Tool Support for Integrated Development of Component-based Embedded Systems

Gerd Kainz[1], Christian Buckl[1], and Alois Knoll[2]

[1] Cyber-Physical Systems
fortiss
Guerickestraße 25
80805 München
{kainz, buckl}@fortiss.org

[2] Robotics and Embedded Systems
Technische Universität München
Boltzmannstraße 3
85748 Garching bei München
knoll@in.tum.de

Abstract: Nowadays many applications are based on embedded systems and more and more tasks are implemented in software. This trend increases the need of embedded systems and raises their complexity. To deal with this situation we present a model-driven development approach supporting all developers (platform, component, and application) equally during the creation of component-based embedded systems.

1 Approach

The development of embedded systems becomes more challenging while more functions are implemented in software. In addition, formerly unrelated applications have to communicate with each other to form so called cyber-physical systems (CPSes).

Embedded systems are comprised beside application code of a high amount of platform code which consists code related to operating systems, components (including drivers), and glue code. The implementation of those parts requires the involvement of various experts.

We are going to present an overview of our previous work focusing on the design of a model-driven development approach. In contrast to existing approaches we do not only support the application developers but also try to support all other developers (component and application developers) equally. By our approach the tasks of the developers are clearly separated and the approach takes care of working dependencies between them.

Figure 1 shows an overview of the approach consisting of the three parts: modeling, model-to-model (M2M) transformations, and model-to-text (M2T) transformation (code generation). Those are described in the following.

The modeling concept is based on a multi-phase development process. In this process each developer (platform, component, and application) can concentrate on his tasks and the system takes care of a correct integration of the different parts. Thereby, the developers build upon the preceding developments and are guided by the approach. The approach uses a modular design principle based on instantiation and configuration. [KBSK10, KBK11]

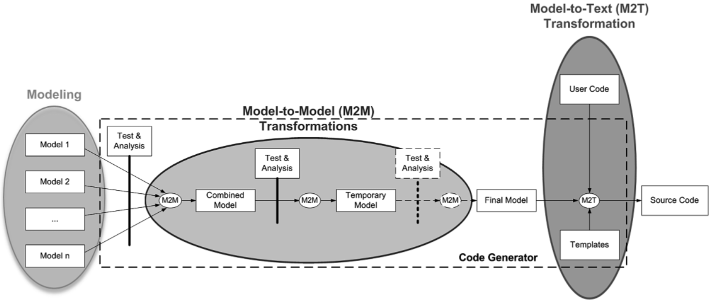

Abbildung 1: Main parts of the presented model-driven development approach for component-based embedded systems

For the transformations from input models to models used for code generation the approach provides a mechanism supporting exogenous M2M transformation chains. This allows breaking down the whole transformation into many small and modular transformations which are easier to understand and maintain. For the creation of the M2M transformation chain the developers only need to specify the structural differences between the input and output metamodels on the metamodel level and the changes to the data of the input model(s) on the model level. The remaining parts of the metamodels/models are handled by a provided tool, which creates the respective output metamodels and models and thereby, reduces any additional effort. [KBK12]

The M2T transformation (code generation) is also based on the multi-phase modeling approach and allows the definition of the various system parts independent of each other. To integrate the various system parts the developer can call functions providing the missing parts. During code generation these function calls are then resolved to the right system parts depending on the provided models. As result of the code generation the integrated system is available.

Literatur

[KBK11] Gerd Kainz, Christian Buckl und Alois Knoll. Automated Model-to-Metamodel Transformations Based on the Concepts of Deep Instantiation. In Jon Whittle, Tony Clark und Thomas Kühne, Hrsg., *Model Driven Engineering Languages and Systems*, Jgg. 6981 of *Lecture Notes in Computer Science*, Seiten 17–31. Springer Berlin / Heidelberg, 2011.

[KBK12] Gerd Kainz, Christian Buckl und Alois Knoll. A Generic Approach Simplifying Model-to-Model Transformation Chains. In RobertB. France, Jürgen Kazmeier, Ruth Breu und Colin Atkinson, Hrsg., *Model Driven Engineering Languages and Systems*, Jgg. 7590 of *Lecture Notes in Computer Science*, Seiten 579–594. Springer Berlin Heidelberg, 2012.

[KBSK10] Gerd Kainz, Christian Buckl, Stephan Sommer und Alois Knoll. Model-to-Metamodel Transformation for the Development of Component-Based Systems. In DorinaC. Petriu, Nicolas Rouquette und ystein Haugen, Hrsg., *Model Driven Engineering Languages and Systems*, Jgg. 6395 of *Lecture Notes in Computer Science*, Seiten 391–405. Springer Berlin / Heidelberg, 2010.