

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

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 are 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) determining their costs and benefits when compared to conventional development techniques, and (ii) encouraging their transfer to mainstream software development. The purpose of this report is to summarize the findings of the 2nd International Workshop on the Assessment of Contemporary Modularization Techniques (ACoM.08), co-located with the 23rd Annual ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2008). The main purpose of the workshop was to bring together researchers and practitioners with different backgrounds to (a) understand the impact of contemporary modularization techniques in practice; (b) explore new, and more effective, assessment techniques to 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.

The workshop was opened by an invited talk from an industrial practitioner who has experience of using con-

temporary modularization techniques in an industrial setting and has an awareness of the importance of their systematic assessment. The rest of the workshop was organised into four technical sessions that was concluded by an open-discussion session involving all workshop participants. During the course of the workshop, the participants discussed a number of important issues that related to previous, ongoing and future issues surrounding contemporary modularity assessment techniques. The ACoM.08 website, including all the accepted papers, presentations and the electronic version of this report can be found at <http://www.comp.lancs.ac.uk/ACoM.08/>.

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

1. Introduction

Numerous modularization techniques are emerging to cope with the challenges of contemporary software engineering, such as Aspect-Oriented Software Development (AOSD) [1] and Feature-Oriented Programming (FOP) [2]. However, it remains unclear to what extent these advanced modularization techniques have been adopted in practice. It is even less clear the impact they have had on software productivity and quality. There is a need for improving the assessment of these techniques and, as a consequence, accelerate their maturity and evolution in a well-informed fashion. Furthermore, it is necessary to understand their current impact in industrial settings, to encourage their adoption through effective assessment methods, and boosting the innovation of both new modularization and assessment techniques.

Effective assessment techniques are fundamental for the derivation of guidelines that support 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 [3]. 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 unclear how to maximize the benefits of contemporary modularity mechanisms. It is also unclear whether traditional coupling-and-cohesion assessment methods are sufficient to measure new modularization techniques. There a number of characteristics intrinsic to new modularity techniques - such as implicit invocation, quantification mechanisms and dynamic weaving - that probably require the quantification of other equally-important software internal attributes. 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 tailored to these new modularization techniques.

Assessment mechanisms of contemporary modularity techniques can also guide their development and subsequent improvement. 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.

As 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 [4–13], software metrics [6, 14, 15], and assessment frameworks [16–18] involving new modularity techniques. They are consistently appearing in the software engineering literature.

The 1st ACoM workshop was organized in May 2007 [19] 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 work-

shop¹ 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 trade-offs from using contemporary techniques to address these weaknesses?

This 2nd ACoM workshop extends the previous by investigating the needs of assessment techniques in industrial settings. This theme of the 2nd workshop was highlighted in the call for papers and shaped the topic of the invited talk. In addition, such a theme led us to decide to host the workshop at OOPSLA, a conference renowned for the industrial participation.

The workshop elicited two categories of submissions on the assessment of modularity techniques: (i) traditional position papers (up to 6 pages) related to the workshop topics, and (ii) very short position statements (1-2 pages), where the authors described their “innovative thoughts”, lessons learned, or points of view with respect to one or more of the workshop topics. Long position papers described work that is not yet advanced enough for a full conference paper. They were expected to have a more solid idea but not necessarily grounded on a sound evaluation. Short position statements were mainly reviewed for their relevance towards the workshop topics, including some feedback from reviewers.

2. Workshop Proceedings and Programme Committee

The accepted papers were compiled into the ACoM.08 workshop proceedings which was published as a technical report [20]. An electronic copy of the proceedings was made available to the participants prior to the workshop. The Pro-

¹ ACoM.08 Website: <http://www.comp.lancs.ac.uk/computing/ACoM.08/>

gramme Committee was composed of the following members:

- 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

3. Workshop Organization and Structure

The organization of the workshop was in charge of the organizing committee members: Phil Greenwood (Lancaster University - UK), Alessandro Garcia (Lancaster University - UK), Elisa Baniassad (The Chinese University of Hong Kong - China), Kevin Sullivan (University of Virginia - USA), Yuanfang Cai (Drexel University - USA), Alan MacCormack (Harvard Business School - USA), Eduardo Figueiredo (Lancaster University - UK) and with the assistance of the Programme Committee.

