Welcome to the first volume of *Transactions on Aspect-Oriented Software Development*. Aspect-oriented methods, tools and techniques are gaining in popularity due to their systematic support for modularizing broadly scoped properties, the so-called *crosscutting concerns*, in software systems. Such crosscutting concerns include security, distribution, persistence, mobility, real-time constraints and so on. As software systems become increasingly ubiquitous, mobile and distributed, the modular treatment of such crosscutting concerns also becomes critical to ensure that software artifacts pertaining to such concerns are reusable, evolvable and maintainable. This modular treatment of crosscutting concerns by aspect-oriented techniques is not limited to code level. In fact, aspect-oriented techniques cover the software life cycle, handling crosscutting concerns in requirements, architecture, design, code, test cases, system documentation, etc.

The aspect-oriented software development community is growing fast, with an increasing number of researchers and practitioners across the world contributing to the development and evolution of the field. The community launched its own conference in 2002, which has since been held with great success on an annual basis. Recent reports from Burton and Gartner groups have put aspect-orientation on the *plateau of productivity* on the evolution cycle of new technologies. One of the key indicators of the maturity of a field is the availability of high quality research of an archival nature. The launch of *Transactions on Aspect-Oriented Software Development*, therefore, signifies a key milestone for the maturity of work in this area. The journal is committed to publishing work of the highest standard on all facets of aspect-oriented software development techniques in the context of all phases of the software life cycle, from requirements and design to implementation, maintenance and evolution. The call for papers is open indefinitely and potential authors can submit papers at any time to: taosd-submission@comp.lancs.ac.uk. Detailed submission instructions are available at: http://www.springer.com/sgw/cda/frontpage/0,,3-164-2-109318-0,00.html. A number of special issues on current important topics in the community are already in preparation. These include special issues on AOP systems, software and middleware; AOP and software evolution; dynamic AOP, and Early Aspects. Calls for such special issues are publicized on relevant Internet mailing lists, Web sites as well as conferences such as the Aspect-Oriented Software Development conference.

The articles in this volume cover a wide range of topics from software design to implementation of aspect-oriented languages. The first four articles address various issues of aspect-oriented modeling at the design level. The first article, “Assessing Aspect Modularizations Using Design Structure Matrix and Net Option Value”, by Lopes and Bajracharya, proposes a methodology and a tool to show how aspects can be beneficial as well as detrimental to a certain design. The second article, “Modularizing Design Patterns with Aspects: A Quantitative Study”, by Garcia et al., analyzes and compares the aspect-oriented and object-oriented implementations of design patterns with respect to quality values such as coupling and cohesion. The article “Directives for Composing Aspect-Oriented Class Models”, by Reddy et al., proposes models for expressing aspect-oriented and non–aspect-oriented properties of
systems and defines techniques to compose these models together. In the article “Aspect Categories and Classes of Temporal Properties”, Shmuel Katz defines a method for classifying aspects with respect to their temporal properties so that application of aspects in a system can be better understood and analyzed.

The following four articles discuss various programming language issues. The article “An Overview of CaesarJ”, by Aracic et al., gives an overview of the CaesarJ programming language, which aims at integrating aspects, classes and packages so that large-scale aspect components can be built. In the article “An Expressive Aspect Language for System Applications with Arachne”, Douence et al. motivate the applicability of the Arachne language in improving systems written in the C language, where system dynamicity and performance play an important role. Monteiro and Fernandes define in their article, “Towards a Catalogue of Refactorings and Code Smells for AspectJ”, a catalogue that helps in detecting aspects in object-oriented programs and in improving the structure of extracted aspects within the context of the AspectJ language. The final paper in the language category is “Design and Implementation of An Aspect Instantiation Mechanism” by Sakurai et al. It proposes association aspects as an extension to AspectJ for flexible descriptions of aspects whose instances are associated with more than one object.

The final article in this volume, “abc: An Extensible AspectJ Compiler”, by Avgustinov et al., describes a workbench for implementing aspect-oriented languages, so that easy experimentation with new language features and implementation techniques are possible.

The inception and launch of Transactions on Aspect-Oriented Software Development and publication of its first volume would not have been possible without the guidance, commitment and input of the editorial board and the reviewers who volunteered time from their busy schedules to help realize this publication. We thank them greatly for their help and efforts. Most important, we wish to thank authors who have submitted papers to the journal so far. The journal belongs to the community and it is the submissions from the community that are at the heart of this first volume and future volumes of Transactions on Aspect-Oriented Software Development.

