ABSTRACT
This paper summarizes the Workshop in Aspect-Oriented Requirements Engineering and Architecture Design.

Categories and Subject Descriptors

General Terms
Design, documentation, languages.

Keywords
Aspects, early aspects, requirements engineering, software architecture, crosscutting concerns.

1. INTRODUCTION
Early aspects are crosscutting concerns that exist in the early life cycle phases of software development, including the requirements engineering, domain analysis and architecture design activities. Whereas conventional aspect-oriented software development approaches are mainly concerned with identifying aspects at the programming level (thus leaving the identification and treatment of aspects to implementers), early aspects work focuses on the impact of crosscutting concerns at the early phases of the software development. Identifying and managing early aspects across phases can:

- help ensure that cross-cutting concerns evident in a system’s problem domain or solution space are captured as aspects in the implementation.
- increase consistency of requirements and architecture with each other and with implementation;
- provide a way for people with system-wide responsibility and scope (such as architects and domain experts) to identify aspects
- provide rationale and traceability for aspects across lifecycle activities.

This is the ninth Early Aspects (EA) workshop, and the first at ICSE. Others have been held at AOSD, OOPSLA, and SPLC. The goal of the overall workshop series is to articulate, develop, and codify best practices for working with aspects beyond just the implementation phase. The theme of this workshop is “Expanding the scope of early aspects to include the entire lifecycle.”

Early Aspects workshops are highly interactive, emphasizing discussion and work over presentation. During the morning session there will be short presentations of selected papers. The bulk of the workshop will be reserved for discussions and overall conclusions. Participants will join working groups on specific topics, chosen by consensus. Then, the discussions will be carried out by raising and debating relevant questions related to every topic. Finally, a member of each group will present the conclusions. Each working group will be obligated to produce a record of the discussion summary that will add to the overall Early Aspects body of knowledge.

In keeping with the theme of this workshop, topics dealing with the influence and impact of early aspects throughout the entire lifecycle will be especially emphasized, as will be topics dealing specifically with the with role of early aspects during post-implementation phases (testing, evolution and maintenance, re-engineering, etc.). Other topics that we hope to explore during the workshop include:

- Aspect-oriented requirements engineering: How to identify and model aspects at the requirements level? How to integrate and compose aspects with other modeling mechanisms, such as goals, viewpoints and use cases, and establish trade-offs? How to trace requirements level aspects through later development stages and during re-engineering? How to validate aspects identified at the requirements level?
- Aspect-Oriented domain engineering: What are the criteria for domain aspect decomposition? How can we derive aspects from domain knowledge? How can we abstract and generalize domain aspects for reuse? What are the composition relations between domain aspects? How do we represent domain aspects?
- Mapping between aspect-oriented requirements, domain analysis and architecture: Should the mapping be formal or informal? To what is a requirements concern mapped onto? What are the language features required to support a
mapping? What is the benefit ratio of mapping/coding? What are the pros and cons of mapping in the first place?

- Aspect-oriented architecture design: How to reason about architectures and aspects to know that the architecture is a good one (trade-offs between aspects)? How to model the architecture to take aspects into account? When designing an architecture, how and when to identify aspects? How to set the scope for a software product line architecture using aspects? How do aspects relate to the Model-Driven Architecture approach?
- Tool support and automation for aspect-orientation: Which tools are there (or should there be) to support aspect-oriented development?
- Formalisms and notations for specifying aspects: What formalisms can be used at early software development stages?

2. STATUS

Workshops to date have identified several approaches available for incorporating aspects in the domain modeling, requirements and architecture phases of software development [1]. Some existing approaches include aspects explicitly; for others, we can straightforwardly add in aspects ourselves. There is not yet an all-encompassing integrated approach, although at least a picture of such an approach is a goal of the workshop series. Given the ever-increasing role that aspects play in software development, and given their new and compelling role in the early design phases, we can hope for such a methodology in the future. But the lack of such a packaged solution need not deter us from gaining the advantages of working with early aspects. During domain modeling, requirements engineering, software architecture design, implementation, and testing they can be identified, captured, and passed on.

The full potential of early aspects is only now being glimpsed, but indications are that it could become a valuable resource for software system designers.

For more information, visit www.early-aspects.net.

3. ACKNOWLEDGMENTS

The author acknowledges the help of workshop’s other organizers: Awais Rashid (Lancaster University, UK), Ana Moreira (Universidade Nova de Lisboa, Portugal), João Araújo (Universidade Nova de Lisboa, Portugal), Elisa Baniassad (Chinese University of Hong Kong, China), and Bedir Tekinerdoğan (University of Twente, The Netherlands).

We also acknowledge with gratitude the workshop’s outstanding program committee:

- Mehmet Askit, University of Twente, The Netherlands
- Jaelson Castro, Universidade Federal de Pernambuco, Brazil
- Krzysztof Czarnecki, University of Waterloo, Canada
- Anthony Finkelstein, University College London, UK
- Alessandro Garcia, University of Lancaster, UK
- Michael Jackson, The Open University, UK
- Julio Leite, Pontifícia Universidade Católica do Rio de Janeiro, Brazil
- John McGregor, Clemson University, USA
- Paulo Merson, Software Engineering Institute, USA
- Bashar Nuseibeh, The Open University, UK
- Mónica Pinto Alarcón, Universidad de Málaga, Spain
- Stan Sutton, IBM T. J. Watson Research Center, USA
- Jon Whittle, George Mason University, USA
- Mansour Zand, University of Nebraska, USA

4. REFERENCES