Submissions were received from around the globe with the majority submitted from North America, South America and Europe. The submitted papers were reviewed and modified prior to the workshop. In total, twelve papers were selected for presentation at the workshop. Each paper was reviewed by at least three members of the PC; the final selection of the papers to be presented was made by the workshop organizers based on the reviewers provided by the PC. The accepted papers were selected because they were seen to offer new perspectives on the workshop topics and they had the potential for stimulating discussion between the workshop participants.

The workshop attracted 22 participants who were all interested in discussing ideas on empirical studies and diverse approaches to assessing contemporary modularization techniques. Prior to the workshop, the participants were asked to read all of the accepted papers in order to facilitate discussion and ensure interesting questions were raised.

The workshop was structured into the following sessions:

- A brief welcome presentation was given by Phil Greenwood who provided an overview of the various sessions

of the day and gave an introduction to the workshop's topics and goals.

- The first presentation of the day was the invited talk from Thomas Cottenier of Hengsoft LLC. The title of Thomas' talk was "Aspect-Oriented Software Development and the Division of Labour in Industrial Software Development Projects".
- Four technical sessions followed which grouped accepted papers according to common themes. The papers presented in first session examined contemporary metrics for assessing modularity. The second session focused on the role assessment plays in the software evolution process. The third session contained papers that described experiences from conducting empirical studies. Finally, the fourth session was dedicated to papers that documented a variety of taxonomies that support software evaluation. At the end of each presentation or session, time was reserved for discussion. A session chair was appointed for each of the four technical sessions to coordinate the discussions. The various discussion topics which were raised throughout the day are summarized in Section 6.
- The final part of the workshop was reserved for open discussion between participants. Rather than dividing into separate discussion groups it was decided that it would be more effective to have an 'open-table' discussion where any workshop participant can raise any discussion point that is relevant to the workshop topics. The outcomes of these discussions are briefly summarized in Section 6.5.

4. Workshop Presentations

As mentioned above, twelve papers were accepted for presentation with three papers presented in each of the four sessions. The first two sessions contained only short position papers, speakers in these two sessions were given 10 minutes. To make the most efficient use of time in these two sessions, questions and topics for discussion were raised at the end of each session rather than after each presentation. For the remaining two sessions, which consisted entirely of papers in the longer format, speakers were given 15 minutes for their presentation and 5 minutes for questions/discussion. The papers and their authors were as follows; summaries of these presentations are presented in the following sections of this report:

- *Measuring Software Design Modularity*; Yuanfang Cai, Sunny Huynh (Drexel University - USA)
- *Towards Probabilistic Assessment of Modularity*; Kevin Hoffman, Patrick Eugster (Purdue University - USA)
- *Measuring Design Volatility Against Design Rule Stability*; Yuanfang Cai, Kanwarpreet Sethi (Drexel University - USA)

- *Assessing Modularity of Feature Models with ACNs*; Kanwarpreet Sethi, Sunny Huynh, Yuanfang Cai (Drexel University - USA)
- *On the Assessment of Pointcut Design in Evolving Aspect-Oriented Software*; Raffi Khatchadourian, Phil Greenwood, Awais Rashid (Ohio State University - USA, Lancaster University - UK)
- *Assessing the Malleability of Modular Design*; Giuseppe Valetto (Drexel University - USA)
- *Using Meta-Data in Aspect-Oriented Frameworks*; Eduardo M. Guerra, Jefferson O. Silva, Fabio F. Silveira, Clovis T. Fernandes (ITA - Brazil, UNIFESP - Brazil)
- *Mining Software Repositories for Evaluating Software Engineering Properties of Language Designs*; Hriday Rajan (Iowa State University - USA)
- *Evaluating the Efficacy of Concern Driven Metrics: A Comparative Study*; Claudio Sant' Anna, Alessandro Garcia, Carlos J. P. Lucena (UFB - Brazil, Lancaster University - UK, PUC-Rio - Brazil)
- *A Close Look at Composition Languages*; Florian Heidenreich, Jendrik Johannes, Steffen Zschaler, Uwe Asmann (TUD - Germany, Lancaster University - UK)
- *Towards a Framework for Guiding Aspect-Oriented Software Maintenance Empirical Studies*; Marcelo Moura, Sergio Soares, Fernando Castor Filho, Mario Monteiro, Phil Greenwood, Alessandro Garcia, Elliackin Figueiredo, Diego Araujo (University of Pernambuco - Brazil, Lancaster University - UK)
- *Assessing Contemporary Modularization Techniques for Middleware Specialization*; Akshay Dabholkar, Anirudha Gokhale (Vanderbilt University - USA)

