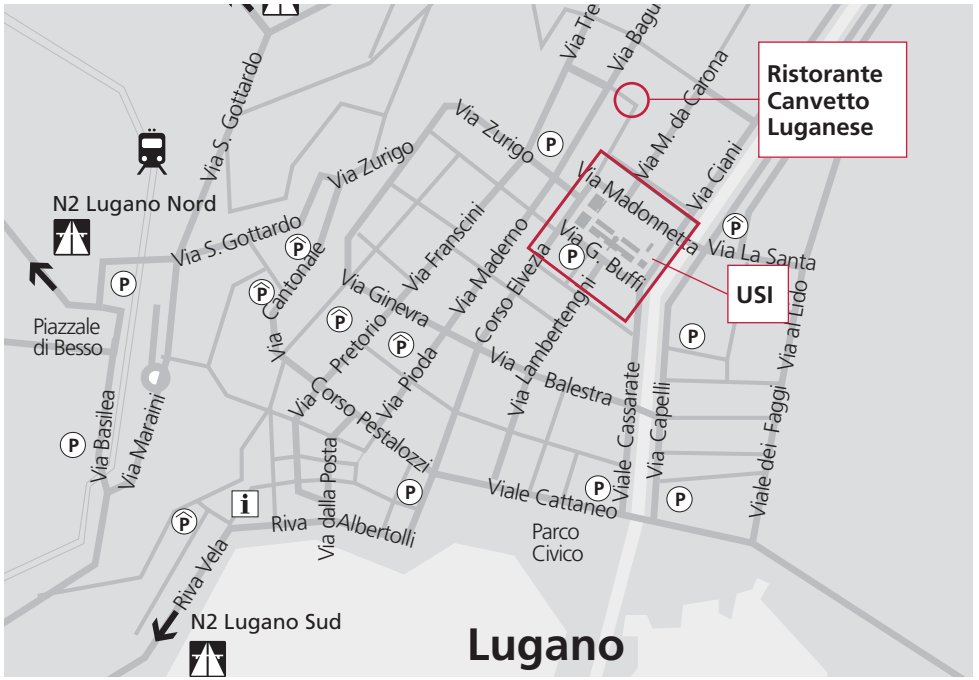
The conference will be located in the **USI Auditorium**, the entrance is on the **3rd floor of the USI Main Building**.



Room 351
Room 321

Meals and Social Events

Reception (Tuesday), lunches (Wednesday-Friday), and dinner banquet (Thursday) will be held at Ristorante Canvetto Luganese (via Rinaldo Simen 14b, 6900 Lugano).
A dedicated dining room will be reserved for the Modularity'14 participants.



Excursion to Cantina Monti (www.cantinamonti.ch)

The excursion will introduce the visitors to the world of the award-winning Swiss winemaker Monti. Switzerland does not have a huge territory, and the morphology of the country makes the growing of grapes more challenging. Thus, it takes passion and perseverance to make good wine here. A degustation of Monti's main products with all the related explanations will be part of the visit. Guests will be able to purchase wine and, depending on quantities, Monti will ship it directly to their homes.

Access to all meals and social events only with the conference badge.

Useful Information

Contact and Venue Address

Modularity'14
Faculty of Informatics
Via G. Buffi 13
6900 Lugano
Switzerland
organization@aosd.net
http://aosd.net/2014

Registration Desk

The registration desk will be open as follows in the foyer of the main building:

Tuesday:	13.00-18.30
Wednesday:	8.00-17.00
Thursday:	8.00-15.30
Friday:	8.00-16.00

Internet Access

Conference delegates have access to a dedicated wi-fi network:
SSID: Modularity14
Password: Lugano2014

Guest tickets

Additional tickets for the Banquet which have been pre-booked via e-mail must be paid in cash at the Registration Desk until Wednesday 10:30.

Emergency numbers

Modularity'14:	+41 58 666 46 90
USI Campus Security:	+41 58 666 47 30 **(17.00-7.00)**
Ambulance:	144
Police:	117