Infosys®

POWERED BY INTELLECT
DRIVEN BY VALUES

CLIPS Plug in for Protégé

Sixth International Protégé Workshop, July 7-9, 2003

Agenda

- Motivation
- CLIPS Overview
- CLIPS Plug in for Protégé
- Solution
- Limitations And Future Possibilities

Motivation for developing CLIPS Plug in

- This integration would allow for complete Expert System development from within Protégé
- An alternative to Jess and Algernon for rule development from Protégé
- CLIPS allows specifying rules using the ontology classes and instances directly using object patterns. No need to switch back and forth between instances and facts.
- The default storage model of Protégé is compatible with CLIPS
- CLIPS is a widely used forward chaining rule-engine and has very relaxed licensing model
- It would bring CLIPS user community closer to Protégé

What is CLIPS?

- CLIPS stands for C Language Integrated Production System
- CLIPS is a popular expert system development tool, written in C
- Initially developed by NASA, currently maintained as public domain software
- Supports multiple programming paradigms
  - Forward chaining rule language based on Rete Algorithm
  - Procedural Programming
  - Object-Oriented Programming (COOL)

Embedding CLIPS

- CLIPS architecture allows it to be embedded in other programs
- It is simple - Just replace the CLIPS main program with user defined main program
- Call InitializeEnvironment function in <sysdep.h> before loading constructs
- Please refer to CLIPS Advanced Programmer's Guide for more details

CLIPS I/O Router System

- The CLIPS I/O router system allows to change the input and output routing of the CLIPS Engine easily.
- Uses logical names to send I/O requests without referring to device or function actually handling the request.
- CLIPS code base uses pre-defined set of logical names like stdin, stdout, wprompt, wdialog, wdisplay, werror, wwarning, wtrace
- Each I/O router has a priority. Priority determines which routers are queried first when determining the router that will handle an I/O request.
- CLIPS allows new routers to be added for handling I/O requests for logical names.

Protégé And CLIPS

- Protégé's default save/load file format, for both classes (.pont) and instances(.pins), is compatible with CLIPS.
- CLIPS Compatible Protégé extensions are represented as CLIPS comments
- There are certain features that are not compatible. Details available on Protégé web site (http://protege.stanford.edu/doc/design/clips.html)

Infosys

Issues in embedding CLIPS in Protégé

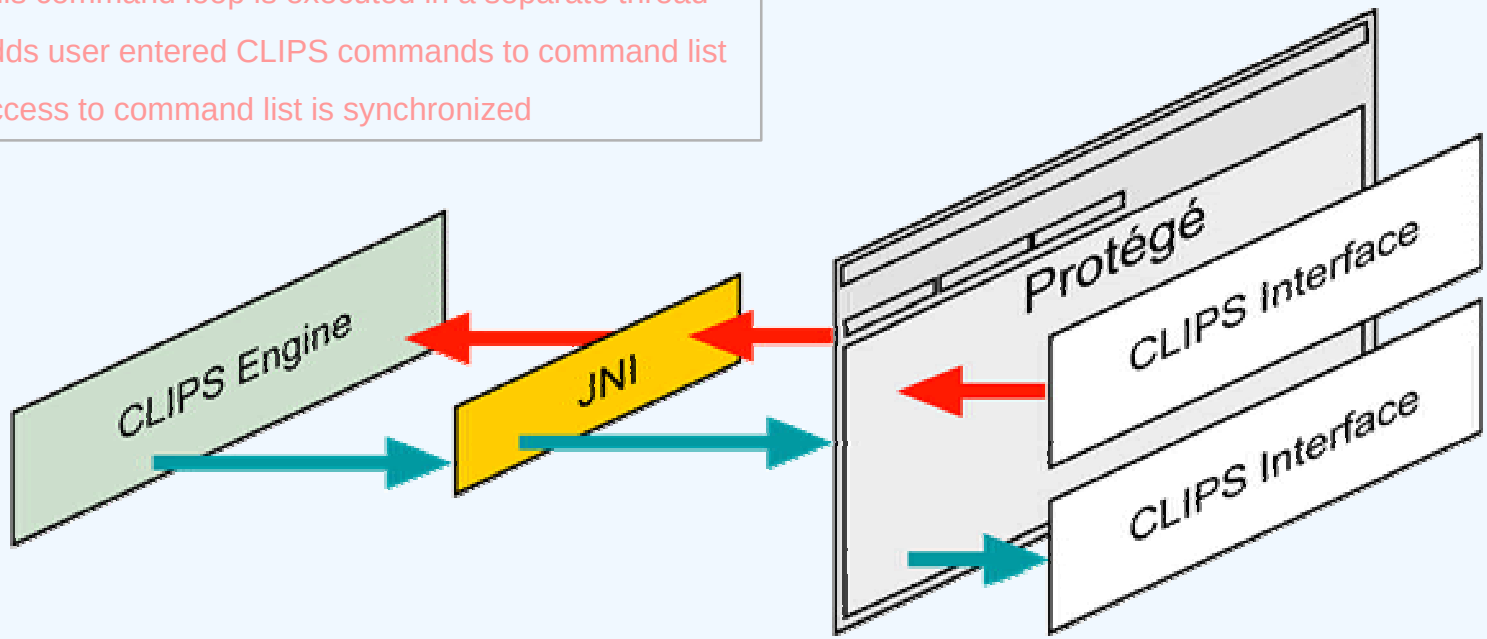
- CLIPS is written entirely in C and Protégé in Java
- Execution of CLIPS command loop should not block the execution of the Protégé's main thread

