Lecture Notes in Computer Science 8706

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Benoît Combemale David J. Pearce
Olivier Barais Jurgen J. Vinju (Eds.)

Software Language Engineering

7th International Conference, SLE 2014
Västerås, Sweden, September 15-16, 2014
Proceedings

Springer

Volume Editors

Benoît Combemale
Olivier Barais
IRISA
Campus de Beaulieu
Rennes, France
E-mail:{benoit.combemale, barais}@irisa.fr

David J. Pearce
Victoria University of Wellington
School of Engineering and Computer Science
Wellington, New Zealand
E-mail: david.pearce@ecs.vuw.ac.nz

Jurgen J. Vinju
University of Amsterdam
Science Park 123
Amsterdam, The Netherlands
E-mail: jurgen.vinju@cwi.nl

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-319-11244-2 e-ISBN 978-3-319-11245-9
DOI 10.1007/978-3-319-11245-9
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014947963

LNCS Sublibrary: SL 2 – Programming and Software Engineering

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

We cheerfully welcome you to SLE 2014, the 7th International Conference on Software Language Engineering, September 15–16, 2014 in Västerås, Sweden! We have worked to put together a program that has broad appeal to researchers, industrial practitioners, students, and educators in the field of software language engineering. The conference was also co-located with the 29th IEEE/ACM International Conference on Automated Software Engineering (ASE 2014) and the 13th International Conference on Generative Programming and Component Engineering (GPCE 2014), along with two workshops: the Industry Track on Software Language Engineering (ITSLE) and the Parsing@SLE workshop.

The SLE conference series is devoted to a wide range of topics related primarily to the use of artificial languages in software engineering. SLE brings together several communities that have traditionally looked at software languages from different and yet complementary perspectives: programming languages, model driven engineering, domain specific languages, and semantic web. Furthermore, SLE crosses a number of different technological spaces, including: context-free grammars, object-oriented modeling frameworks, rich data, structured data, object-oriented programming, functional programming, logic programming, term-rewriting, attribute grammars, algebraic specification, etc. Supporting these communities in learning from each other, and transferring knowledge is the guiding principle behind the organization of SLE.

The conference program included a keynote presentation, 16 technical paper presentations, and 3 tool paper demonstrations. The invited speaker was Prof. Colin Atkinson (University of Mannheim, Germany), with a talk entitled "From Language Engineering to Viewpoint Engineering". An extended abstract of the keynote presentation is also included in the conference proceedings.

We received 64 full submissions from 75 abstract submissions. From these submissions, the Program Committee (PC) eventually selected 19 papers: 16 out of 53 research papers (for an acceptance rate of 30%), and 3 out of 8 tool papers (for an acceptance rate of 37%). Each submitted paper was reviewed by at least three PC members and discussed in detailed during the electronic discussion period. Awards were also given out as part of the program for the overall best paper, the overall best student paper and the best reviewer.

SLE 2014 would not have been possible without the significant contributions of many individuals and organizations. The SLE Steering Committee provided invaluable assistance and guidance, whilst the Program Committee (and additional reviewers) undertook with dedication the critical task of reviewing and discussing the submissions. We are also grateful to members of the Organizing Committee for making the necessary arrangements and helping to publicize the conference and prepare the proceedings. We thank the authors for their efforts in writing and revising their papers in accordance with feedback from the

reviewers. We would also like to thank our sponsors: Google (main sponsor), the GEMOC initiative, and Itemis. Finally, we would also like to thank the hosting organization, Mälardalen University.

We hope you enjoy the conference!

