2nd Workshop on Assessment of Contemporary Modularization Techniques (ACoM 2008)

Phil Greenwood and Alessandro Garcia
Lancaster University, UK
{greenwop, garciaa}@comp.lancs.ac.uk

Elisa Baniassad
The Chinese University of Hong Kong. China
elisa@cse.cuhk.edu.hk

Kevin Sullivan
University of Virginia, USA
sullivan@cs.virginia.edu

Yuanfang Cai
Drexel University, USA
yfcai@cs.drexel.edu

Alan MacCormack
Harvard Business School, USA
amaccormack@hbs.edu

Abstract

A variety of new modularization techniques is emerging to cope with the challenges of contemporary software engineering, such as Aspect-Oriented Software Development (AOSD), Feature-Oriented Programming (FOP), and the like. The effective assessment of such technologies plays a pivotal role in (i) understanding their costs and benefits when compared to conventional development techniques, and (ii) their effective transfer to mainstream software development.

The goal of the 2nd ACoM workshop is to put together researchers and practitioners with different backgrounds to (a) understand the impact of contemporary modularization techniques in practice; (b) explore new, and potentially more effective, modularity modeling and assessment methods to account for and guide the application of modularization techniques, and (c) discuss the potential of using modularity assessment results to improve software development outcomes, to improve existing modularization techniques, and to foster the development of new techniques.

Categories and Subject Descriptors D.2.8 [*Software Engineering*]: Metrics; D.1.0 [*Programming Techniques*]: General

General Terms Design, Experimentation, Measurement

Keywords adoption, assessment, metrics, modularity

OOPSLA'08, October 19–23, 2008, Nashville, Tennessee, USA.
ACM 978-1-60558-220-7/08/10.

1. Main Theme

Numerous modularization techniques are emerging to cope with the challenges of contemporary software engineering, such as Aspect-Oriented Software Development (AOSD) and Feature-Oriented Programming (FOP). However, it remains unclear to what extent these advanced modularization techniques have been adopted in practice, and what impact they have had on software productivity and quality. This workshop emphasizes the need for improving the assessment of these modern techniques and, as a result, accelerating their maturity and evolution in a well-informed fashion. We also aim at understanding their current impact in real industrial settings, fostering their principled adoption through effective assessment methods, and boosting the innovation of both new modularization and assessment techniques.

We seek for assessment techniques that can lead to principled guidelines to facilitate the comparison, reconciliation, and synthesis of these techniques in practice. These techniques differ in terms of supported abstractions and composition mechanisms, but they follow some common underlying principles, such as information hiding. While there is some evidence that conventional modularization techniques are overly constrained, some of the emerging techniques have been criticized for promoting non-modular solutions. One approach may be optimal in some circumstances, but not in others [1]. In many cases, these techniques are complementary to each other, and their combinations can best improve software quality.

Therefore, without effective assessment techniques, it is not clear how to maximize the benefits of contemporary modularity mechanisms. Contemporary modularization techniques call for advanced assessment techniques. It is not clear whether traditional source-based coupling-

and-cohesion assessment methods are sufficient to measure new modularization techniques, such as Web-based implicit invocation or dynamic weaving. Modern programming languages and modeling techniques (e.g. UML) are being enhanced with additional modularity mechanisms and abstractions, such as aspects, features, and the like. There is a pressing need to define proper assessment mechanisms, techniques and methods for these new modularization techniques.

We also solicit innovative ideas regarding assessment-based improvement of modularization techniques. Assessing modularization techniques will reveal their benefits and drawbacks, and may reveal the need for advances in programming languages, identification of contemporary architecture styles, or the novel combination of existing techniques. In particular, empirical studies along with supporting assessment techniques provide the basic means to improve our understanding of the benefits and drawbacks of new software decomposition techniques, especially when compared to techniques from other development paradigms.