Solution

- Use Java Native Interface (JNI) to interface with CLIPS C Code
- Run the CLIPS Command line on a separate thread

1. Plug in loads the CLIPS shared library that implements the CLIPS command loop as a JNI method.
2. This command loop is executed in a separate thread
3. Adds user entered CLIPS commands to command list
4. Access to command list is synchronized

1. Initializes the CLIPS Engine
2. Command Loop makes JNI method calls to access the command list



Infosys

JNI Basics

- The Java Native Interface (JNI) is the native programming interface for Java that is part of the JDK.
- The JNI allows Java code that runs within a Java Virtual Machine (VM) to collaborate with applications and libraries written in other languages, such as C, C++, and assembly.
- The framework also allows native(C, C++ etc) code to call Java methods.

Java code

- Two classes implement the CLIPS Tab plug-in
  - CLIPSTab.java
    - Loads the native shared library containing the C object code
    - Extends the AbstractTabWidget class, implements the tab and instantiates the CLIPSRuntime
  - CLIPSRuntime.java
    - Implements runnable interface and executes the native CLIPS command loop in the run() method.

C code

- Since the CLIPS code is compiled as shared library no main method required
- The code provides the implementation of the native command loop method which is called from the Java side.
- Implements the PrintInterfaceRouter for redirecting the outputs from the CLIPS engine to the console of the Protégé CLIPS Tab
- Implements GetcInterfaceRouter which accesses the CLIPSRuntime on the Java side to get the next character from the command string
- The routers use a priority of 10 in-order to re-define the default user-interface I/O router