August 2014

Benoit Combemale
David J. Pearce
Olivier Barais
Jurgen Vinju

Organization

Program Committee

Emilie Balland	Inria, France
Tony Clark	Middlesex University, UK
Zinovy Diskin	McMaster University/University of Waterloo, Canada
Martin Erwig	Oregon State University, USA
Anne Etien	University of Lille, France
Joerg Evermann	Memorial University of Newfoundland, Canada
Jean-Marie Favre	University of Grenoble, France
Robert France	Colorado State University, USA
Andy Gill	University of Kansas, USA
Martin Gogolla	University of Bremen, Germany
Pieter Van Gorp	Eindhoven University of Technology, The Netherlands
Giancarlo Guizzardi	Federal University of Espirito Santo, Brazil
Gorel Hedin	Lund University, Sweden
Markus Herrmannsdoerfer	Technische Universität München, Germany
Jean-Marc Jézéquel	University of Rennes, France
Thomas Kuehne	Victoria University of Wellington, New Zealand
Ralf Laemmel	Universität Koblenz-Landau, Germany
Peter Mosses	Swansea University, UK
Sean Mcdirmid	Microsoft, China
Kim Mens	Université catholique de Louvain, Belgium
Marjan Mernik	University of Maribor, Slovenia
Pierre-Alain Muller	University of Haute-Alsace, France
Nathaniel Nystrom	University of Lugano, Switzerland
Klaus Ostermann	University of Marburg, Germany
Oscar Nierstrasz	University of Bern, Switzerland
Richard Paige	University of York, UK
Fiona Polack	University of York, UK
Arnd Poetzsch-Heffter	University of Kaiserslautern, Germany
Davide Di Ruscio	Università degli Studi dell'Aquila, Italy
João Saraiva	Universidade do Minho, Portugal
Bran Selic	Malina Software Corp., Canada
Jim Steel	University of Queensland, Australia
Tijs Van der Storm	Centrum Wiskunde & Informatica, The Netherlands
Juha-Pekka Tolvanen	MetaCase, Finland

Michael Whalen	University of Minnesota, USA
Eric Van Wyk	University of Minnesota, USA
Steffen Zschaler	King's College London, UK

Additional Reviewers

Al Lail, Mustafa
Al-Refai, Mohammed
Allen, Wyatt
Barais, Olivier
Bennett, Phillipa
Bieniusa, Annette
Brauner, Paul
Brunnlieb, Malte
Büttner, Fabian
Chen, Sheng
Chis, Andrei
Degueule, Thomas
Feller, Christoph
Fernandes, Joao
Fors, Niklas
Hamann, Lars
Hilken, Frank
Inostroza, Pablo
Kuhlmann, Mirco
Kurnia, Ilham
Kurs, Jan
Lukyanenko, Roman
Martins, Pedro
Mendez, David
Milojković, Nevena
Mukkamala, Raghava Rao
Nan, Shan
Osman, Haidar
Oumarou, Hayatou
Passos, Leo
Pereira, Rui
Pierantonio, Alfonso
Polito, Guillermo
Santos, Gustavo
Smeltzer, Karl
Sun, Wuliang
Teruel, Camille
van der Ploeg, Atze
van Rozen, Riemer
Vinju, Jurgen
Vojtisek, Didier
Walkingshaw, Eric
Weber, Mathias

From Language Engineering to Viewpoint Engineering (Invited Talk)

Colin Atkinson

University of Mannheim
B6, Mannheim, Germany

Abstract. As software systems increase in size and complexity, and are expected to cope with ever more quantities of information from ever more sources, there is an urgent and growing need for a more view-oriented approach to software engineering. Views allow stakeholders to see exactly the right information, at exactly the right time, in a way that best matches their capabilities and goals. However, this is only possible if the information is represented in the optimal languages (i.e. domain- and purpose-specific), with the necessary context information and the optimal manipulation/editing features - that is, if information is viewed from the optimal viewpoints. Rather than merely engineering languages, therefore, software engineers in the future will need to engineer viewpoints, which augment language definitions (e.g. meta-models, syntax ...) with context information (e.g. elision, location, perspective ...) and user-interaction information (e.g. editing pallets, view manipulation services ...). In this talk Colin Atkinson will outline the issues faced in supporting the flexible and efficient engineering of viewpoints and will present some key foundations of a fundamentally view-oriented approach to software engineering.

Keywords: Viewpoint engineering, separation of concerns

As software systems increase in size and complexity, and are expected to cope with ever more quantities of information from ever more sources, there is an urgent and growing need for a more view-oriented approach to software engineering. Views allow stakeholders to see exactly the right information, at exactly the right time, in a way that best matches their capabilities and goals. Domain-specific languages are a key foundation for supporting views by allowing them to display their contents in a customized way, but the current generation of software language engineering technologies do not go far enough. In particular, they currently lack the ability to convey the precise relationship between the information shown in a view and the information it is a view of. They also focus on describing how model elements should be visualized but provide little or no support for describing how stakeholders should edit and interact with them.

