Modelling and analysis of real-time coordination patterns

Kemper, S.

Abstract

The increasing size of present-day embedded software systems makes verification of these systems an increasingly difficult task. Even more, not only the size of systems increases, but to faithfully model and analyse real-life applications, an increasing number of features and formalisms need to be developed and supported. The two most important features demanded from embedded software systems are that they need to involve handling of dense real-time, and that they can be developed in a modular, component-based way. The last point amounts to specifying the system by a set of components (implementing the behaviour), together with a set of component connectors (implementing the coordination patterns for inter-component communication protocols).

To ensure that the behaviour of the final system is correct (that means, behaves as expected) and safe (that means, nothing bad can ever happen), it needs to be verified before it is being put into operation. To this end, two things are needed: first, formal models that are powerful enough to faithfully describe all aspects of the system, and in particular support handling of dense real-time as well as constructs to combine components and connectors. Second, formal methods to analyse the formal model of the system and verify that it satisfies certain properties, in particular including correctness of the coordination pattern.

In this thesis, we propose both formal models and formal methods to model and analyse component-based real-time systems and their coordination patterns. We present three formal models of real-time systems: Timed Automata, Timed Constraint Automata, and Timed Network Automata, which have different modelling power with respect to communication, communication primitives, and expressible constraints on the communication. We then present a translation for each of these formal models into a representation in propositional logic with linear arithmetic, which allows to use well-established SAT and SMT solver tools to analyse real-time properties of the underlying system. We give a correctness proof for the representation, which shows that results established for the representation carry over to the respective formal model.

We then present an abstraction technique that works on the representation, and reduces the size of the system by removing parts that are considered irrelevant to the verification of a particular property. This allows to further increase the manageable system size. We prove the abstract system to be an over-approximation

of the original system, such that infeasibility of some erroneous behaviour (up to a certain execution bound) in the abstract system entails infeasibility of the erroneous behaviour (up to a certain execution bound) in the original system. Finally, we prove the applicability and usability of our framework with a tool implementation, that supports the design and analysis process of component-based real-time systems.