6
Conclusion

Theoretical studies arrived on the need of graph based theorem provers. These theorem provers can handle descriptions of circuit-structured deduction systems as well as tree-oriented systems. The Cathedral Theorem Proving Platform, presented in this dissertation is a toolset to instantiate such theorem provers.

The defined and implemented platform contains a standalone virtual machine and a standalone upper level language compiler. These tools have well defined interfaces, ready to be extended. As a project, the proving platform can be integrated with a richer IDE, like Eclipse [7].

It is important to have integrated tools to help users on the logic specification phase, in the equation writing phase and in the execution phase. Eclipse provides the integration platform needed. It will also give the machine's graphical user interface a professional quality library and a fast development at low cost.

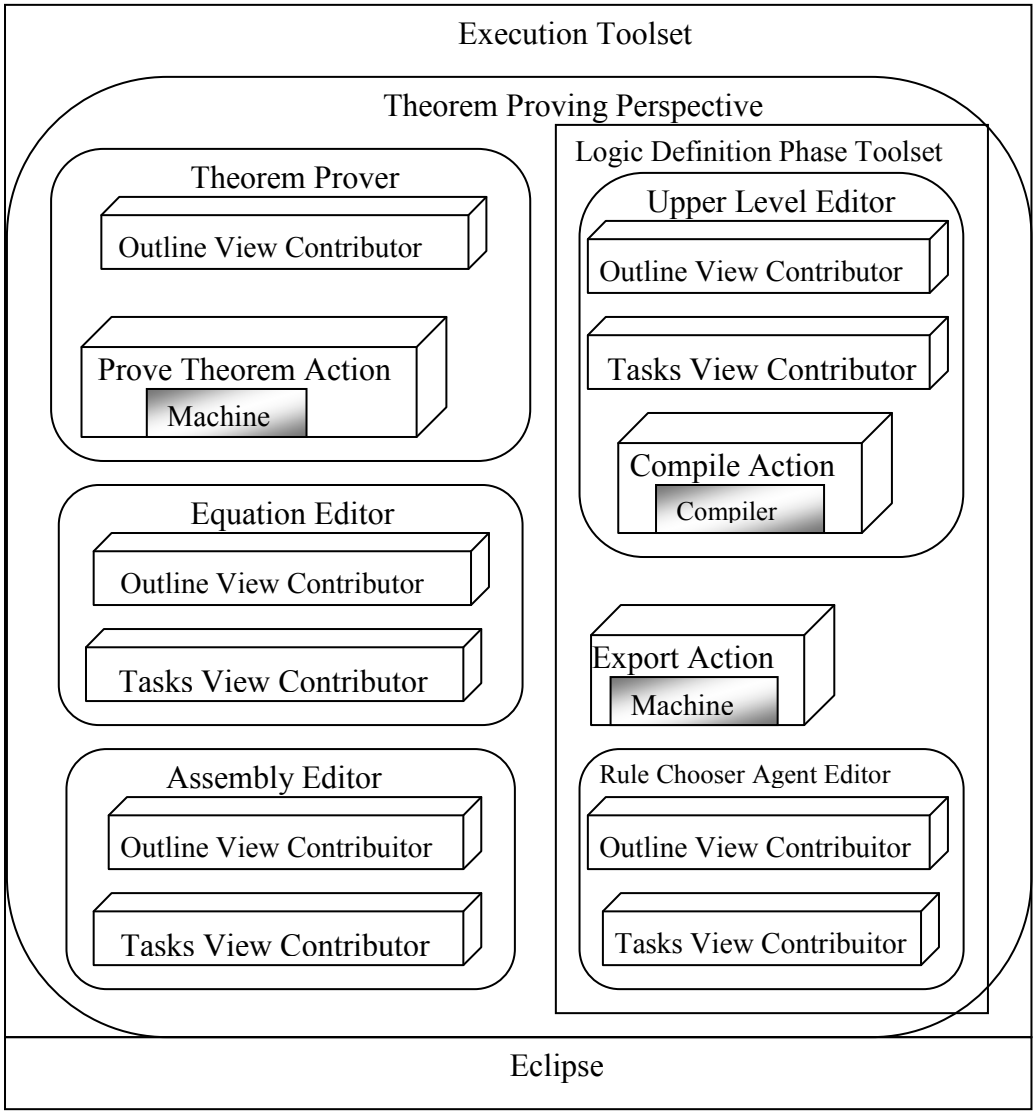

Figure 11 The Saint Thomas Aquinas Cathedral Plug-in Architecture

The components diagram above shows all the Eclipse extensions in the planned distribution. The only stand alone parts are the gray boxes: The Saint Thomas Aquinas Machine and Fr William of Moerbeck's Compiler. They're always related to an action, which is a contribution to the eclipse that starts a standalone execution – a button in a taskbar. Fr William of Moerbeck's Compiler will be called by the "Compile Action" to compile a file edited in the Upper Level Editor. The Saint Thomas Aquinas Machine can be called by the "Prove Theorem Action" to prove a theorem specified in the Theorem Prover, or by the "Export Action" which exports a machine and a Chooser Agent as a standalone Java application. This stand alone application JAR has also an API to be used in

integration. In this way, The Saint Thomas Aquinas Machine can be a component in a larger project.

Two Eclipse plug-ins are provided: the Execution Toolset and the Logic Definition Phase Toolset. Four editors are provided. Each editor contributes with a task view adapter which helps the user visualizing errors and missing code, and with an outline view adapter which helps navigating in the editor.

Execution Toolset aggregates all the Eclipse's extensions. Its Theorem Proving Perspective masks Eclipse's undesired tools. Prove Theorem Action is the extension responsible for running the virtual machine. It requires as input an equation file and a program file with the reduction rules described. The editors help the development of these files with syntax highlight, error checking and text auto-completion.

The Logic Definition Phase Toolset is a plug-in that extends Execution Toolset. To be run, the contributions to Eclipse on it depend on the Theorem Proving Perspective. The toolset presents Fr William of Moerbeck's Compiler and its upper level language, the best way to describe reduction rules and theorem provers. The toolset has an export action which generates a Java based inference machines (or proof assistants) gathering machine's jar, the reduction rules description file, some interfaces as API, configuration files for standalone running, and eventually a rule choosing agent with the knowledge of reduction rule decision. This agent's knowledge can be programmed in its appropriated editor, which is also provided.

The virtual machine's and the compiler's stand alone jars are fully functional in a graphical interface with these toolsets. Four editors and three actions are provided, extending Eclipse integrated environment, and providing full functionality.