Abstract

Since object-oriented languages model worlds of interacting, independent objects, admitting concurrency in such languages seems to be straightforward. But the integration of concurrency and object-oriented structuring concepts, especially inheritance, induces a new object model and calls for new formal techniques and a new design method. We aim at the design of safety-critical systems in a high-level object-oriented language, whose correctness can only be ensured by the formal techniques of refinement and verification.

Our work is based on the object-oriented concurrent specification language Maude. Maude's distinctive features are its abstract and simple object model and its abstract synchronization and communication mechanisms.

We develop concepts for designing, structuring and reusing object-oriented specifications in Maude, and techniques for verifying and for refining these specifications.

In verification, we provide abstraction techniques and develop notions of invariants appropriate for Maude specifications. In refinement, we develop concepts and techniques appropriate for the refinement of classes, of communication and synchronization patterns and of specifications.

A focal point is the reuse of Maude specifications. We develop a set of concepts for structuring and reuse: states as classes, inheritance, subconfigurations and message algebras. We gain a high degree of code reusability and demonstrate this with examples. We establish the connection between the reuse constructs of Maude and relations between classes of models, and characterize the classes of properties that can be inherited via the different reuse concepts.

Maude is our main specification language. We employ algebraic and coalgebraic specifications and the modal μ-calculus, for covering facets of concurrent objects and levels of abstraction not represented in Maude. We instantiate our object model and our structuring and reuse concepts at various levels of abstraction.

The bounded buffer and a specification of an airport are the running examples. All specifications have been implemented in CafeOBJ.