

SUAVE: Extending VHDL to Improve Modeling Support

Position Statement for Second Workshop on the Future of VHDL

Peter J. Ashenden

Dept. Computer Science,
University of Adelaide, SA 5005
Australia

petera@cs.adelaide.edu.au

Philip A. Wilsey and Dale E. Martin

Dept. ECECS, PO Box 210030
University of Cincinnati
Cincinnati, OH 45221- 0030
USA

phil.wilsey@uc.edu, dmartin@ececs.uc.edu

Abstract

VHDL is widely used by designers of digital systems for specification, simulation and synthesis. Increasingly, designers are using VHDL at high levels of abstraction as part of the system-level design process. At this level of abstraction, the aggregate behavior of a system is described in a style that is similar to that of software. Data is modeled in abstract form, rather than using any particular binary representation, and functionality is expressed in terms of interacting processes that perform algorithms of varying complexity. A subsequent partitioning step in the design process may determine which aspects of the modeled behavior are to be implemented as hardware subsystems, and which are to be implemented as software.

Experience in the software engineering community has led to adoption of object-oriented design and programming techniques for managing complexity through abstraction data types (ADTs) and re-use. Features included in programming languages to support these techniques are abstraction and encapsulation mechanisms, inheritance, and genericity. The term “object-based” is widely used to refer to a language that included abstraction and encapsulation mechanisms. The term “object-oriented” is used to refer to a language that additionally includes inheritance.

While VHDL can be used for modeling at the system level, it has some deficiencies that make the task more difficult than it would otherwise be. These difficulties center around language features (or lack of some features) for supporting complexity management. VHDL is currently somewhat less than object-based, as its encapsulation mechanism are weak. It is certainly not object-oriented, as it does not include any form of inheritance. While it does include a mechanism for genericity, that mechanism is severely limited, allowing only parameterization of units by constant values. We have discussed these issues in a previous paper.

SUAVE aims to improve support for high-level modeling in VHDL by extending the language with features for object-orientation and genericity. As well as adding specific language features, some existing features are generalized. We have previously argued that extending VHDL in this way has the side-effect of improving its expressiveness at all levels of abstraction.

Workshop participants can access our publications at <http://www.ececs.uc.edu/~petera/suave.html>. Included is a working draft of a Technical Report describing the specific language extensions we propose.

* This work was partially supported by Wright Laboratory under USAF contract F33615- 95- C- 1638.