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

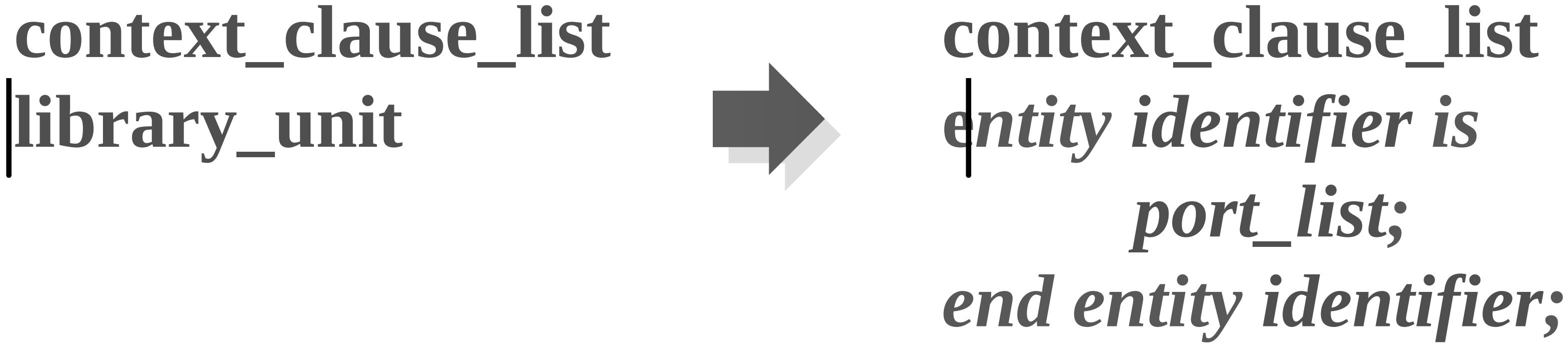
- Editor helps designer create syntactically correct models
  - correct by construction
- Template-based

```
context_clause_list
entity identifier is
        port_list;
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



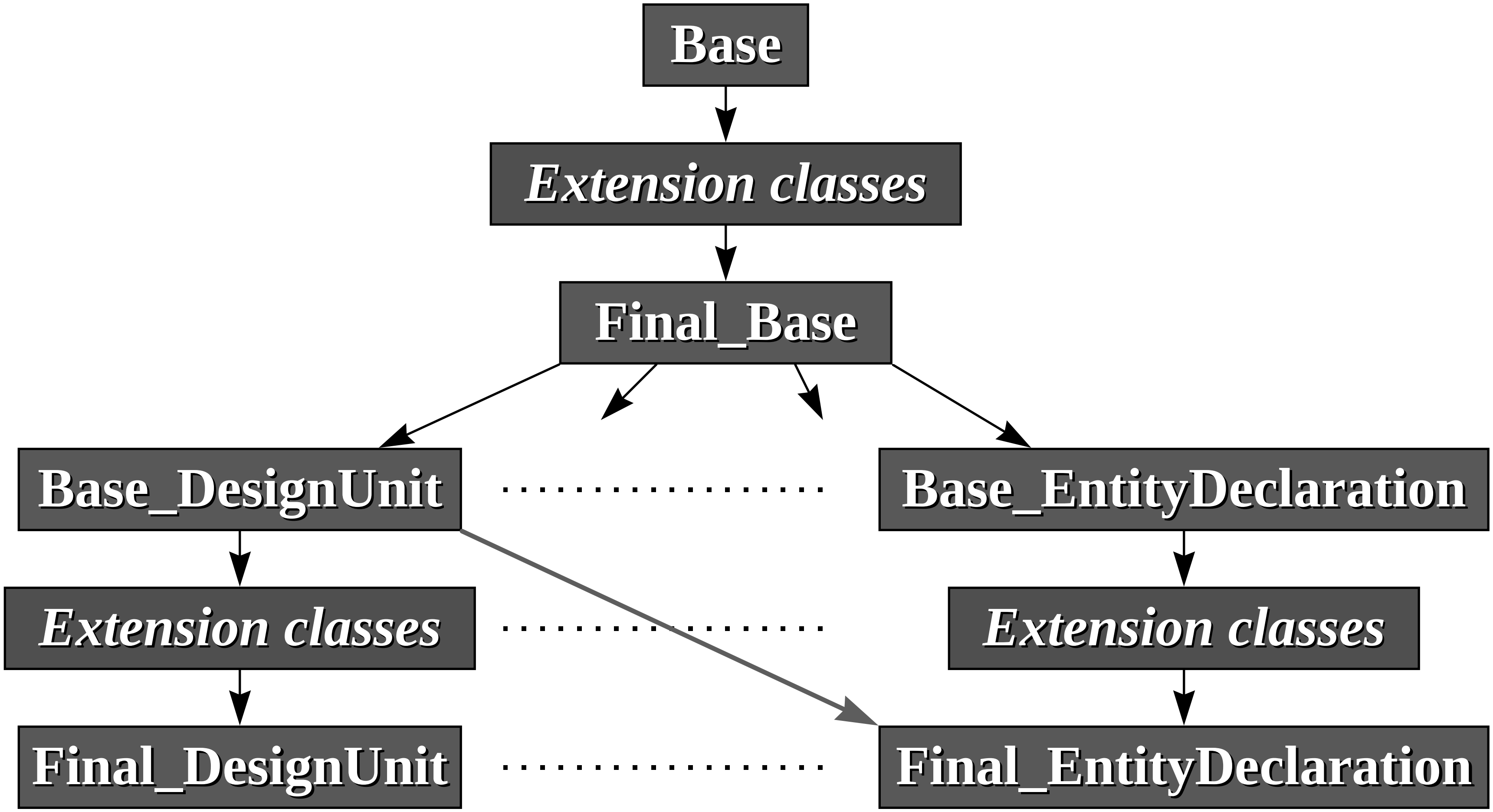
context_clause_list
library_unit

context_clause_list
entity identifier is
port_list;
end 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

. . . .