2. Relevance and Timeliness

Since new modularization techniques, such as AOSD and FOP, are starting to reach some level of maturity, assessment is becoming a central issue to researchers and industrial practitioners. The relevance of the topic becomes even more evident when we look at the number of qualitative and quantitative case studies [2–11], software metrics [4, 12, 13], and assessment frameworks [14–16] involving new modularity techniques. They are consistently appearing in the software engineering literature. The 1st ACoM workshop was organized in May 2007 [17] as a first initiative to put together researchers and practitioners in order to discuss the multi-faceted issues that emerge in the assessment and/or comparison of new modularization techniques. The theme of the 2nd ACoM workshop[1] extends the first edition and intends to stimulate discussions on important open questions, including:

1. How do new modularization techniques affect working practices and help with software development and evolution? What guidelines can be established from assessment results to improve working practices?
2. What is the impact of using conventional quantitative metrics to assess software modularity? Are they effective enough to assess contemporary modularity techniques? How can we validate assessment mechanisms?
3. To what extent does assessment depend on extensive experience in practice? How can observations of practitioners help in assessment?
4. What are the potential paths leading to improved/new, and more effective, modularization techniques?
5. How can we compare these modularization techniques, reconcile their seemingly different appearance, and synthesize their applications to design software more effectively?
6. What are the fundamental weaknesses of traditional modularization techniques that affect software productivity and quality? What are the tradeoffs from using contemporary techniques to address these weaknesses?

[1] ACoM.08 Website: http://www.comp.lancs.ac.uk/computing/ACoM.08/

3. Workshop Goals

In a nutshell, the workshop goals are the following: (i) solicit experience reports from practitioners that help to better understand the impact of modularity assessment in practice and how contemporary modularization techniques have affected software development; (ii) stimulate innovative ideas of new and more effective modularity assessment methods which incorporate emerging modularization techniques; (iii) discuss the potential impact of assessment techniques, in terms of leading to improvement in the application of contemporary techniques in practice [18], and to new modularization techniques; (iv) improve our understanding on such issues as how to effectively differentiate their applicability in different circumstances, how to combine these techniques in practice; and (v) foster a collaborative environment for both practitioners and researchers interested in the effective assessment, principled application, and innovative improvement of new development techniques. Our long term goals include the standardization of software modularity assessment.

4. Activities and Format

We plan a one-day workshop. The first session in the morning will include a restricted number of short presentations, followed by discussions and round-tables. To ensure the focus centers on industrial application of assessment techniques, an invited talk will be given by a prominent industrial speaker. A number of discussion groups will be formed in the afternoon session that will discuss issues raised by the papers presented in the morning session. The exact discussion topics will be determined by the papers presented and also the interests of the workshop participants. However, certain generic questions/topics that could be covered include:

- How do contemporary modularization techniques affect dependencies and how these different dependencies can be compared.
- In what way are contemporary modularization techniques used in industry and how can they be assessed.
- How the modularity properties of software artifacts affect their evolution.
- What are the exact attributes associated with modularity that need to be assessed and how to define these attributes.

- How to assess contemporary modularization techniques in terms of the heterogeneity involved in large-systems that cross network, organizational and even cultural boundaries.
- How to advance the state-of-the-art and improve existing modularization techniques and their application.

The outcome of these discussions will be reported back to all participants through short presentations by each discussion group.

5. Programme Committee

- Mehmet Aksit, University of Twente, The Netherlands
- Sven Apel, University of Passau, Germany
- Paulo Borba, Federal University of Pernambuco, Brazil
- Yvonne Coady, University of Victoria, Canada
- Marc Eaddy, Columbia University, USA
- Eduardo Figueiredo, Lancaster University, UK
- Rachel Harrison, Stratton Edge Consulting, UK
- George Heineman, Worcester Polytechnic Institute, USA
- James D. Herbsleb, Carnegie Mellon University, USA
- Arno Jacobsen, University of Toronto, Canada
- Gregor Kiczales, University of British Columbia, Canada
- Christa Schwanninger, Siemens AG, Germany
- Peri Tarr, IBM Watson Research Center, USA
- Robert Walker, University of Calgary, Canada

6. Post-Workshop Activities

A workshop report (in the form of a technical report) will be produced that will document the proceedings and discussions of the workshop. This report will then be disseminated through appropriate channels and to the workshop participants. However, the workshop aims to go beyond this by creating a community that will conduct studies in collaboration. Often effort is duplicated throughout a number of studies, creating this community could reduce this duplication by utilizing shared resources. Such an initiative has already been undertaken by a number of institutions through the creation of a Testbed for Aspect-Oriented Software Development (TAO) [19]. However, such an initiative requires expansion to include other contemporary modularization techniques and increase the data set. Furthermore, by bringing academics and industrial personnel together to conduct such studies, mutual benefits can be experienced by each partner. For example, academics can utilize industrial case-studies and experiences, and industrial personnel can utilize cutting-edge assessment techniques.

References

[1] K. Sullivan, L. Gu, and Y. Cai. Non-modularity in aspect-oriented languages: integration as a crosscutting concern for aspectj. In *AOSD '02: Proceedings of the 1st international conference on Aspect-oriented software development*, pages 19–26, Enschede, The Netherlands, 2002.

