The rCOS language and modeler

Volker Stolz, University of Oslo & UNU-IIST

April 15, 2010

ABSTRACT

The rCOS language (Refinement of Component and Object Systems) offers a unique opportunity for teaching formal methods to software engineers. The language combines a mathematical framework for reasoning about component and object systems with the state of the art of model-based development. UML diagrams are used to capture the static structure of a software system, and the dynamic behaviour of its components. Its use-case driven approach ensure a consistent method of deriving artifacts from the (informal) description.

The business logic of the artifacts is then first specified in a mathematical framework based on the Unifying Theories of Programming with extension for object orientation. Through refinement (semi-automated or manual), correct executable code can be derived from the specification. Refinement is also applied on the modeling level, for example to evolve from an object-based system to a component-based system, or transformation of components.

The tool supports various backends for verification, like component compatibility and reactive behaviour (through model checking), refinement checks by theorem proving, and model-based testing. The interplay between tight integration of the different aspects of designing a model and support for different levels of abstractions allows the lecturer to cover the different phases of designing a software system. Yet the rigorous formal foundation ensures consistency and correctness.