5. Invited Talk

Thomas Cottenier was invited to open the workshop by giving an invited talk on a related topic of his choosing. Thomas is a Member of the Technical Staff at Hengsoft LLC, a newly established software company with operations in the USA, Ukraine, Russia and China. Prior to joining Hengsoft, Thomas worked as a researcher in the Software and System Engineering Research Lab at Motorola where he designed an aspect-oriented extension to the UML modelling language tailored to the development of telecom infrastructure software. Over three years, Thomas has been actively involved in the design, development and deployment of custom Aspect-Oriented Software Development solutions in different business units at Motorola, including the Wireless Broadband division and the Mobile Device division. Thomas holds a Ph.D. in Computer Science from the Illinois Institute of Technology, Chicago and a MS in Telecom System Engineering from the Universit Libre de Bruxelles, Belgium. Thomas' experience of using contemporary modularization techniques and having an awareness of the impor-

tant role assessment plays in an industrial setting made him an ideal candidate for giving an invited talk at ACoM.

Thomas' talk centred upon discussing the impact of AOSD technologies on the division of labour within large software development organizations. Thomas presented a real-world case study from an industrial telecom software development project at Motorola. Thomas then presented the software development organizational structure that can be composed of hundreds of software engineers, organized into a number of development teams. Thomas then returned to Parnas' definition of the software module, in that a module is a work/responsibility assignment and not some subroutine or language element. This definition plays an important role in terms of mapping software to the physical components of the system and the allocation of responsibilities with respect to the implementation of different features the system to make the most efficient use of resources (good vs bad programmers). Thomas then went on to discuss how aspect-oriented decomposition can be applied to the system based on an analysis of the system requirements. Thomas highlighted that AOSD enables the division of labour within the organization to be structured according to more specialized, cohesive slices of behaviour and that development teams can be assigned responsibilities with respect to the implementation of more specific requirements of the system. Thomas then went on to show the benefits from applying AOSD in such circumstances and that productivity can be increased. However, a drawback from doing this is the increase of cost/effort brought about from the increase in the coordination of development tasks and the resolution of interactions between behaviours.

Discussions regarding the invited talk centred upon a variety of maintenance issues that such division of labour raises. For example, would it be possible to automate the deployment of the models generated or even possible to map the modelled concepts to components. Other topics of discussion related to how development tasks were actually assigned, it was suggested that some type of tool support may be necessary to aid this process to ensure the each task is assigned to the most appropriate developer.

6. The Sessions

The workshop consisted of four sessions of presentations and discussion. Common themes within the accepted papers were identified to create the four sessions with each paper assigned to the relevant session.

6.1 Session I: Contemporary Metrics for Assessing Modularity

The first paper presentation of this session was given by Yuanfang Cai and was entitled "Measuring Software Design Modularity". As with the invited talk, this paper considered Parnas' definition of a module being related to independent task assignment. Yuanfang pointed out that the major-

ity of modularity metrics are not applicable to this definition and so a new way of measuring modularity in terms of task assignment is necessary. Yuanfang then argued that design structure matrices (DSMs) can be used to represent a design and automatically clustered to reveal independent modules. This output can then be used to determine to what extent the design can be independently implemented or changed. A metric called level of independence was proposed to measure this attribute.

The second presentation, given by Kevin Hoffman, was titled “Towards Probabilistic Assessment of Modularity”. Kevin began his presentation by highlighting a series of questions which developers may want answering with regard the modularity of a program. These questions included: Which program elements are the most depended upon? Which program elements are the most likely to change? (i.e. the most fragile?) Which program elements are the most likely to be affected by rippled effects given a set of changes? Which program elements were most influential during some execution of a test case? Kevin argued that Assessment Graphs created from weighted DSMs can be used to answer such questions. Stationary distribution of the Markov chain extracted from the assessment graph creates the appropriate metrics. Proposed metrics include: dependency propagation ranking, impact propagation ranking and impact shift ranking.

