Abstract

This paper presents a technique for incrementally constructing safety specifications, abstract algorithm descriptions, and simulation proofs showing that algorithms meet their specifications. The technique for building specifications (and algorithms) allows a child specification (or algorithm) to inherit from its parent by two forms of incremental modification: (a) interface extension, where new forms of interaction are added to the parent's interface, and (b) specialization (subtyping), where new data, restrictions, and effects are added to the parent's behavior description. The combination of interface extension and specialization constitutes a powerful and expressive incremental modification mechanism for describing changes that do not override the behavior of the parent, although it may introduce new behavior.

Consider the case when incremental modification is applied to both a parent specification $S$ and a parent algorithm $A$. A proof that the child algorithm $A'$ implements the child specification $S'$ can be built incrementally upon a simulation proof that algorithm $A$ implements specification $S$. The new work required involves reasoning about the modifications, but does not require repetition of the reasoning in the original simulation proof.

The paper presents the technique mathematically, in terms of automata. The technique has already been used to model and validate a full-fledged group communication system (see [Keidar and Khazan, 2000]); the methodology and results of that experiment are summarized in this paper.