

Enabling Software Evolution Via AOP and Reflection

Report on the Workshop RAM-SE at ECOOP 2007

Manuel Oriol¹, Walter Cazzola², Shigeru Chiba³, Gunter Saake⁴,
Yvonne Coady⁵, Stéphane Ducasse⁶, and Günter Kiesel⁷

¹ ETH Zurich, Zurich, Switzerland
moriol@inf.ethz.ch

² Università degli Studi di Milano, Milano, Italy
cazzola@dico.unimi.it

³ Tokyo Institute of Technology, Tokyo, Japan
chiba@is.titech.ac.jp

⁴ Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany
saake@iti.cs.uni-magdeburg.de

⁵ University of Victoria, Victoria, Canada
ycoady@cs.uvic.ca

⁶ INRIA, Lille, France

stephane.ducasse@inria.fr

⁷ University of Bonn, Bonn, Germany
gk@cs.uni-bonn.de

Abstract. Following last three years' RAM-SE (Reflection, AOP and Meta-Data for Software Evolution) workshop at the ECOOP conference, the RAM-SE'07 workshop was a successful and popular event. As its name implies, the workshop's focus was on the application of reflective, aspect-oriented and data-mining techniques to the broad field of software evolution. Topics and discussions at the workshop included mechanisms for supporting software evolution, technological limits for software evolution and tools and middleware for software evolution. The workshop's main goal was to bring together researchers working in the field of software evolution with a particular interest in reflection, aspect-oriented programming and meta-data. The workshop was organized as a full day meeting, partly devoted to presentation of submitted position papers and partly devoted to panel discussions about the presented topics and other interesting issues in the field. In this way, the workshop allowed participants to get acquainted with each other's work, and stimulated collaboration.

1 Workshop Description and Objectives

Software evolution and adaptation is a research area that offers stimulating challenges for both academic and industrial researchers. The evolution of software systems, to face unexpected situations or just for improving their features, relies

on software engineering techniques and methodologies. Nowadays a similar approach is not applicable in all situations e.g., for evolving nonstopping systems or systems whose code is not available.

Features of reflection such as transparency, separation of concerns, and extensibility seem to be perfect tools to aid the dynamic evolution of running systems. Aspect-oriented programming (AOP) can simplify code instrumentation whereas techniques that rely on meta-data can be used to inspect the system and to extract the necessary data for designing the heuristic that the reflective and aspect-oriented mechanism use for managing the evolution.

We feel the necessity to investigate the benefits brought by the use of these techniques on the evolution of object-oriented software systems. In particular we would determine how these techniques can be integrated with more traditional approaches to evolve a system and the benefits we get from their use.

The overall goal of this workshop was that of supporting circulation of ideas between these disciplines. Several interactions were expected to take place between reflection, aspect-oriented programming and meta-data for the software evolution, some of which we cannot even foresee. Both the application of reflective or aspect-oriented techniques and concepts to software evolution are likely to support improvement and deeper understanding of these areas. This workshop has represented a good meeting-point for people working in the software evolution area, and an occasion to present reflective, aspect-oriented, and meta-data based solutions to evolutionary problems, and new ideas straddling these areas, to provide a discussion forum, and to allow new collaboration projects to be established. The workshop was a full day meeting. One part of the workshop was devoted to presentation of papers, and another to panels and to the exchange of ideas among participants.

2 Workshop Topics and Structure

Every contribution that exploits reflective techniques, aspect-oriented programming and/or meta-data to evolve software systems were welcome. Specific topics of interest for the workshop have included, but were not limited to:

- aspect-oriented middleware and environments for software evolution;
- adaptive software components and evolution as component composition;
- evolution planning and deployment through aspect-oriented techniques and reflective approaches;
- aspect interference and composition for software evolution;
- feature- and subject-oriented adaptation;
- unanticipated software evolution supported by AOSD or reflective techniques;
- MOF, code annotations and other meta-data facilities for software evolution;
- software evolution tangling concerns;
- techniques for refactoring into AOSD and to get the separation of concerns;
- early aspect evolution, i.e., to design evolution by evolving the design information or the application in its early stages of development.

To ensure lively discussion at the workshop, the organizing committee has chosen the contributions on the basis of topic similarity that will permit the beginning of new collaborations. To grant an easy dissemination of the proposed ideas and to favorite an ideas interchange among the participants, accepted contributions are freely downloadable from the workshop web page:

<http://homes.dico.unimi.it/RAM-SE07.html>.

The workshop was a full day meeting organized in four sessions. The first session was devoted to the Shigeru Chiba's keynote speech on "*How We Should Use Aspects*". Each of the remaining sessions has been characterized by a dominant topic that perfectly describes the presented papers and the related discussions. The two dominant topics were: *Classic Software Evolution*, and *Aspect-Oriented and Reflection for Software Evolution*. During each session, paper presentations took 15 minutes with a 5 minutes discussion. At the end of the day a special session was devoted to discussions. The discussion related to each session has been brilliantly lead respectively by Mario Südholt, Walter Cazzola, Manuel Oriol and Gunter Saake.