The final paper of this session, “Measuring Design Volatility Against Design Rule Stability”, was again presented by Yuanfang Cai. Firstly, Yuanfang gave an overview of the various aspects which are important to consider when proposing metrics to assess design volatility. These points included: the number of modules, the size of each module, the likelihood that a module will change and how other modules depend on the module. Yuanfang argued that given the extensive use of DSMs in modularity assessment that volatility metrics should be developed based on them. Yuanfang described three metrics to measure stability based on DSMs which included: decision volatility, design rule volatility, and design volatility. Yuanfang then went on to describe how these metrics could be calculated.

Given the prominence DSMs had in the three papers presented in this session, the discussion naturally focused on DSMs. Topics of discussion mostly related on the creation of the DSMs, including how elements of the DSMs can be identified and how to select an appropriate level of detail. Discussion then went on to management of the DSMs including how DSMs can be compared and how to handle traceability when employing DSMs.

6.2 Session II: Assessment in Software Evaluation

The second session began with Yuanfang Cai presenting a paper titled “Assessing Modularity of Feature Models with ACNs”. In this presentation Yuanfang highlighted a problem of model-driven development (MDD) which causes the modularity of generated code to be lost with various impor-

tant design decisions becoming implicit in the code. In order to allow these design decisions to be externalised and allow the module structure to be visualized, the authors propose to use Augmented Constraint Networks (ACNs). ACNs contain variables which allow design decisions and constraints representing dependencies to be represented which can be created during the transformation processes employed in MDD. From an ACN a DSM can be automatically derived which can be used to assess modularity.

Raffi Khatchadourian then presented his paper “On The Assessment of Pointcut Design in Evolving Aspect-Oriented Software”. Raffi first gave a brief introduction to the concepts used in AOSD and then highlighted some of problems associated with using AOSD, with emphasis on the difficulty of creating optimal pointcuts that capture the developer’s true intentions and the subsequent pointcut fragility that occurs. Raffi proposed a mechanism that can derive and measure a pointcut’s ability to capture the developer’s intention. This approach involves constructing a concern graph for the join points which a pointcut advises and then measuring how representative that graph is of other join points which the same pointcut advises.

The final paper of this session, “Assessing the Malleability of Modular Design” was presented by Giuseppe Valetto. Giuseppe stated that one of the main purposes of modularization is to allow the creation of a malleable design. He claimed a good design should be able withstand the “injury of time” and “bend but not break”. Giuseppe raised the question of whether it is possible to measure the malleability of a design to be able to make informed design decisions. Giuseppe then went on to define what he thought malleability was in terms of system and design change using relative code churn and modifications to design decisions. Giuseppe also described an experiment using MobileMedia which allowed the malleability of an OO and AO design to be compared.

Discussion following these presentations focused on technical aspects associated with Raffi’s presentation regarding how the concern graphs can be derived accurately and the use of these in other scenarios.

6.3 Session III: Experience in Empirical Studies

The third sessions begun with a presentation from Eduardo M. Guerra who presented his paper entitled “Using Meta-Data in Aspect Oriented Frameworks”. Eduardo started his presentation with an example of what can happen when a cross-cutting concern, such as logging, needs varying depending on the location of where it is applied. This variation causes “advice explosion” where the number of advice which is needed to implement all possible variations increases exponentially. Eduardo proposes to use meta-data instead to limit this problem and provide various other advantages including: eliminate syntactic coupling, provide extensibility on variabilities, reduce the number of advice and simplify the pointcut management.

Discussion and questions on this presentation centred upon the applicability of the solution. One particular question raised the possibility of using meta-data in this manner simply shifts the problem to elsewhere in the program and also increases the scattering and tangling of the concern which the framework is target. Another question raised related to the observance of the problem in real case-studies and whether the problem was unique to logging. It was mentioned that such a problem exists in JBoss with transactions particularly susceptible to the problem.

The next presentation of this session was given by Hridesh Rajan who spoke about his paper entitled “Mining Software Repositories for Evaluating Software Engineering Properties of Language Designs”. In his paper, Hridesh attempts to provide a mechanism for studying the features of programming languages. This mechanism is based on utilising the rich version control history available. Hridesh relates this to his experience of developing Ptolemy, a language which combines best ideas of II and AO. Hridesh explained that version history contains real changes which enables the evolution properties of languages to be accurately assessed.

