Message from the MDE Session Chair

Martin Törngren
KTH, Stockholm
martin@md.kth.se

Model Driven Engineering (MDE) — or model based development as it is often called in traditional engineering disciplines — is an approach that promises to accelerate development, to improve systems quality, to reduce costs, and also to enable reuse. However, while the use of models and related computer aided engineering tools is very common in established engineering disciplines, MDE approaches are still rarely used in industry for the development of embedded systems. This is despite the realization that higher levels of abstraction are required to describe and analyze systems, to avoid large problems in systems integration, and despite numerous efforts in developing modeling languages, methods and tools. These deficiencies along with industrial needs have stimulated many research efforts to overcome the current situation — the area is indeed very active.

Problems facing researchers and developers include the fact that the area of embedded systems is very large. It does not only cover many application domains and different requirements, but is also strongly characterized by multidisciplinarity, where each discipline has its own traditions, concepts, modeling languages and tools. Embedded systems are characterized by a multitude of relations and interactions between its constituent units and with the environment. Moreover, products including embedded systems typically have to be not only cost-efficient, but also dependable and flexible. These facts to some extent explain the multitude of efforts in different directions, and the incompatibility between different modeling languages and tools.

This session gathers papers dealing with essential facets of model and component based development including modeling language specialization considering the need to express specific aspects, reliability analysis based on architectural models, a comparison of component based and model based software development, and finally efforts towards model and tool integration. The two final papers focus specifically on the latter theme which however is also touched upon by the other papers.

In the first part of the session, the paper by Immonen and Niskanen addresses a gap that often appears between modeling languages and aspect specific analysis, in this case reliability analysis at the architectural level. A tool has been developed to assist both in architecture modeling and reliability/availability analysis on these models. The paper by Bertolini et al. addresses the need for developing and expressing models that incorporate properties of direct relevance for application development. The paper provides a definition of a domain specific language, a profile called NAP (Network Processors Applications Profile), built on the top of UML infrastructure and compliant with the UML meta-model, including specific annotations to deal with performance aspects. The paper by Törngren et al. provides a characterization of component based and model based development. It is shown that the two approaches complement each-other. In particular, model based development requires improved handling of ‘model’ components whereas component based development requires improved component models to assure component composition and reuse. Thus the corresponding research communities need to find better ways of interacting.

In the second part of the session, the paper by El-khoury et al. describes research on model integration and management where an integration platform has been developed to support a database centered integration and management of multiple heterogeneous models representing different views (or aspects) of the product under development. The paper by Alanen and Porres describes and evaluates the evolving model interchange standard for UML models (XMI and XMI-DI) and discusses needs and open issues regarding the current versions of these standards.

I hope that you will find this this session stimulating!