A Comparison of Alternative Extensions for Data Modeling in VHDL

Peter J. Ashenden <petera@cs.adelaide.edu.au>

Philip A. Wilsey <p.wilsey@ececs.uc.edu>

Technical Report TR-02/97

Department of Computer Science
The University of Adelaide, SA 5005
Australia

Technical Report TR-203/05/97/ECECS

Department of Electrical and Computer Engineering
and Computer Science
University of Cincinnati,
PO Box 210030
Cincinnati, OH 45221−0030
USA

This work was partially supported by Wright Laboratory
under USAF contract F33615−95−C−1638

Abstract

High-level modeling of digital systems involves behavioural descriptions that use abstract representations of the data manipulated by the system. For a hardware description language to be used successfully for high-level modeling, it must provide appropriate features, such as object-oriented features, for managing the complexity in a design.

A number of proposals for object-oriented extensions to VHDL deal with object-oriented extensions to support the modeling of data. They provide features for expressing data abstraction, encapsulation, and inheritance with polymorphism, features that are central to object oriented techniques. There are two approaches to object-oriented data modeling: *class-based* and *programming by extension*. This report compares the two approaches in the context of hardware modeling in VHDL. It outlines the two approaches and presents examples illustrating how they might be included in VHDL. It discusses the issue of integration with signal semantics, and compares the way in which the approaches deal with encapsulation, initialization or objects, and inheritance. The report concludes that, while both approaches are viable, the programming by extension approach is preferred.

1 Introduction

As the complexity of digital systems is increasing, designers are becoming more reliant on high-level modeling as a stage in the design flow. High-level modeling involves behavioural descriptions that use abstract representations of the data manipulated by the system. The focus is on capturing the required functionality of the system without committing to implementation details, such as binary encoding of data and communication protocols between components.

For a hardware description language to be used successfully for high-level modeling, it must provide appropriate features for managing the complexity in a design. The software-engineering community has adopted object-oriented techniques as the means of managing complexity in software systems [2]. Since the task of high-level modeling using a hardware description language is very similar to software development, it is appropriate to consider introduction of object-oriented features into a hardware description lanugage such as VHDL.

A number of proposals for object-oriented extensions to VHDL have been published, and are surveyed in [1]. A number of them [4, 8, 10, 11, 13] deal with object-oriented extensions to support the modeling of data. They provide features for expressing data abstraction, encapsulation, and inheritance with polymorphism, features that are central to object oriented techniques.

There are two approaches to object-oriented data modeling: *class-based* and *programming by extension*. These approaches are briefly outlined in [1]. The proposals of Radetzki *et al* [10] and Willis *et al* [13] are class-based, whereas those of Dunlop [4], Mills [8], and