Several key and difficult questions of utilising version history in this manner were highlighted by Hridesh during his presentation but others were also raised by other participants, including the potential for vital information to be missing from the version history which may make it difficult to replay the changes accurately. Furthermore, questions were raised with regard who is responsible for describing the change which will be replayed.

The final paper in this session, “Evaluating the Efficacy of Concern-Driven Metrics: A Comparative Study”, was presented by Phil Greenwood due to the paper’s author, Claudio Sant’Anna, unable to attend the workshop. The paper outlines the need for new metrics to be applied beyond conventional metrics to accurately assess the attributes of contemporary modularization techniques. Conventional metrics typically evaluate the attributes of modules rather than the attributes of the concerns, which are primary focus of contemporary modularization techniques. The paper highlights a study conducted that compares the ability of conventional and concern based metrics to identify common modularity-related design flaws and concludes that concern-driven metrics are useful vehicle for assessing design modularity.

One notable question asked what the study actually assessed about the metrics and whether it was assessing the intuitiveness of the metrics rather than their effectiveness. Phil agreed with this thought but added that intuitiveness is also an important property of the metrics due to the large number of metrics available and picking the right one to identify the relevant design flaws. Having intuitive metrics is a desirable characteristic and so is a valid property to assess.

6.4 Session IV: Taxonomies for Software Evaluation

The final session began with a presentation from Steffen Zschaler who presented the paper “A Close Look at Compo-

sition Languages”. Steffen began his presentation by highlighting the problem faced by researchers when they wish to compare different languages. Typically each language will utilise their own terminology making comparison difficult. Steffen and the authors of this paper claim that unifying terminology is necessary to allow these comparisons to be made. Steffen went on to describe commonalities of composition languages and presented a vocabulary for their comparison.

Some questions were raised about the mapping and interpretation of the relationships between composition languages. For example, Steffen claimed that advice could be mapped to a component in component-based composition. However, this mapping was questioned and it was stated that the aspect should be the component and advice be the operations of the component. Steffen agreed that this is indeed a valid interpretation and open to debate.

This presentation was followed by Sergio Soares presenting the paper “Towards a Framework for Guiding Aspect-Oriented Software Maintenance Empirical Studies”. Sergio highlighted the current challenges facing researchers when conducting empirical studies involving AOSD. These challenges centre upon the lack of systematic evaluation of various properties that make up an empirical study. Sergio and his colleagues propose a framework that will aid researchers and practitioners when developing AO software maintenance studies. Sergio went on to describe the process of how the framework will influence the design of a study. In order to demonstrate the benefits of such a framework Sergio described a comparison performed between the planning of two experiments conducted, one designed by a postgraduate student with the aid of the framework and another designed by an expert without using the framework. Interestingly, the experiment designed with the use of the framework made considerations not addressed by the expert.

The final paper, “Assessing Contemporary Modularization Techniques for Middleware Specialization”, was presented by Akshay Dabholkar. Akshay first highlighted the increasing need for specialization in middleware systems given that they often need to: minimize its footprint, maximize throughput, handle real-time requirements and adapt to unpredictable runtime changes. Specializing middleware raises a number of challenges which include: when specialization needs to be applied, specializing different elements, and how the specialization should be applied. In order to address these challenges Akshay presented a taxonomy that would allow developers to select an appropriate technology to address the what, when and how challenges.

One question raised regarding this work was the applicability of the taxonomy in domains other than middleware and whether it can be used to guide specialization of other applications.

6.5 Discussions

Rather than breaking up into separate discussion groups it was decided to have a round-table discussion whereby any participant can raise any topic for discussion to make more efficient use of time. Firstly, the remaining twelve participants gave a brief introduction to their interests and background. There was a mixture of people from academia and industry and, therefore, views from both researchers and practitioners were collected and discussed. This was particularly attractive given the aims of the workshop.

Interesting insights were gained from the industrial participants regarding the role of refactoring using innovative modularity mechanisms in commercial software. Managers often need convincing that refactoring using these new techniques is a valuable part of the software development process and is worth committing effort to perform. As refactoring does not produce any tangible improvement of the software (i.e. new feature added) it is often viewed by managers as a non-essential task. It was suggested that software assessment could be a mechanism to demonstrate the benefits of refactoring to managers. However, these metrics must be in a form which are easily understood by managers and can improve the productivity of the developers.