[2] J. Hannemann and G. Kiczales. Design pattern implementation in java and aspectj. In *Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 161–173, Seattle, Washington, USA, 2002.

[3] A. Garcia, C. Sant'Anna, E. Figueiredo, U. Kuleska, C. Lucena, and A. Von Staa. Modularizing design patterns with aspects: A quantative study. In *4th International Conference on Aspect-Oriented Software Development (AOSD)*, pages 3–14, Chicago, USA, 2005.

[4] M. Ceccato and P. Tonella. Measuring the effects of software aspectization. In *1st Workshop on Aspect Reverse Engineering*, The Netherlands, 2004.

[5] J. Kienzle and R. Guerraoui. Aop: Does it make sense? the case of concurrency and failures. In *ECOOP '02: Proceedings of the 16th European Conference on Object-Oriented Programming*, pages 37–61, London, UK, 2002. Springer-Verlag.

[6] S. Soares, E. Laureano, and P. Borba. Implementing distribution and persistence aspects with aspectj. In *OOPSLA '02: Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 174–190, Seattle, Washington, USA, 2002.

[7] S.Apel, T. Leich, and G. Saake. Aspectual mixin layers: aspects and features in concert. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, pages 122–131, Shanghai, China, 2006.

[8] E. Figueiredo, N. Cacho, C. Sant' Anna, M. Monteiro, and A. Garcia et al. Evolving software product lines with aspects: An empirical study on design stability. In *30th International Conference on Software Engineering (ICSE08)*, Leipzig, Germany, 2008.

[9] R. Coelho, A. Rashid, A. Garcia, F. C. Ferrari, N. C., U. Kulesza, A. von Staa, and C. J. P. de Lucena. Assessing the impact of aspects on exception flows: An exploratory study. In J. Vitek, editor, *ECOOP*, volume 5142 of *Lecture Notes in Computer Science*, pages 207–234. Springer, 2008.

[10] P. Greenwood, T. T. Bartolomei, E.o Figueiredo, M. Dósea, A. F. Garcia, N. Cacho, C. Sant'Anna, S. Soares, P. Borba, U. Kulesza, and A. Rashid. On the impact of aspectual decompositions on design stability: An empirical study. In E. Ernst, editor, *ECOOP*, volume 4609 of *Lecture Notes in Computer Science*, pages 176–200. Springer, 2007.

[11] G. C. Murphy, R. J. Walker, E. Baniassad, M. P. Robillard, A. Lai, and M. Kersten. Does aspect-oriented programming work? *Communications of the ACM*, 44(10), 2001.

[12] J. Zhao and B. Xu. Measuring aspect cohesion. In *Proc. Fundamental Approaches to Software Engineering (FASE'2004.)*, 2004.

[13] A. Zakaria and D. Hosny. Metrics for aspect-oriented design. In *Workshop on Aspect-Oriented Modeling*, San Francisco, USA, 2003.

[14] C. V. Lopes and S. K. Bajracharya. An analysis of modularity in aspect oriented design. In *AOSD '05: Proceedings of the 4th international conference on Aspect-oriented software development*, pages 15–26, Chicago, Illinois, 2005.

[15] E. L. A. Baniassad, G. C. Murphy, C. Schwanninger, and M. Kircher. Managing crosscutting concerns during software evolution tasks: an inquisitive study. In *AOSD '02: Proceedings of the 1st international conference on Aspect-oriented software development*, pages 120–126, Enschede, The Netherlands, 2002.

[16] Y. Cai, S. Huynh, and T. Xie. A framework and tool supports for testing modularity of software design. In *ASE '07: Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*, pages 441–444, Atlanta, Georgia, USA, 2007.

[17] A. Garcia, P. Greenwood, G. Heineman, R. Walker, Y. Cai, H. Yang, E. Baniassad, C. Schwanninger C. Lopes, and J. Zhao. Assessment of contemporary modularization techniques - acom'07: Workshop report. In *ACoM 07*, volume 32, pages 31–37, Minneapolis, USA, 2007. SIGSOFT Software Engineering Notes.

[18] A. MacCormack, C. Kemerer, M. Cusumano, and B. Crandall. Trade-offs between productivity and quality in selecting software development practices. *IEEE Softw.*, 20(5):78–85, 2003. 942702.

[19] http://www.comp.lancs.ac.uk/ greenwop/tao/. Tao a testbed for aspect oriented software development, 2008.