Versions of software used

- Protégé-2000

- CLIPS 6.2

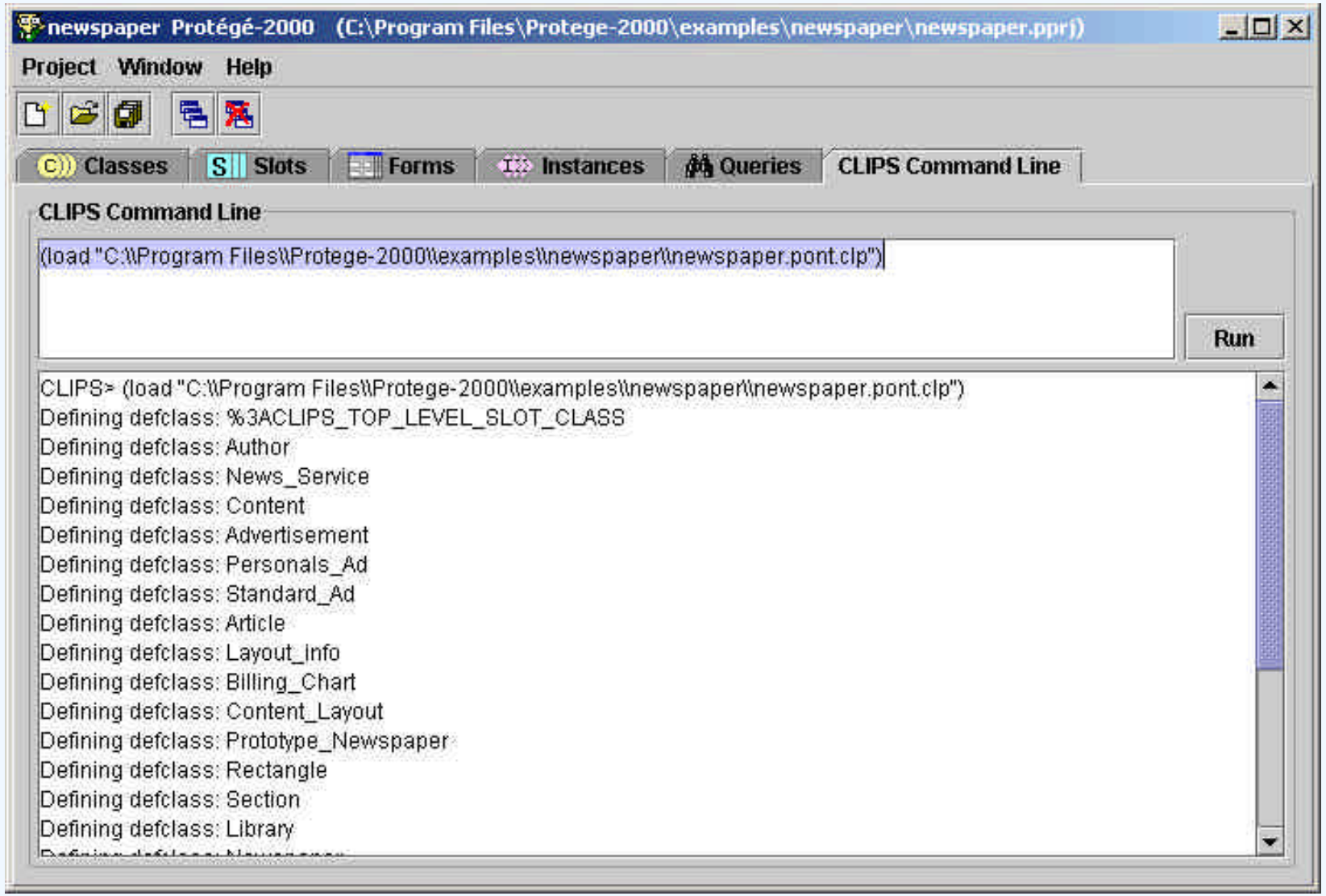
- JDK 1.4

- Windows 2000

- Visual Studio .NET

Infosys

Screen Shot


newspaper Protégé-2000 (C:\Program Files\Protege-2000\examples\newspaper\newspaper.pprj)
Project Window Help
Classes Slots Forms Instances Queries CLIPS Command Line
CLIPS Command Line
(load "C:\\Program Files\\Protege-2000\\examples\\newspaper\\newspaper.pont.clp")
Run
CLIPS> (load "C:\\Program Files\\Protege-2000\\examples\\newspaper\\newspaper.pont.clp")
Defining defclass: %3ACLIPS_TOP_LEVEL_SLOT_CLASS
Defining defclass: Author
Defining defclass: News_Service
Defining defclass: Content
Defining defclass: Advertisement
Defining defclass: Personals_Ad
Defining defclass: Standard_Ad
Defining defclass: Article
Defining defclass: Layout_Info
Defining defclass: Billing_Chart
Defining defclass: Content_Layout
Defining defclass: Prototype_Newspaper
Defining defclass: Rectangle
Defining defclass: Section
Defining defclass: Library

Limitations

- Plug-in does not load the classes and instances of the Knowledge Base automatically due to Protégé CLIPS incompatibilities like pattern-match facet

- The development and testing was done on Windows 2000. The tab has not been tested on other Operating Systems

- The plug in can only accept one command at a time. Command List will be locked while CLIPS Engine processes a command.

Future Possibilities

- Idea was to develop a simple Proof-Of-Concept application
- Can improve the interface to support more functionalities in line with wxCLIPS (Graphical Command-line interface for CLIPS)
- Can be enhanced to JessTab like two-way bridge
- Source will be given to Protégé community

Demo

Infosys®

POWERED BY INTELLECT
DRIVEN BY VALUES

Thank You

Krishnakumar Pooloth (krishnakumarp@infosys.com)
Rizwan Ameer (rizwan_ameer@infosys.com)