Principles of aspect-oriented programming languages, design dimensions and the composition filters approach

Mehmet Aksit

TRESE Group
University of Twente
Enschede, The Netherlands
aksit@cs.utwente.nl

ABSTRACT. In traditional engineering disciplines, design is a problem solving process. It aims at mapping problems to realizable solutions, expressed using the artifacts in the corresponding engineering domain. Traditional engineering disciplines have introduced the concepts of canonical component models, systemic models, system construction rules and multiple (simultaneous) models. The emerging phenomena of the aspect-oriented software development techniques can be explained and motivated within this context. The concepts of component models, systemic models, system construction rules, multiple models as defined in traditional engineering disciplines correspond respectively to the concepts of base-level models, aspect models, join point models and multi-dimensional separation of concerns of aspect-oriented languages. Aspect-oriented languages can model the natural systemic properties of problem domains. This allows them to be a better representation than conventional object-oriented languages. This talk will explain the principles of aspect-oriented programming languages from the problem solving perspective. Based on this conceptual framework, aspect-oriented language design dimensions will be outlined. In the second part of the talk, the approach taken by the composition filters model will be explained. The talk will end by comparing filters with other approaches, advantages and shortcomings.

KEYWORDS: aspect-oriented programming, composition filters.