Verification Framework for Object-Oriented Methodology

Masaki Hanada

School of Information Science,
Japan Advanced Institute of Science and Technology

February 15, 1999

Keywords: Object-oriented, Formal Model, Verification Framework.

Today many object-oriented methodologies are proposed. OMT is one of these methodologies. In OMT, a target system is described in three models which are *object model*,*dynamic model* and *function model*. They allow us to model the target system in three orthogonal views. But it is difficult to provide computer supports because of lack of their formalism.

To solve this problem, a formal model for object-oriented methodologies called *formal models* has been proposed. The formal model is provided by formalizing models in OMT, and it is defined using sets and functions.

In this paper, I propose a computer support method for verifications based on fomal model. I call a software that supports a verification on computer *verification application*.

First,we consider how to support these verifications based on the formal model, which are *a consistency verification among models with respect to dataflow*,*a consistency verification method for useing invarians* and *an attribute property verification with respect to a state*. In the verification method dataflows described in a basic functional model are proved using an axiomatic system which consists of axioms and inference rules. In the verification we assign an invariant assertion described in higher order logic to a class,then this assertion is always satisfied despite of any axiom being performed. In the last verification method, we check an attribute property in a state described in higher order logic. It was found through support for the first verification methods that we can automatically prove dataflows using specific algorithm, and through support for the rest that We can not automatially prove programs appearing in both methods but we can support then using a theorem prover.

There are many verifications except these verifications. In this paper,*verification framework* is proposed. In verification framework,verification application is decomposed as frozen spot and hot spot ; rest frozen spot is a common part in verification applications.It is implementation in this framework. Hot spot is various for each application. If

we want to build verification applications, we have to implement hot spot. Thus the cost of building verification application decrease and we can verify with computer easily.

We implement *verification framework* on HOL and ML. In *verification framework*,built models and added information are stored in the reference variables of ML. The reason is because ML is easier to select or process models than HOL. Objects on HOL are built models, associated with them an environment on HOL is generated that is suitable for a specific verification. This is done using AF, TF and GF. AF assigns to HOL objects each element of a built formal model and additional information.TF gives an interpretation of the formal model and additional information. GF generates an environment on HOL based on the formal model and additional information. There are some parts in the above functions mainly depending on the formal model. Such parts are implemented as a frozen spot based on it semantic definition. verification applications are constructed by adding hot spots that implements additional concepts.