Panel III

OCL Goes Real-Time

Moderator:
Franz J Rammig, Heinz Nixdorf Institut, Univ. Paderborn, Germany

Participants:
Stefan Flake, C-LAB, Univ. Paderborn, Germany
Ben Watson, Tri-Pacific, Dallas, USA
John Hogg, Rational, Canada
Wei-Tek Tsai, Arizona State Univ., USA

Summary

The Unified Modelling Language (UML) is becoming more and more the standard for abstract modelling of complex software systems during early phases of the software engineering process. Recent efforts are made to cope with real-time systems as well. UML-RT and RT-UML may serve as examples for such efforts. However discussions are ongoing whether these approaches are sufficient. Strict modelling means for real-time features are not explicitly included in these techniques.

Modelling is one aspect of early phases in software engineering. Another extremely important aspect is the analysis of complex systems. Finally formal verification of models against specification is desired. For this purpose it must be possible to specify consistency conditions and invariants. In the context of UML for this purpose the Object Constraint Language (OCL) has been defined.

OCL as it exists today is very restricted. E.g. it is not possible to make state-based statements about behaviour-oriented UML diagrams like StateCharts, Activity Diagrams or Sequence Charts.

Therefore potential extensions of OCL towards the ability to express real-time will be one topic to be discussed in this panel. Proposals will be presented how to do this. These proposals then have to be cross-checked against needs in practical applications. The basic question of course is whether we really can expect any improvements in the design process of real-time systems if we make use of OCL or extended OCL. It will be discussed which deficiencies can be identified today and what are the main origins of these deficiencies. After this analysis it can be discussed whether OCL or its extensions solve the relevant problems. For such a discussion the potential application areas have to be identified.

One first class of such application areas is clearly the area of analysis and formal verification. Concerning formal verification, embedded systems did inherit from hardware design the technique of Model Checking as the main approach to be followed currently. Model Checking means to prove whether an automaton is a model of a specification given by temporal formulas in CTL or related forms. As the operational modelling technique included in UML is given by StateCharts half of the story my already be covered. But then one would expect OCL to cover the remaining part. Now any available Model Checker software should be applicable.

Unfortunately this dream is not really true. It has to be discussed how this dream to be turned into reality by doing both, enhancing Model Checkers and extending UML/OCL.

This leads to another problem to be solved. Code generated from StateCharts is correct only if this timeliness is guaranteed. Code generation from StateCharts therefore needs to have a sophisticated Worst Case Execution Time (WCET) analysis as integral part. It is an open question how such a system can be supported by specifications given in extended OCL form.

Real-Time Operating Systems (RTOS) and Real-Time Communication Systems (RCOS) constitute another interesting application area for extended OCL specifications. RTOS/RCOS constitute another interesting application area for extended OCL specifications. RTOS/RCOS means guarantee of deadlines. And these deadlines have to be specified. RT OCL may serve to support analysis and verification or an extended OCL may serve as specification means.

In any case clear semantics and a concise, easy to understand syntax will be the prerequisites for such OCL extensions to be accepted by the real-time community.