Abstract

This paper presents a formal technique for incremental construction of system specifications, algorithm descriptions, and simulation proofs showing that algorithms meet their specifications. The technique for building specifications and algorithms incrementally allows a child specification or algorithm to inherit from its parent by two forms of incremental modification: (a) signature extension, where new actions are added to the parent, and (b) specialization (subtyping), where the child's behavior is a specialization (restriction) of the parent's behavior. The combination of signature extension and specialization provides a powerful and expressive incremental modification mechanism for introducing new types of behavior without overriding behavior of the parent; this mechanism corresponds to the subclassing for extension form of inheritance. In the case when incremental modifications are applied to both a parent specification S and a parent algorithm A, the technique allows a simulation proof showing that the child algorithm A' implements the child specification S' to be constructed incrementally by extending a simulation proof that algorithm A implements specification S. The new proof involves reasoning about the modifications only, without repeating the reasoning done in the original simulation proof. The paper presents the technique mathematically, in terms of automata. The technique has been used to model and verify a complex middleware system; the methodology and results of that experiment are summarized in this paper.