Getting Farther on Software Evolution via AOP and Reflection

Report on the 5th RAM-SE Workshop at ECOOP 2008

Manuel Oriol[1], Walter Cazzola[2], Shigeru Chiba[3], and Gunter Saake[4]

[1] Department of Computer Science,
University of York, York, United Kingdom
manuel@cs.york.ac.uk

[2] DICo - Department of Informatics and Communication,
Università degli Studi di Milano, Milano, Italy
cazzola@dico.unimi.it

[3] Department of Mathematical and Computing Sciences,
Tokyo Institute of Technology, Tokyo, Japan
chiba@is.titech.ac.jp

[4] Institute für Technische und Betriebliche Informationssysteme,
Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany
saake@iti.cs.uni-magdeburg.de

Abstract. Following last four years' RAM-SE (Reflection, AOP and Meta-Data for Software Evolution) workshop at the ECOOP conference, the RAM-SE'08 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 of the aspect-oriented and reflective approaches to software evolution and tools devoted to 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. We hope this helped participants in improving their ideas and the quality of their future publications.

The workshop's proceedings, including all accepted position papers can be downloaded from the workshop's web site and a post workshop proceeding, including an extension of the accepted paper is planned to be published by the University of Magdeburg.

In this report, we provide a session-by-session overview of the workshop, and then present our opinions about future trends in software evolution.

P. Eugster (Ed.): ECOOP 2008 Workshop Reader, LNCS 5475, pp. 63–69, 2009.

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 non stopping 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 favourite an ideas interchange among the participants, accepted contributions are freely downloadable from the workshop web page:

`http://homes.dico.unimi.it/RAM-SE08.html`

The workshop was a full day meeting organized in three sessions. The morning was devoted to scientific presentations with six refereed papers in the first session and with a keynote speech by hidehiko Masuhara in the second session just before lunch. In the afternoon the workshop became a working group lead by Shigeru Chiba, this has permitted to exchange new ideas in a lively discussion with the several people attending.

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

RAM-SE Paper Presentation

The first morning session focused on presenting accepted papers. The session was moderated by Manuel Oriol.

[**1**] A Case Study for Aspect Based Updating. *Susanne Cech Previtali* and *Thomas R. Gross* (ETH Zurich, Switzerland).

Susanne Cech Previtali gave the presentation.

[**2**] Runtime Adaptations within the QuaD^2-Framework. *Steffen Mencke, Martin Kunz,* and *Mario Pukall* (Otto von Guericke University Magdeburg, Germany).

Mario Pukall gave the presentation.

[**3**] Modeling Context-Dependent Aspect Interference Using Default Logics. *Frans Sanen, Eddie Truyen,* and *Wouter Joosen* (K.U. Leuven, Belgium).

Frans Sanen gave the presentation.

[**4**] Object Roles and Runtime Adaptation in Java. *Mario Pukall* (Otto von Guericke University Magdeburg, Germany).

Mario Pukall gave the presentation.

[**5**] Exploring Role Based Adaptation. *Sebastian Götz* and *Ilie Savga* (Dresden University of Technology, Germany).

Sebastian Götz gave the presentation.

[**6**] Annotations for Seamless Aspect Based SW Evolution. *Susanne Cech Previtali* and *Thomas R. Gross* (ETH Zurich, Switzerland).

Susanne Cech Previtali gave the presentation

Keynote on Toward Right Abstraction of Crosscutting Concerns

In the second session, Hidehiko Masuhara gave a keynote talk moderated by Walter Cazzola:

> **Toward Right Abstraction of Crosscutting Concerns.**
>
> **Abstract.** *Abstraction mechanisms in programming languages are crucial for modular software development, by drawing a clear boundary among program entities, giving names to those bounded entities, and hiding implementation details. Aspect-oriented programming (AOP) mechanisms can also be viewed as abstraction mechanisms for crosscutting concerns, but differ from traditional ones in what details they hide. In this talk, we discuss the properties of AOP mechanisms that are needed to be right abstraction of crosscutting concerns.*

Matsuhara's presentation gave elements on the definition of crosscutting abstraction as a boundary over the code. The main issue that aspects programmers have to face is that this boundary is actually a very difficult thing to draw precisely. In practice, it is easy to express pointcuts that have a clear mapping into the code, but more complex ones are almost inexpressible.

"Are we Doomed?" as Matsuhara asked. To help defining complex crosscutting concerns it is possible to consider example-based pointcuts. In particular, test-based pointcuts [7] can be of help. The idea is to use unit tests as the main way of defining pointcuts by analysing static execution history. This has several advantages, the main one being that if test cases are maintain, pointcuts should evolve automatically with the application.

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

- A boundary might not be the correct abstraction: what space/points can be considered?
- As a lot of people in the workshop saw aspects like a tool for dynamic adaptation, boundaries were usually very easily drawn, but would it be possible to create test-based pointcuts that would draw them?

RAM-SE'09? Towards the Future

The workshop ended with a session led by Shigeru Chiba on the future of the RAM-SE workshop and fostered lively discussions. Chiba pointed out that dynamic adaptation was one of the main topics this year. It was mostly coded with dynamic aspects and using dynamic aspects to allow adaptation in the future. He also pointed out the fight between statically typed languages and dynamically typed ones and wondered if we are statically typed dynamic languages people. The remaining part of the session was dedicated to how we understand systems that have been adapted over time. Do we need to do aspects refactoring? Should we replace the whole system when adapting it? In the end, are aspects just a tool that enables dynamic refactoring? While discussing all these topics, some pointed out that object-orientation is already aspect-oriented due to multiple dispatch. Aspects are indeed a tool, but they are very good for producing prototypes in a fast and convenient manner.