The premise of this talk is that software language engineering technologies need to evolve to support an enhanced approach to modeling in which model content can be set in context relative to the underlying source from which it is de-

rived – an approach we refer to as "contextualized modeling". These technologies would then be more accurately characterized as "view engineering" technologies rather than "language engineering"" technologies since they would support all aspects of view definition, including the context in which the content is to be interpreted and the mechanisms by which model elements are to be visualized and edited. Some of the key additional capabilities that the current generation of language engineering technologies need to support in order to become viewpoint engineering languages include -

Enriched Designation. The most important context information in a view is its model elements' location in the three key hierarchies of the underlying information model – the classification hierarchy, the inheritance hierarchy and the containment (i.e. ownership) hierarchy. These are supported to various degrees in today's language engineering technologies through a mix of explicit symbolism and location-defining designators (a.k.a. headers) in model elements. However, they are not supported in a uniform and consistent way, and are often severely limited in what they can express. In particular most contemporary language engineering technologies only allow one level of classification to be expressed at a time. Fully contextualized modeling requires a comprehensive, systematic and deep designation notation which allows a model element's exact location in each hierarchy to be expressed in its designator.

Explicit Elision Symbolism. Since views almost always convey only a subset of the information contained in the underlying model, an important requirement in viewpoint engineering is to support the description of what things are not included in a view, as well as the description of what things are. This is a challenging task since it involves subtle interactions between explicit omission statements (e.g. "...." in UML generalization sets), explicit completeness statements (e.g. complete and disjoint in UML generalization sets) and background "world" assumptions (e.g. "open world" versus "closed world" assumption). Fully contextualized modeling therefore requires comprehensive and systematic support for elision, both in the form of explicit elision symbols and elided model element designators.

Explicit Derivation Symbolism. As well as omitting information from the underlying subject of a view it is possible to derive new information that the subject does not explicitly contain. Such derivation operations can be driven by the application of basic characterization relationships such as inheritance and classification (e.g. subsumption) or by more complex inference operations based on the principles of logic. In both cases, contextualized modeling must incorporate the ability to express what information in a view has been derived and what information has been explicitly asserted by a human modeller. This is important for resolving conflicts and signalling the weight that should be given to the information represented within views.

Language Symbiosis. Domain-specific representations of information have the advantage that they are optimized for particular classes of stakeholders or communities of experts, whereas general-purpose languages have the ad-

vantage that they are widely known and can represent information in quasi-standard ways. In order to enjoy both benefits simultaneously, contextualized models should be represented by highly flexible, symbiotic languages that allow different visualizations of model elements to be mixed and interchanged at will.

Viewpoint Environment Definition. A user's experience of a view is determined not only by the way in which its contents are displayed, but also by the way in which the user can interact with the model and, when it is editable, input information. This impacts all aspects of the environment in which the view is displayed, including the menu items, the pallets of predefined types and models elements and the range of operations that can be applied to the content (e.g. checking, printing, persisting etc.). The engineering of viewpoints therefore involves much more than just the engineering of languages it also involves the definition of the associated interaction experience.

In this talk Colin Atkinson will introduce the vision of contextualized modeling and explain these key ingredients needed to turn the current generation of software language engineering technologies into fully fledged viewpoint-engineering technologies

Biography

Colin Atkinson has been the leader of the Software Engineering Group at the University of Mannheim since April 2003. Before that he has held positions at the University of Kaiserslautern, the Fraunhofer Institute for Experimental Software Engineering and the University of Houston - Clear Lake. His research interests are focused on the use of model-driven and component based approaches in the development of dependable and adaptable computing systems. He was a contributor to the original UML development process and is one of the original developers of the deep (multi-level) approach to conceptual modelling. He received his Ph.D. and M.Sc. in computer science from Imperial College, London, in 1990 and 1985 respectively, and his B.Sc. in Mathematical Physics from the University of Nottingham 1983.

Table of Contents