The workshop has been very lively, the debates very stimulating, and the high number of participants (see appendix A) testifies the interest in the application of reflective, aspect- and meta-data oriented techniques to software evolution as well as software evolution in general.

3 Important References

The following publications are important references for people interested in learning more about the topics of this workshop:

- Pattie Maes. Computational Reflection. PhD thesis, Vrije Universiteit Brussel, Brussels, Belgium, 1987.
- Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Videira Lopes, Jean-Marc Loingtier, and John Irwin. Aspect-Oriented Programming. In *11th European Conference on Object Oriented Programming (ECOOP'97)*, LNCS 1241, pages 220–242, Helsinki, Finland, June 1997. Springer-Verlag.
- The proceedings of the International Conference on Aspect-Oriented Software Development (AOSD) from 2002 onward. See also <http://aosd.net/archive/index.php>.
- Several tracks related to aspect-oriented software development and evolution at the International Conference on Software Maintenance (ICSM) and the Working Conference on Reverse Engineering (WCRE), from 2002 onward.
- The software evolution website at the Program Transformation wiki:

<http://www.program-transformation.org/twiki/bin/view/Transform/SoftwareEvolution>.

- The workshops proceedings of the USE workshop series:

<http://www.informatik.uni-bonn.de/~gk/use/>.

4 Workshop Overview: Session by Session

Session on How We Should Use Aspects

In the first morning session, Shigeru Chiba gave a keynote talk that was moderated by Mario Südholt:

How We Should Use Aspects

Abstract. *Besides classic logging and the observer pattern, several applications of aspect-oriented programming (AOP) have been proposed so far. This talk reviews those applications and discusses what properties of AOP are significant and promising for software evolution. It will also discuss what are unique features of AOP against related technology such as reflection and mixin layers.*

Chiba's provocative talk presented several applications where AOP should be used. The first application presented is logging as it consists of a multitude of similar calls that can be located anywhere in the code. This is one of the possibilities used by IBM field engineers. The Aspect-Oriented interactive debuggers [1], high performance computing [2] could also be interesting. Application-level scheduling [3] is also a possible use of AOP and as an example, the application level scheduling achieved better performances than the Linux scheduler. Non-functional requirements do not appear to be the best fields to which aspects can be best applied as they are generally very class centric. One of the lessons drawn is that most aspects are heterogeneous and thus AOP does not avoid iterative code of programming idioms. Another lesson is that pointcut advices are seldom used, thus the question is if pointcut is a primary mechanism.

The keynote talk fostered further discussions which triggered the following points:

- It was suggested that AOP is better than the meta-object protocol (MOP) for persistence.
- The lessons outlined by Chiba are mostly valid because he is drawing them from AspectJ, using another aspect language (e.g., a higher level one) could lead to very different results.
- There are a handful of applications that use AOP (MySQL, WebSphere,...) while many applications use MOP. The reason for such a fact may be that the AOP community did not focus on a simple set of “evident” applications to help people getting in the methodology.

Session on Classic Software Evolution

Classical software evolution was the main focus of the second session. The session was moderated by Walter Cazzola.

[4] Toward Computer-Aided Usability Evaluation Evolving Interactive Software. *Yonglei Tao* (Grand Valley State University, USA).

Yonglei Tao gave the presentation.

- [5] Towards Runtime Adaptation in a SOA Environment. *Florian Irmert, Marcus Meyerhofer* and *Markus Weiten* (Friedrich-Alexander Universität Erlagen-Nürnberg, Germany).

Florian Irmert gave the presentation.

- [6] IDE-integrated Support for Schema Evolution in Object-Oriented Applications. *Marco Piccioni, Manuel Oriol*, and *Betrand Meyer* (ETH Zürich, Switzerland).

Marco Piccioni gave the presentation.

- [7] Property-preserving Evolution of Components Using VPA-Based Aspects. *Dong Ha Nguyen* and *Mario Südholt*, Ecole des Mines de Nantes, France.

Mario Südholt gave the presentation.

Session on Aspect-Oriented and Reflection for Software Evolution

Aspect-oriented and reflection for software evolution was the main focus of the third session. The session was moderated by Manuel Oriol.

- [8] Characteristics of Runtime Program Evolution. *Mario Pukall*, and *Martin Kuhlemann* (Otto von Guericke University Magdeburg, Germany).

Mario Pukall gave the presentation.

- [9] Aspect-Based Introspection and Change Analysis for Evolving Programs. *Kevin Hoffman, Murali Krishna Ramanathan, Patrick Eugster*, and *Suresh Jagannathan* (Purdue University, USA).