Awais Rashid and Mehmet Aksit
Coeditors-in-chief
Organization

Editorial Board

Mehmet Aksit, University of Twente
Don Batory, University of Texas at Austin
Shigeru Chiba, Tokyo Institute of Technology
Siobhán Clarke, Trinity College Dublin
Theo D’Hondt, Vrije Universiteit Brussel
Robert Filman, Google
Shmuel Katz, Technion-Israel Institute of Technology
Gregor Kiczales, University of British Columbia
Karl Lieberherr, Northeastern University
Mira Mezini, University of Darmstadt
Ana Moreira, New University of Lisbon
Linda Northrop, Software Engineering Institute
Harold Ossher, IBM Research
Awais Rashid, Lancaster University
Douglas Schmidt, Vanderbilt University
David Thomas, Bedarra Research Labs

List of Reviewers

Jonathan Aldrich
Joao Araujo
Elisa Baniassad
Lodewijk Bergmans
Lynne Blair
Paulo Borba
Silvia Breu
Johan Brichau
Shigeru Chiba
Ruzanna Chitchyan
Siobhán Clarke
Yvonne Coady
Wesley Coelho
Maja D’Hondt
Theo D’Hondt
Pascal Dürr
Ulrich Eiselecker
Tzilla Elrad
Eric Ernst
Robert France
Alessandro Garcia
Andy Gokhale
Jeff Gray

Jean-Marc Jezequel
Joerg Kienzle
Micheal Kircher
Barbara Kitchenham
Shriram Krishnamurthi
Ramnivas Laddad
Karl Lieberherr
Roberto Lopez-Herrejon
David Lorenz
Hidehiko Masuhara
Marjan Mernik
Mattia Monga
Ana Moreira
Juan Manuel Murillo
Gail Murphy
Harold Ossher
Klaus Ostermann
Andres Diaz Pace
Monica Pinto
Ragghu Reddy
Christa Schwanninger
Domink Stein
Stan Sutton
### Organization

<table>
<thead>
<tr>
<th>Name</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Grundy</td>
<td>Wim Vanderperren</td>
</tr>
<tr>
<td>Charles Haley</td>
<td>Kris de Volder</td>
</tr>
<tr>
<td>Stephan Hannenberg</td>
<td>Robert Walker</td>
</tr>
<tr>
<td>Jan Hannemann</td>
<td>Nathan Weston</td>
</tr>
<tr>
<td>Wilke Havinga</td>
<td>Jianjun Zhao</td>
</tr>
</tbody>
</table>
# Table of Contents

Assessing Aspect Modularizations Using Design Structure Matrix and Net Option Value  
*Cristina Videira Lopes, Sushil Krishna Bajracharya* .............................. 1

Modularizing Design Patterns with Aspects: A Quantitative Study  
*Alessandro Garcia, Cláudio Sant'Anna, Eduardo Figueiredo,  
Uirá Kulesza, Carlos Lucena, Arndt von Staa* ............................... 36

Directives for Composing Aspect-Oriented Design Class Models  
*Y.R. Reddy, S. Ghosh, R.B. France, G. Straw, J.M. Bieman,  
N. McEachen, E. Song, G. Geory* ........................................... 75

Aspect Categories and Classes of Temporal Properties  
*Shmuel Katz* ................................................................. 106

An Overview of CaesarJ  
*Ivica Aracic, Vaidas Gasiunas, Mira Mezini, Klaus Ostermann* .... 135

An Expressive Aspect Language for System Applications with Arachne  
*Rémi Douence, Thomas Fritz, Nicolas Loriant, Jean-Marc Menaud,  
Marc Séguera-Devillechaise, Mario Südholt* ................................ 174

Towards a Catalogue of Refactorings and Code Smells for AspectJ  
*Miguel P. Monteiro, João M. Fernandes* ................................. 214

Design and Implementation of an Aspect Instantiation Mechanism  
*Kouhei Sakurai, Hidehiko Masuhara, Naoyasu Ubayashi,  
Saeko Matuura, Seiichi Komiya* ........................................... 259

*abc*: An Extensible AspectJ Compiler  
*Pavel Avgustinov, Aske Simon Christensen, Laurie Hendren,  
Sascha Kuzins, Jennifer Lhoták, Ondřej Lhoták, Oege de Moor,  
Damien Sereni, Ganesh Sittampalam, Julian Tibble* .................... 293

**Author Index** ................................................................. 335