


SAVANT: An Extensible Object-Oriented Intermediate for VHDL*

Dale E. Martin and Philip A. Wilsey
Computer Architecture Design Laboratory
Dept. of ECECS, PO Box 210030
Cincinnati, OH 45221-0030
dmartin,paw@ece.uc.edu

Praveen Chawla
MTL Systems, Inc.
Dayton, OH

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

 COMPUTER ARCHITECTURE DESIGN LABORATORY

Abstract

The SAVANT project is a joint effort between the University of Cincinnati and MTL Systems, Inc. to build an extensible, object-oriented intermediate form (IF) for VHDL. The primary goal of SAVANT is to stimulate the integration of VHDL technology into the research CAD community. The use of an extensible object-oriented format means that the class definitions of the IF can be extended with additional data and method definitions; user's benefit from the fact that the nodes are self defining and that overloading can be exploited for building IF node processing. Since the IF is fully extensible, no procedural interface is provided or needed.

The SAVANT project includes a C++ implementation of the IF. Important in this implementation is the design of a class hierarchy that allows extensibility of the IF without allowing the modification of the core IF definition. Essentially this is achieved by organizing the base IF nodes into two parts, namely: a set of "base" class definitions, and a corresponding set of "final" derived class definitions. User defined extensibility is achieved by (i) deriving new (user defined) classes from the "base" node definitions and (ii) revising the inheritance path of the "final" nodes to the new user defined classes.

This SAVANT project includes a suite of software tools to implement a VHDL parser to IF translator (called SCRAM) and extensions to the IF to support node rewriting and code generation. In particular, the static rewriting rules from the RASSP formal modeling project are implemented as a collection of overloaded methods called transmute. This rewriting eliminates several of the redundant nodes from a specific instance of the IF (for example replacing concurrent signal assignment statements with their equivalent process statement representation). Code generation capabilities exist to regenerate VHDL or to generate C++ suitable for linking with the TyVIS VHDL simulation kernel (the UC parallel VHDL simulation kernel). The SAVANT project is more fully documented on the www at <http://www.mtl.com/projects/savant> and <http://www.ece.uc.edu/~paw/savant>.

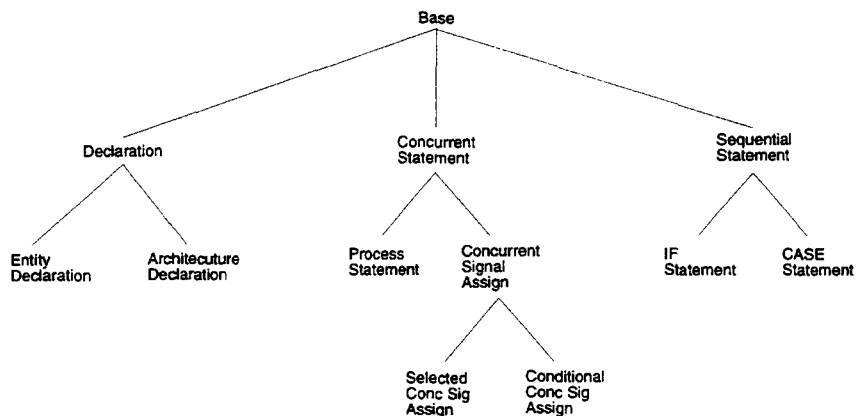
 COMPUTER ARCHITECTURE DESIGN LABORATORY

1

IF Definition: Traditional Approach

1. Define IF as OO Structure
2. Implemented as non-OO data structure
3. Support with procedural interface to:
 - (a) Access/traverse data structure
 - (b) Define, set, and access user defined attributes
4. Unable to benefit from OO structure