Base_EntityDeclaration

Extension classes

. . . .

Extension classes

Final_DesignUnit

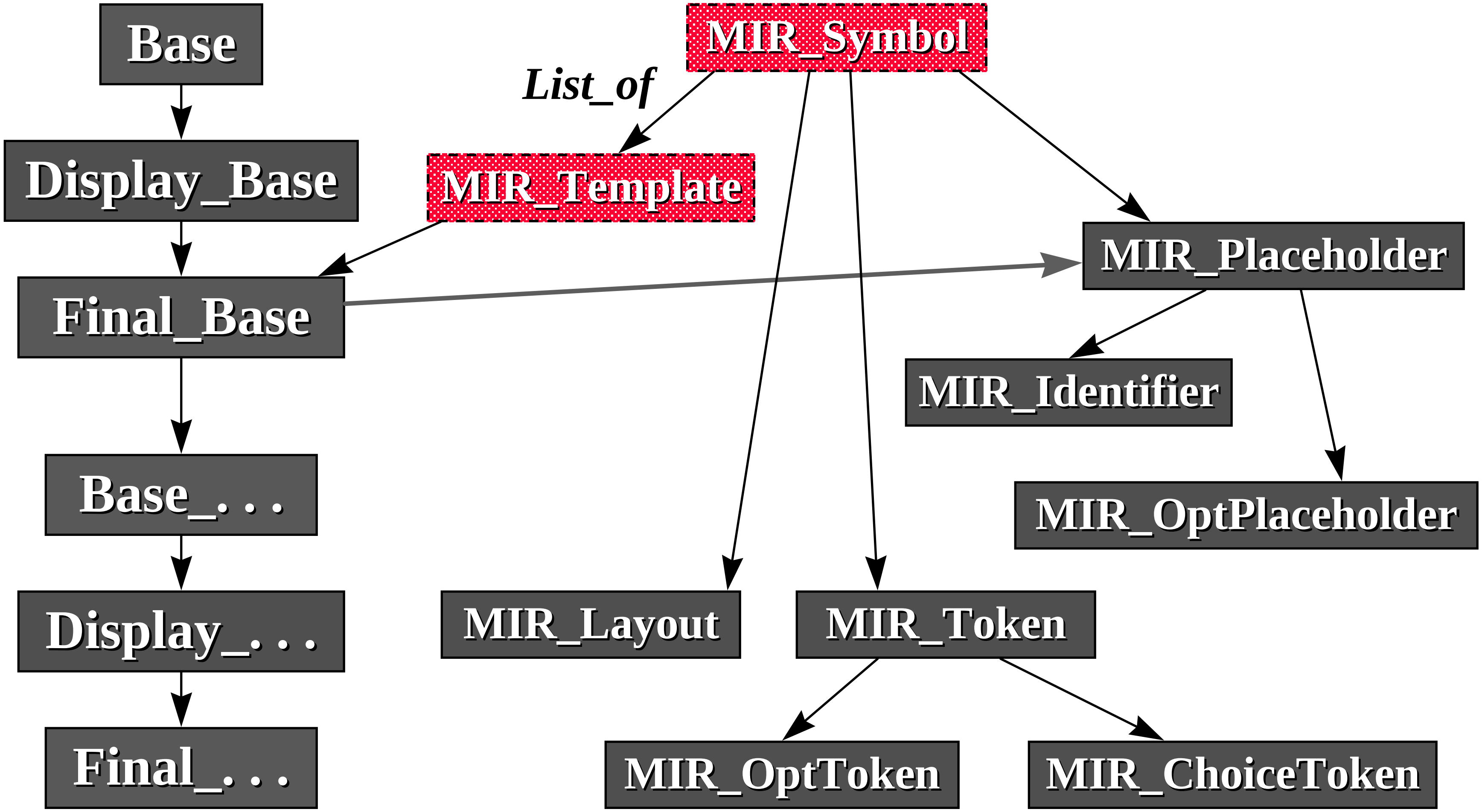
. . . .

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



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