

Integrating VSE's refinement in HETS

Mihai Codescu

DFKI Laboratory, Bremen

Sinaia School on Formal Verification of Software Systems, 2008



- General discussion on refinement notion in VSE (Verification Support Environment);
- Example - implementation of natural numbers as lists of bits;
- Using HETS for proving the correctness of the refinement - sketch and future work.



When specifying a system, we ideally begin with an abstract specification which only describes the requirements, even in an informal way, but says nothing about the way the system is going to be implemented. Then, in a stepwise manner, more and more design decisions (e.g. the choice of a certain algorithm or some data structure) are incorporated until we reach a specification which can be easily translated into a program. Each of the steps is called a refinement step, and a refinement step is correct if each model of the more 'concrete' specification is also a model of the 'abstract' one.



The Verification Support Environment (VSE) is a tool for formally specifying and verifying complex systems.

The logic underlying its specification language is multi-sorted first-order logic with equality and induction principles (i.e. allows restriction of classes of models to term generated or freely generated ones).

Along with structuring operations like enrichments and sums of specifications, a notion of refinement is included in the language constructs. The target logic of refinements is Dynamic Logic.



Dynamic Logic (DL) extends predicate calculus with formulae of the form $[\Pi]\phi$ and $\langle \Pi \rangle \phi$ where ϕ is a DL-formula and Π is a program written in a Pascal-like language (with *skip*, *abort*, assignments, *if then else fi*, *while do od* and mutually recursive procedures). These formulae allow us to reason about termination of programs: $[\Pi]\phi$ has the meaning that if the program Π terminates, the formula ϕ holds after the execution of Π , while $\langle \Pi \rangle \phi$ means that the execution of Π terminates and the formula ϕ holds after the execution.



VSE has the notion of *mapping*, where is defined the way the sorts and the (function and predicate) symbols of the abstract specification are implemented by the sorts and procedures of the concrete DL specification.

Natural numbers






We consider the abstract data type of natural numbers with 0, successor, addition and a predecessor function and we choose to implement them more efficiently as lists of bits (see DynLogic.het).

Using HETS for proving the refinement is correct

Tool available at www.dfki.de/sks/cofi/hets



References

-  CoFI (The Common Framework Initiative).
CASL Reference Manual.
LNCS 2960 (IFIP Series). Springer, 2004.
-  Till Mossakowski.
Heterogeneous specification and the heterogeneous tool set.
Habilitation thesis, University of Bremen, 2005.
-  Till Mossakowski and Christian Maeder and Klaus Lüttich.
The Heterogeneous Tool Set.
Editors: Orna Grumberg and Michael Huth, TACAS 2007.
-  Dieter Hutter et al.
Verification Support Environment (VSE)
Journal of High Integrity Systems, 1996
-  David Harel, Dexter Kozen, Jerzy Tyurin.
Dynamic Logic.
MIT Press , 2000

