

Editorial: Special Section on Early Aspects

Ruzanna Chitchyan
Department of Computer Science
Lancaster University
Lancaster, UK
rouza@comp.lancs.ac.uk

Steffen Zschaler
Department of Computer Science
Lancaster University
Lancaster, UK
szschaler@acm.org

As software is becoming larger and ever more complex, new Software Engineering approaches addressing these complexities arise—for example, Model Driven Development, Software Product Lines Engineering, and Aspect-Oriented Software Development (AOSD) [1, 4]. AOSD has in its core the principle of separation of concerns [3] which aims to simplify system development by allowing the developers to focus on one matter of interest (or concern) at a time. In particular, AOSD advocates the idea that single hierarchical structures (e.g., classes in code or models, or use cases / viewpoints in requirements) are too limiting to effectively separate all concerns in complex systems [1]. Thus, additional structures (termed aspects) are useful for better separation of concerns which would otherwise be scattered and tangled across the single hierarchical structures.

Work on Early Aspects, as exemplified by papers presented in this special section, studies the issues related to identification, separation, modelling, analysis, and composition of aspects in requirements and architectural artefacts. Questions considered include developing formalisms appropriate for identification and capturing early aspects; integration of different types of early-aspect models into a consistent one; conflict management in and with early aspects; relating different Early-Aspect approaches; consideration of effort required for early aspect approaches; etc. This special section showcases some solutions for the early-aspects related issues with five full papers accepted at the Early Aspects'2010 Workshop [2]. These papers can be grouped into three themes: Aspect Identification, Conflict Management, and Relationships between Technologies.

Aspect Identification

Research in this area studies how aspects can be first identified in software requirements.

“A Method Based on Petri Nets for Identification of Aspects” by V. Abdelzad et al. uses Petri Nets to formalise requirement descriptions and relations between requirements. These can then be analysed to identify candidate crosscutting concerns, which may be further developed into aspects in an implementation.

“Towards a Domain-Oriented Approach for Identifying Aspects in Software Requirements” by E. A. Nasser et al. uses stability analysis and formal concept analysis to identify enduring and crosscutting themes in requirements. The authors argue that these themes can be used as candidate crosscutting concerns and developed into aspects.

Conflict Management

Research in this area is concerned with identifying conflicts between different early aspects of a system.

“Conflict Management in Aspect-Oriented Requirements Engineering” by A. Sardinha et al. proposes a formalisation of requirements prioritisation by a set of stakeholders and shows how optimisation techniques can be used to trade-off conflicting requirements maximising overall stakeholder satisfaction.

Relationships between Technologies

The two papers in this area discuss the relationships between different techniques for dealing with crosscutting concerns at an early stage.

“On the Role of Features and Goal Models in the Aspect-Oriented Development of Software Product Line” by L. Silva et al. discusses the relationships between feature models and goal models and how these can be exploited in the context of software product lines.

“Are Themes and Use Cases the Same?” by V. Vranic et al. compares use cases as used for modelling aspects at the requirements level and themes as used in Theme/Doc and shows how these notions are interrelated by defining transformations transforming use-case models into Theme/Doc models and vice versa.

The articles presented here represent a subset of the work currently undertaken on Early Aspects. We hope that the present publication, as well as the continuing series of Early Aspects workshops will foster further work and discussion on this topic.

Finally, we would like to thank all of the authors and Early Aspects 2010 workshop PC who have helped in selection of papers and preparation of the workshop and of this special section. In addition, we would like to thank Valentino Vranic for his help in making this special section possible.

References

- [1] Aspect-Oriented Software Association. *aosd.net*. Published on-line at <http://aosd.net/>, 2010.
- [2] R. Chitchyan and S. Zschaler. Workshop on aspect-oriented requirements engineering and architecture design (early aspects 2010). Published on-line at <http://aosd-europe.net/eaAOSD2010/>, 2010.
- [3] E. W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, Englewood Cliffs, NJ, 1976.
- [4] R. E. Filman, T. Elrad, S. Clarke, and M. Akşit. *Aspect-Oriented Software Development*. Addison-Wesley Professional, 2004.