This led on to a discussion regarding what the current role of metrics have in an industrial setting. It was agreed that one of their most prominent roles was in terms of task allocation so that the most complex tasks were assigned to the most experienced developers, This view was consistent with Thomas' insights presented at the start of the workshop. However, this led to a comment that there is often a struggle in industry in terms of things that are good for developers and things that are good for managers. Often practices, such as assessment, refactoring and innovative modularity techniques, are often overlooked by managers although they are beneficial to developers. Managers need to consider the long term benefits of their developers and introduce initiatives, such as assessment and new development techniques, to improve the quality of the software. In addition, to improving the quality of the software it was viewed by the workshop participants that an added benefit of performing assessment and applying metrics is that it gets developers to communicate and share ideas which can also contribute to the improvement of the software.

The discussion then went on to how researchers and practitioners could work together in order to conduct empirical studies that assess contemporary modularity techniques. It was highlighted that problems are often encountered when trying to use industrial applications within studies due to often incomplete documentation being provided (due to intellectual property rights issues etc.). It was mentioned that using open-source software is one possibility (as suggested in Hridayesh Rajan's presentation). However, this itself raises possible issues such as how is in guaranteed with all the con-

tributors that a modular design is maintained and how can you be sure a stable release is available.

7. Conclusion

The purpose of this workshop was to discuss the various issues surrounding the assessments of contemporary modularization techniques with an emphasis on their role in industrial settings. The workshop was a large success due to the high quality of papers submitted and the level of participation of the attendees. ACoM.08 achieved its goals and provided a forum for researchers and practitioners to discuss issues related to evaluating contemporary modularization approaches. The outcome from the discussion session highlighted a variety of outstanding issues that the community must address to bridge the gap between an academic and industrial setting.

The workshop achieved the following aims:

1. Debate the open issues on the assessment of contemporary modularization techniques.
2. Bring together researchers and practitioners to highlight the importance of rigorous evaluation of emerging modularization techniques in both an academic and industrial environment.
3. Motivate the expansion of research and practice associated with assessment of emerging modularization technologies.
4. Foster a collaborative environment for both practitioners and researchers interested in effective assessment of new development techniques.

All the position papers and presentation slides are available at the workshop website. All the results obtained by the discussions were summarized and made electronically available at the workshop website. The aim is to highlight issues that shall become part of the forthcoming research agenda.

Acknowledgements

The organizers would like to thank all those who contributed with submissions to the workshop and the program committee members who invested their time on reviewing such submissions. We would sincerely like to thank again all the ACoM.08 participants for their active involvement in the meeting and the level of their contributions to the discussion session.

References

- [1] Robert E. Filman, Tzilla Elrad, Siobhán Clarke, and Mehmet Akşit, editors. *Aspect-Oriented Software Development*. Addison-Wesley, Boston, 2005.
- [2] Don Batory, Jacob Neal Sarvela, and Axel Rauschmayer. Scaling step-wise refinement. In *ICSE '03: Proceedings of the 25th International Conference on Software Engineering*,

- pages 187–197, Washington, DC, USA, 2003. IEEE Computer Society.
- [3] 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.
- [4] 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.
- [5] A. Garcia, C. Sant’Anna, E. Figueiredo, U. Kuleska, C. Lucena, and A. Von Staa. Modularizing design patterns with aspects: A quantitative study. In *4th International Conference on Aspect-Oriented Software Development (AOSD)*, pages 3–14, Chicago, USA, 2005.
- [6] M. Ceccato and P. Tonella. Measuring the effects of software aspectization. In *1st Workshop on Aspect Reverse Engineering*, The Netherlands, 2004.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] P. Greenwood, T. T. Bartolomei, E. 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.
- [13] 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.
- [14] J. Zhao and B. Xu. Measuring aspect cohesion. In *Proc. Fundamental Approaches to Software Engineering (FASE'2004.)*, 2004.
- [15] A. Zakaria and D. Hosny. Metrics for aspect-oriented design. In *Workshop on Aspect-Oriented Modeling*, San Francisco, USA, 2003.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] Phil Greenwood, Alessandro Garcia, Yuanfang Cai, Kevin Sullivan, Elisa Baniassad, and Alan. International workshop on assessment of contemporary modularization techniques. Number COMP-004-2008. Lancaster University, October 2008.