

Editorial



Dear Readers,

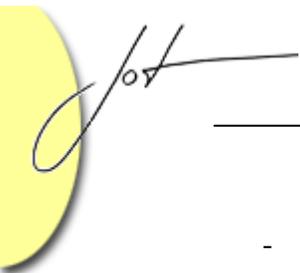
SPLAT is a series of workshops held at the AOSD conference; SPLAT stands for "Software engineering Properties of Languages and Aspect Technologies". Software engineering properties, also referred to as quality or non-functional requirements, or 'ilities', describe the various characteristics of software beyond mere functionality.

Software engineering properties, such as comprehensibility, evolvability, modularity, and analyzability, are crucial dimensions to consider in the assessment of the quality of software engineering activities and products. Generally, designers and users of aspect-oriented languages and systems must understand the effect on the 'ilities' of any aspect-oriented language, feature, system, tool, style, etc. that they choose to use, from the perspectives of multiple stakeholders, including end users, language designers, and tool providers. Quality in software engineering activities and products is often a question of balancing contradictory forces and ideals. It is therefore also critical to understand these trade-offs.

Aspect technologies aim also particularly at the improvement of software engineering properties, especially the above mentioned modularity, comprehensibility and evolvability. As (aspect) languages are being pushed to meet their boundaries and limitations, the trade-offs in language design become increasingly difficult to make. In particular, a trade-off may be perfectly sensible in one application context, but much less so in another.

In this special issue you will find three full papers, originating from the SPLAT workshop series, that address this last issue. Each of these papers resolves a fundamental language design trade-off by offering a language mechanism that lifts the design decisions to the application programmers:

- The first paper, "User-Defined Join Point Selectors-An Extension Mechanism for Pointcut Languages" by Breuel and Reverbel, addresses the fact that aspect programmers are pushing the boundaries of pointcut languages. As a result, aspect language designers have to make a trade-off between limited expressiveness of traditional types of pointcuts, and the complexity of novel and innovative general-purpose pointcut languages. This paper proposes an interface for user-defined pointcuts, which allows for the definition of new, possibly domain-specific, kinds of pointcuts without extending the language.

- 
-
- The second paper, "Developing Law-Governed Systems Using Aspects", by Serban, Tyszberowicz, Feldman and Minsky, does not address a particular problem in aspect language design, but employs aspects as a means to address a problem in the domain of system design. In the design of software systems it is often the case that general (architectural) rules can be defined that the implementation must adhere to. Such rules can be domain specific or application specific, or in some cases even language specific. An example of the latter which is discussed in the paper is rules for encapsulation in combination with inheritance. The paper discusses how such rules can be separated from the language by specifying them as separate programmer-defined 'laws', thereby avoiding the trade-offs implied by making such rules part of the language semantics.
 - The third paper, "Gradual Encapsulation", by Stephan Herrmann, discusses the particular case of the strictness of encapsulation in object-oriented and aspect-oriented languages, and when this is or is not desirable. The paper discusses that the design trade-off varies in different situations, so fixing it in a language is undesirable. It then proposes a flexible solution where several gradations of encapsulation are supported, thereby again resolving the trade-offs that were observed in the first place.

We would like to conclude by thanking, first of all the program committee members of SPLAT'08, who also reviewed the papers for this special issue, and secondly the participants to the SPLAT workshop in 2008 as well as earlier years for their contributions in the form of papers, posters, and many inspired discussions.

Best regards

Erik Ernst, University of Aarhus, Denmark

Lodewijk Bergmans, University of Twente, The Netherlands

Kris Gybels, Vrije Universiteit Brussel, Belgium

SPLAT workshop organizers and guest editors