STEVE: A Syntax-Directed Editor for VHDL Based on SAVANT

Katrina E. Kerry
Peter J. Ashenden
Michael J. Oudshoorn
University of Adelaide
VIDE

• Framework for integrated graphical and textual VHDL design entry
  – graphical for structural aspects
  – textual for procedural aspects
Syntax Directed Editing

• Editor helps designer create syntactically correct models
  – correct by construction
• Template-based
  
  \[
  \text{context\_clause\_list} \\
  \text{entity identifier is} \\
  \text{port\_list;} \\
  \text{end entity identifier;}
  \]

• Select template from menu or by command
Automatic Template Insertion

• Extends the idea of auto-completion
  – replace place-holder by a valid template
  – designer starts typing prefix to select template

```vhdl
context_clause_list
tlibrary_unit e
tcontext_clause_list
tentity identifier is
tport_list;
tend entity identifier;
```
SAVANT IR

• Object-oriented Intermediate Representation for integrating tools
• *Extensible* hierarchy of C++ classes
  – classes represent syntactic categories in VHDL
SAVANT IR

Base

Extension classes

Final_Base

Base_DesignUnit

Extension classes

Final_DesignUnit

Base_EntityDeclaration

Extension classes

Final_EntityDeclaration
Extensions for STEVE

- Each syntactic category needs to include
  - its place-holder
  - its template
  - operations to display and edit, including selection of templates for insertion
Extensions for STEVE

[Diagram showing relationships between Base, Final_Base, Display_Base, Base_, Display_, Final_, MIR_Symbol, List_of, MIR_Template, MIR_OptPlaceholder, MIR_Placeholder, MIR_Identifer, MIR_OptPlaceholder, MIR_Layout, MIR_OptToken, MIR_Token, MIR_OptChoiceToken]
Identifier Completion

• Like Emacs and tcsh
• Candidates for completion:
  – visible names, of the appropriate kind
• Symbol table class
  – has static members: pools of symbols organized by kind
  – methods for prefix matching
  – integrated into SAVANT class hierarchy
Conclusions

• SAVANT class structure
  – framework for integrating new tool features
  – extensibility works well

• Automatic Template Insertion
  – accelerates design entry
  – syntax “correct by construction”

• Future work
  – more of VHDL, migrate to AIRE