Kevin Hoffman gave the presentation.

- [10] Morphing Software for Easier Evolution. *Shan Shan Huang* and *Yannis Smaragdakis* (University of Oregon, USA).

Yannis Smaragdakis gave the presentation.

- [11] AOP vs Software Evolution: a Score in Favor of the Blueprint. *Walter Cazzola* (DICO Università degli Studi di Milano, Italy), and *Sonia Pini* (DISI Università degli Studi di Genova, Italy).

Sonia Pini gave the presentation.

Session on Future Evolutions of RAM-SE

The workshop ended with a session led by Gunter Saake on the future of the RAM-SE workshop and fostered lively discussions. Most of the discussion focused on the fact that aspects are polarizing people either positively or very negatively. In order to develop aspects further, it is needed to show very simple examples in which aspects have an evident applicability and ease the task. Even if it is probably not possible to find use cases where only aspects could solve the problem there are numerous areas in which the most elegant solution would use aspects. The cost of having people understand and use aspects is probably to propose and advocate emblematic simplifications. This is inspired by reflection being what probably stands out of the meta-object protocol.

5 Tendencies in Reflection, AOP and Meta-data for Software Evolution

The workshop outlined at least three major areas which are currently active:

- Evolution enabling technologies.
- Applications of aspects to understanding or controlling the evolution of programs.
- New trends in AOP for smooth evolution.

The first area consists of technologies that improve the direct evolution capabilities of programs. As such, the work of Piccioni *et al.* on schema evolution [6] enables the easy programming of persistent applications by using reflexive techniques. The work of Irmert *et al.* on dynamic adaptation of applications through the use of dynamic aspects [5] opens new directions in the runtime evolution of applications. The work from Nguyen *et al.* on the dynamic evolution of pushdown automata [7] opens new leads in the correctness of dynamic updates. The work of Pukall *et al.* [8] analyzes languages and technologies according to the time of evolution and the type of evolution that they enable while effecting runtime evolution.

The second area consists of applications of the aspects technologies that ease the understanding of the state or of the evolution of programs. For example the work by Tao on the evaluation of usability of interactive software [4] uses aspects to trace users actions. The work by Hoffman *et al.* [9] instruments programs using aspects to gather information on the state of a program while it executes in order to understand the changes that were performed at runtime when updated.

The third area consists of new trends in AOP that ease the evolution of applications coded with aspects. As an example the work by Huang *et al.* [10] allows for a better evolvability of programs by defining the morphing technique. The work of Cazzola *et al.* [11] proposes a new aspect language that solves the fragile pointcut issue and evaluate the solution with evolving programs.

6 Final Remarks

The main goal of the workshop was to bring together researchers interested in the field and have them communicate on their respective work. The workshop lived up to its expectations, with high-quality submissions and presentations, and lively and stimulating discussions. The vitality of the work as well as the lively discussions that took place during the workshop show that the issues addressed by the workshop are plainly relevant and need such a forum to be discussed. We hope participants found the workshop interesting and useful, and encourage them to finalize their position papers and submit them as full papers to international conferences interested in the topics of this workshop.

Acknowledgements. We wish to thank Mario Südholt both for his interest in the workshop and for his help during the workshop as chairman and speaker. We wish also to thank all the researchers that have participated to the workshop.

We have also to thank the Department of Informatics and Communication of the University of Milan, the Department of Mathematical and Computing Sciences of the Tokyo institute of Technology, ETH Zurich and the Institute für Technische und Betriebliche Informationssysteme, Otto-von-Guericke-Universität Magdeburg for their various supports.

References

1. Usui, Y., Chiba, S.: Bugdel: An aspect-oriented debugging system. In: APSEC 2005. 12th Asia-Pacific Software Engineering Conference, December 15-17, 2005, Taipei, Taiwan, pp. 790–795 (2005)
2. Nishizawa, M., Chiba, S., Tatsubori, M.: Remote pointcut: a language construct for distributed aop. In: AOSD 2004. Proceedings of the 3rd International Conference on Aspect-Oriented Software Development, pp. 7–15 (2004)
3. Kourai, K., Hibino, H., Chiba, S.: Aspect-oriented application-level scheduling for j2ee servers. In: AOSD 2007. Proceedings of the 6th International Conference on Aspect-Oriented Software Development, pp. 1–13 (2007)
4. Tao, Y.: Toward Computer-Aided Usability Evaluation Evolving Interactive Software. In: In Cazzola, W., Chiba, S., Coady, Y., Ducasse, S., Kniesel, G., Oriol, M., Saake, G. (eds.) Proceedings of ECOOP 2007. Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2007), Berlin, Germany (2007)
5. Irmert, F., Meyerhöfer, W.M.: Towards Runtime Adaptation in a SOA Environment. In: Cazzola, W., Chiba, S., Coady, Y., Ducasse, S., Kniesel, G., Oriol, M., Saake, G. (eds.) Proceedings of ECOOP 2007 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2007), Berlin, Germany (2007)
6. Piccioni, M., Oriol, M., Meyer, B.: IDE-integrated Support for Schema Evolution in Object-Oriented Applications. In: Cazzola, W., Chiba, S., Coady, Y., Ducasse, S., Kniesel, G., Oriol, M., Saake, G. (eds.) Proceedings of ECOOP 2007 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2007), Berlin, Germany (2007)