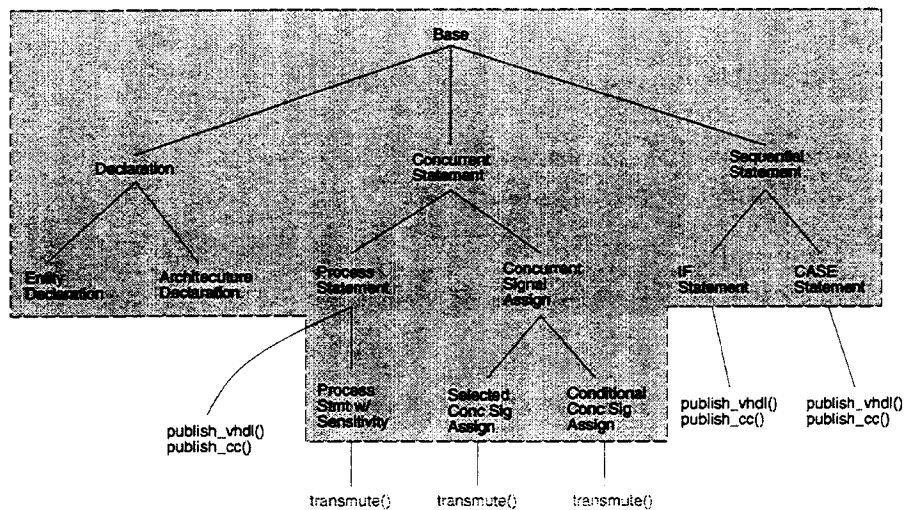
Traditional IF Definition



SAVANT IF Objectives

1. Build OO IF
2. Exploit self definition of objects
3. Define an extensible IF
4. Support CAD tool integration
5. Useful for industrial *and* academic communities
6. Publically available
7. Pursue approach to enable users to benefit from OO structure

An Extensible IF



Extension Requires Cohabitation

- Extensions must span wide range of CAD back-ends
- Separation of extensions for distinct CAD back-ends
- Integration of extensions for distinct CAD back-ends

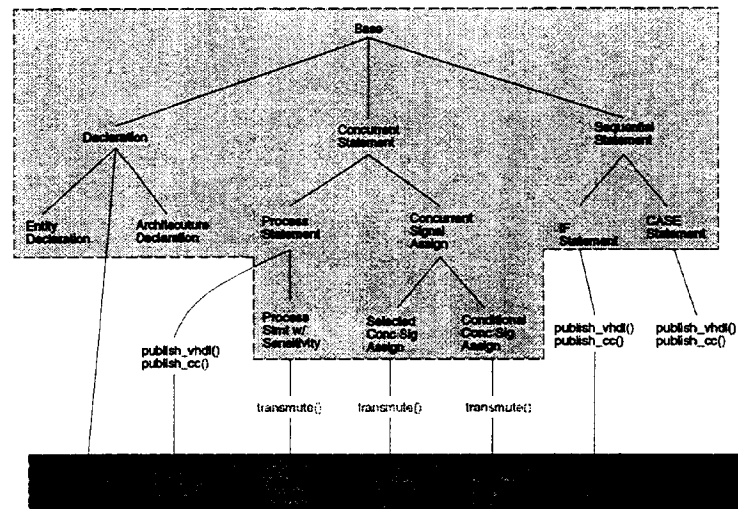
SAVANT IF Structure

1. SAVANT IF contains only data declarations
2. Contains base node from which all elements are derived
3. Annotations treated as extensible components
4. All procedural access routines defined as extensions

SAVANT Implementation

1. Need portable software (C++)
2. Low cost solution (free or nearly free)
3. Well-documented (ideally, automatically generated from C++)
4. Hyperlinked documentation

SAVANT IF Implementation

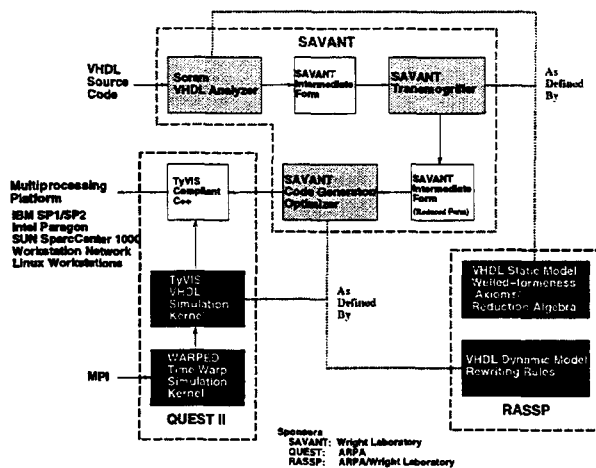


SAVANT Software

- SCRAM: VHDL analyzer-to-IF
- transmute: implements RASSP formal models reduction algebra
- archive: file IF (jointly developed w/ FTL Systems, Inc)
- publish: code generation (to TyVIS kernel)
- autdoc: automatic HTML/texti documentation from C++ headers

SAVANT/QUEST/RASSP Project Integration

 **Computer Architecture Design Laboratory**



Summary

- VHDL '93 only
- SAVANT Software freely available for non-commercial use:
 - Releases via web at <http://www.mtl.com/projects/savant>
 - Alpha release available
 - Linux/SOLARIS versions available, NT versions coming
- Simulation kernels:
 - TyVIS info at <http://www.ece.uc.edu/~paw/tyvis>
 - WARPED info at <http://www.ece.uc.edu/~paw/warped>
- RASSP formal modeling:
 - working document at <http://www.ece.uc.edu/~paw/rassp>

Acknowledgements

- Joint effort w/ MTL Systems, Inc.
- File format joint w/ FTL Systems, Inc.
- Supported by Wright Patterson