Real-Time Systems: Specification of Properties in UML

E.E. Roubtsova, TU/e, EESI E.Roubtsova@tue.nl
J.van Katwijk, TU Delft
W.J. Toetenel, W.J.Toetenel@its.tudelft.nl
R.C.M.de Rooij

As the Unified Modeling Language (UML) has become the standard or object-oriented design, users have become more and more interested in using it to elaborate industrial applications. The major attraction of UML is the set of several notations (Class-, UseCase-, Collaboration-, Sequence-, and State-, Statechart- and Activity diagrams) that allow specifying systems on different level of abstraction. To specify properties of systems, UML offers the Object Constraint Language (OCL). OCL is intended to specify invariants on classes and types.

However, the practice of design has shown some of specification problems that take place in UML.The first problem is general for object-oriented approaches. Those approaches have no any standard way handling of time. It rises some difficulties in specifying real time industrial systems. In this class of systems, timing constraints are often as important as functional and ordering ones. Another problem is the lack of means in UML to specify properties of computation paths of systems. Diagrams allow showing a finite set of paths, OCL has no any path notion. Any extension of OCL to present properties of computation paths breaks the idea of the language and makes the language eclectic.

We present extensions of UML that enable specification of real-time systems and its properties and that make possible verification of UML projects.

1. We propose a UML data type Dense Time, the type to be used for clock variables. Each class in a UML specification of a real-time system can define a set of clocks to represent temporal aspects of behaviour.
2. We conseder that the system specification consists of the class diagram and the statechart diagram of the real-time system. This pair of diagrams can be transformed into a specification based on timed automata, which can be translated into computation tree.
3. To specify properties of real-time systems we define specification-classes. Each property is represented as a specification-class that has a predefined formal constraint with parameters. The constraint is presented in an extended variant of Time Computation Tree Logic (TCTL). The variant of TCTL has been put into basis of verification tool Prototype Model Checker.

As a result of our extensions, we are able to give both a specification of a system and a specification of its properties to be verified.