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

This year, the main area of the workshop was runtime adaptation through aspects. This is a radical shift in the community. It seems that using aspects is nowadays the easiest way to instrument code (at load-time). This meets last year's invited talk presented by Shigeru Chiba that pointed out that logging and transactions were actually the killer-applications for aspects. One more application seems to be the easy instrumentation of code. Because the workshop is targeted at evolution, aspects and reflection, it is thus not too surprising that contributions would focus on it this year.

One of the new adaptation proposed by Cech Previtali and Gross [1] consists in writing updates as an aspect. The paper presents a feasibility study based on the tomcat server and reveals that although all changes cannot be expressed as aspects most of them can. In order to guide the aspect weaver, Cech Previtali and Gross [6] even go as far as to propose annotations in the code so that updates are actually easier to interpret. To achieve a similar goal, Pukall [4] proposes a strategy based on object wrapping and the hotswap technology to change implementation.

Götz and Savga [5] also cope with adapters for objects. They propose a role-based mechanism to manage adapters that would reduce the code complexity.

Another aspect of dynamic adaptation is that adaptation can also be considered not as a timeline punctuated with versions, but rather as a potential for quality of service. Mencke *et al* [2] detail such a system and state that the quality of a combination of a components should govern the choice of components to use.

Sanen *et al* [3] treat a more fundamental question for aspects: how to model aspect interference by using default logic. This would allow developers to use the information when the application evolves.

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 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. Cech Previtali, S., Gross, T.: A case study for aspect based updating. In: Cazzola, W., Chiba, S., Coady, Y., Oriol, M. (eds.) Proceedings of ECOOP 2008 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2008), Paphos, Cyprus (2008)

2. Mencke, S., Kunz, M., Pukall, M.: Runtime adaptations within the QuaD^2-framework. In: Cazzola, W., Chiba, S., Coady, Y., Oriol, M. (eds.) Proceedings of ECOOP 2008 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2008), Paphos, Cyprus (2008)
3. Sanen, F., Truyen, E., Joosen, W.: Modeling context-dependent aspect interference using default logics. In: Cazzola, W., Chiba, S., Coady, Y., Oriol, M. (eds.) Proceedings of ECOOP 2008 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2008), Paphos, Cyprus (2008)
4. Pukall, M.: Object roles and runtime adaptation in java. In: Cazzola, W., Chiba, S., Coady, Y., Oriol, M. (eds.) Proceedings of ECOOP 2008 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2008), Paphos, Cyprus (2008)
5. Götz, S., Savga, I.: Oexploring role based adaptation. In: Cazzola, W., Chiba, S., Coady, Y., Oriol, M. (eds.) Proceedings of ECOOP 2008 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2008), Paphos, Cyprus (2008)
6. Cech Previtali, S., Gross, T.: Annotations for seamless aspect based software evolution. In: Cazzola, W., Chiba, S., Coady, Y., Oriol, M. (eds.) Proceedings of ECOOP 2008 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2008), Paphos, Cyprus (2008)
7. Sakurai, K., Masuhara, H.: Test-based pointcuts for robust and fine-grained join point specification. In: D'Hondt, T. (ed.) AOSD, pp. 96–107. ACM, New York (2008)

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
Arcelli, Francesca	University of Milano Bicocca	Italy	arcelli@disco.unimib.it
Bierman, Gavin	Microsoft Research	The Netherlands	gmb@microsoft.com
Cazzola, Walter	Università degli Studi di Milano	Italy	cazzola@dico.unimi.it
Cech Previtali, Susanne	ETH Zürich	Switzerland	scech@inf.ethz.ch
Chiba, Shigeru	Tokyo Institute of Technology	Japan	chiba@is.titech.ac.jp
de Roo, Arjan	University of Twente	The Netherlands	a.j.deroo@ewi.utwente.nl
Figueiredo, Eduardo	Lancaster University	UK	e.figueiredo@lancaster.ac.uk
Götz, Sebastian	Technical University of Dresden	Germany	sebastian.goetz@mail.inf.tu-dresden.de
Guerra, Eduardo	ITA - Brazil	Brazil	guerra@ita.br
Havinga, Wilke	University of Twente	The Netherlands	havingaw@ewi.utwente.nl
Herrmann, Stephan	Technical University of Berlin	Germany	stephan@cs.tu-berlin.de
Kakousis, Constantinos	University of Cyprus	Cyprus	kakousis@os.ucy.ac.cy
Masuhara, Hidehiko	University of Tokyo	Japan	masuhara@graco.c.u-tokyo.ac.jp
Oriol, Manuel	ETH Zürich	Switzerland	moriol@inf.ethz.ch
Ostrowski, Krzystof	Cornell University	USA	krzys@cs.cornell.edu
Pukall, Mario	University of Magdeburg	Germany	pukall@iti.cs.uni-magdeburg.de
Sanen, Frans	KULeuven	Belgium	frans.sanen@cs.kuleuven.be
Vandemonde, Yves	KULeuven	Belgium	yves.vandemonde@cs.kuleuven.be