7. Nguyen, D.H., Südholt, M.: Property-preserving Evolution of Components Using VPA-Based Aspects. In: Cazzola, W., Chiba, S., Coady, Y., Ducasse, S., Kniesel, G., Oriol, M., Saake, G. (eds.) Proceedings of ECOOP 2007 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2007), Berlin, Germany (2007)
8. Pukall, M., Kuhlemann, M.: Characteristics of Runtime Program Evolution. In: Cazzola, W., Chiba, S., Coady, Y., Ducasse, S., Kniesel, G., Oriol, M., Saake, G. (eds.) Proceedings of ECOOP 2007 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2007), Berlin, Germany (2007)
9. Hoffman, K., Ramanathan, M.K., Eugster, P., Jagannathan, S.: Aspect-Based Introspection and Change Analysis for Evolving Programs. In: Cazzola, W., Chiba, S., Coady, Y., Ducasse, S., Kniesel, G., Oriol, M., Saake, G. (eds.) Proceedings of ECOOP 2007 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2007), Berlin, Germany (2007)
10. Huang, S.S., Smaragdakis, Y.: Morphing Software for Easier Evolution. In: Cazzola, W., Chiba, S., Coady, Y., Ducasse, S., Kniesel, G., Oriol, M., Saake, G. (eds.) Proceedings of ECOOP 2007 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2007), Berlin, Germany (2007)
11. Cazzola, W., Pini, S.: AOP vs Software Evolution: a Score in Favor of the Blueprint. In: Cazzola, W., Chiba, S., Coady, Y., Ducasse, S., Kniesel, G., Oriol, M., Saake, G. (eds.) Proceedings of ECOOP 2007 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2007), Berlin, Germany (2007)

A Workshop Attendee

The success of the workshop is mainly due to the people that have attended it and to their effort to participate to the discussions. The following is the list of the attendees in alphabetical order.

Name	Affiliation	Country	e-mail
Blair, Gordon	Lancaster University	United Kingdom	gordon@comp.lancs.ac.uk
Cazzola, Walter	Università degli Studi di Milano	Italy	cazzola@ dico.unimi.it
Chiba, Shigeru	Tokyo Institute of Technology	Japan	chiba@is.titech.ac.jp
Greenwood, Phil	Lancaster University	United Kingdom	greenwood@comp.lancs.ac.uk
Hoffman, Kevin	Purdue University	USA	Kevin.John.Hoffman@gmail.com
Huang, Shan Shan	Georgia Tech	USA	ssh@cc.gatech.edu
Irmert, Florian	Universität Erlagen-Nürnberg	Germany	florian.irmert@informatik.uni-erlangen.de
Kienle, Holger	University of Victoria	Canada	kienle@cs.uvic.ca
Masuhara, Hidehiko	University of Tokyo	Japan	masuhara@graco.c.u-tokyo.ac.jp
Mens, Kim	Université Catholique de Louvain	Belgium	km@info.ucl.ac.be
Mosser, Sebastian	University of Nice/CNRS	France	mosser@polytech.unice.fr
Oriol, Manuel	ETH Zürich	Switzerland	moriol@inf.ethz.ch
Piccioni, Marco	ETH Zürich	Switzerland	marco.piccioni@inf.ethz.ch
Pini, Sonia	Università degli Studi di Genova	Italy	pini@disi.unige.it
Pukall, Mario	University of Magdeburg	Germany	pukall@iti.cs.uni-magdeburg.de
Rashid, Awais	Lancaster University	United Kingdom	marash@comp.lancs.ac.uk
Südholt, Mario	École des Mines de Nantes	France	sudholt@emn.fr
Sørensen, Fredrik	University of Oslo	Norway	fredrso@ifi.uio.no
Saake, Gunter	University of Magdeburg	Germany	saake@iti.cs.uni-magdeburg.de
Smaragdakis, Yannis	University of Oregon	USA	yannis@cs.uoregon.edu
Tanter, Eric	University of Chile	Chile	etanter@dcc.uchile.cl
Tao, Yonglei	Grand Valley State University	USA	taoy@gvsu.edu
Ueyama, Jô	University of Campinas	Brazil	joueyama@ic.unicamp.br
Yonezawa Akinori	University of Tokyo	Japan	yonezawa@is.s.u-tokyo.ac.jp