Model-Based Development of Safety-Critical Systems

Christian Buckl & Matthias Regensburger

Embedded Systems & Robotics, Faculty of Informatics, Technical University of Munich (München)

Model-based design has become state of the art in software engineering. Especially, the existence of diverse tools for automatic code generation like Matlab/Simulink or SCADE is very attracting. Particularly for the domain of safety-critical applications, where the developers are typically application domain experts with less background in programming fault-tolerant real-time systems, the possibility to provide extensive code generation would be crucial.

Unfortunately, the code generation abilities of existing tools cover only the functional aspects of the applications, like the control functions. System aspects like process management, scheduling, inter-process communication, communication within the distributed system and fault-tolerance mechanisms are not addressed in general. One reason is the absence of adequate models with an explicit semantic. The widely used Unified Modeling Language UML, for example, lacks the precision and rigor needed for code generation. Only few models, such as class or state machine diagrams, can be therefore used for automatic code generation. A solution to this problem is the usage of domain specific languages (DSL). Another big problem is the platform dependency of system level code. Since safety-critical real-time software is typically embedded in a larger system, there exists a huge heterogeneity of used platforms, the combination of the hardware, operating system and programming language. Due to this variety, it is not possible to design a code generator for system aspects that supports a priori all these platforms. Rather, the code generator must support an easy extension as well of the underlying model, as of the code generation ability. Template-based code generators are a promising approach to obtain extensibility regarding the code generation ability.

In this talk, we will present an approach using meta code generators, applying template-based code generation, to achieve extensibility both on the model and the generation side. In addition, we will discuss the properties of models required for the use in model-based development of system aspects for safety-critical real-time systems. The developed code generator uses a time-triggered model for the specification of the software architecture. Within further models, the developer can specify the hardware architecture, the fault model and the fault-tolerance mechanisms that should be applied. Within two lab application, the approach is tested: a simple control application (balancing a rod by switched solenoids) with control cycles of 500 Hz implemented on a triple-modular redundancy architecture and the control of an elevator by a hot-standby system. All the fault-tolerance mechanisms, as well as the other system aspects as mentioned before, are generated automatically. Therefore, the rate of generated code reaches up to 95% of the whole system.

In the future, we will also apply the approach in an industrial project (funded by BMBF). In addition, we are cooperating with the TüV Süd to receive a certification for the code generator.