**Project Description**

We will be investigating different expert system development tools commercially and freely available. If sufficiently advanced tools exist that are not discussed in the class, we will look into creating teaching material that use these tools.

**User Feedback**

At this point, no user feedback is available, as the desired product of this project is new teaching materials for AI classes covering Expert Systems. We have begun researching the various tools available for developing Expert Systems, and are generating some preliminary reports on usability, which this milestone will focus around.

We also will discuss our findings with our peers, as they are in the same position as the eventual students receiving this material, so they will hopefully be well equipped to understand the concepts presented.

**System Design**

Algernon / Algernon-J


Developed at the university of texas in the early 90s. Interfaces with Lisp. Available for download at [http://www.cs.utexas.edu/users/qr/algv/download.html](http://www.cs.utexas.edu/users/qr/algv/download.html). Work on the original system appeared to end in the late 90s. Has been re-written in Java with the last update in 2005. Some documentation. Uses the same representation as the frame-based knowledge base management system, Protégé. Also available as a plug in for Protégé. Last updated in 2005.

Proceed to next stage: YES

Babylon

Implemented and embedded in Common Lisp. There is a published book on Babylon [The AI-Workbench BABYLON: An Open and Portable Development Environment for the Expert Systems,î written in i92. However it is available only used from Amazon. No longer available from the German National Research Center for Computer Sciences, however CMU seems to have a copy available online. However, this system appears to be a dead end as research stopped in the mid 90s and there is very little documentation or information.

Proceed to next stage: NO

Discrete Event Calculus Reasoner


Performs automated commonsense reasoning. Documentation updated as recently as 2008. There is a book that deals with commonsense reasoning and this program called, “Commonsense Reasoning.” Runs within python. Well documented with examples. More research is needed to discover if this “commonsense” framework can be used more generally as an expert system shell or if it is restricted in some way to a certain set of problems.

Proceed to next stage: YES
Drools
http://jboss.org/drools/
Business rule management system written in Java. Lots of documentation and in active development. The rule engine looks like any other expert system shell. Production rules are written in if-then statements. Supports forward chaining only, however “later” releases will support backwards chaining. Looks very promising.
Proceed to next stage: YES

Euler Proof Mechanism
http://www.agfa.com/w3c/euler/
Implemented in Java, C#, Python, Javascript, and Prolog. Supports logic based proofs. Still maintained/ supported. Interfaces with SQL. Uses logic based proofs to tell if a given conclusion is supported by facts. More research is needed to find out if it can do the work of an expert system shell or is limited to certain domains/ types of problems.
Proceed to next stage: YES

Expect
Has a book, “EXPECT: Intelligent support for knowledge base refinement.” Forms expectations as to what knowledge is missing or incorrect in a knowledge base. Several research papers. Still looking for a website.
Proceed to next stage: MAYBE

F-Logic
Knowledge representation and ontology language. A declarative language to describe object-oriented and frame based languages. Not an expert system shell.
Proceed to next stage: NO

FOCL
www.ics.uci.edu/~mlearn/FOCL.html. Developed in LISP. The machintosh interface allows for building rule bases, but the windows version is only useful for machine learning. Out of development since the mid 90s.
Proceed to next stage: NO

G2
Proceed to next stage: NO

Hank (cognitive modeling environment)
A cognitive modeling environment for non-programmers. Production rules take the form of fact card and instruction cards. Last update in 2003. Designed for use with undergraduate cognitive psychology course. Not a lot of documentation.
Proceed to next stage: MAYBE

InfoSapient
http://info-sapient.sourceforge.net/
Rues engine for Java. Last updated in 2002. Has support for fuzzy logic. Some (not much) documentation in the form of java class descriptions.
Proceed to next stage: MAYBE

E2qlite
http://www.expertise2go.com/webesie/e2gdoc/
-Availability
It is available as a .jar. The text interface for laying out the flow of java forms for a knowledgebase is fairly simple. Cool tool, but not for this class.
Proceed to next stage: YES

JADEX
http://sis-www.informatik.uni-hamburg.de/projects/jadex/jadex-0.96x/tutorial/index.single.html
This tool seems more focused around the development of intelligent agents, rather than knowledge based systems. A tutorial exists for it, but it doesn't seem as simple as Jess or CLIPS.
Proceed to next stage: NO

JAM
Also seems more focused around intelligent agents and learning based agents. Less emphasis is placed on the knowledge based system portion.
Proceed to next stage: NO

JTP
http://ksl.stanford.edu/software/JTP/
Looks something like what we're looking for. A distinct lack of documentation makes this option much less attractive, though. Maybe downloading it would provide more information?
Proceed to next stage: MAYBE

Jason
http://jason.sourceforge.net/
Seems focused around Beliefs-Desires-Intentions (BDI), like JADEX and JAM. If we want to go that direction, we can, but it's more of an intelligent agent tool than a knowledge based system tool.
Proceed to next stage: NO

Jeops
http://www.di.ufpe.br/~jeops/manual/
Looks like an excellent candidate (and simple to use). This gets my stamp of approval.
Proceed to next stage: YES

MIKE (micro interpreter for knowledge engineering)
http://kmk.open.ac.uk/people/marc/mike_text.html
Unable to find resource
Proceed to next stage: NO

Manderx Inference Engine
Unable to find resource
Proceed to next stage: NO

OPS83 & OPS5 (a rule-based programming language)
http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi?OPS5
Possible, but looks to be old. Last updated in 1995. Little documentation exists.
Proceed to next stage: MAYBE

OpenCYC
Looks like a viable language. Lots of documentation available, active development, probably an excellent candidate.
Proceed to next stage: YES

Poplog (multi-language AI development environment)
http://www.poplog.org
Looks like a general purpose programming language. Maybe not what we want?
Proceed to next stage: MAYBE

Prolog (Jlog)
This is an interpreted language, and current development and extensive documentation on prolog makes this an excellent candidate.
Proceed to next stage: YES

Protege
http://protege.stanford.edu/
Free and available via the Stanford site. Used by private, commercial, and governmental groups use it for various knowledge ontological means. It seems like a nice interface for doing OWL, and documentation is rather good.
Proceed to next stage: MAYBE

PyExpect
http://www.noah.org/wiki/Pexpect
People seem to be calling this Pexpect. It is a python module based on Don Libes’ Expect. Expect is a tool for monitoring/automating other applications for either general use or testing. Although it looks fun, it is not for our needs. Proceed to next stage: NO

Ruleby
http://ruleby.org/wiki/Ruleby
This tool uses a Ruby Domain Specific Language and I cannot yet determine how much of a learning curve it has. There is some documentation, but it seems to have a limited following. Some features seem underdeveloped. Proceed to next stage: YES

Shaken (only found KAES)
http://kaes.anthrosciences.net/
Looks like an underdeveloped graduate student project. Proceed to next stage: NO

SnePS
http://www.cse.buffalo.edu/sneps/
This is a tool that uses common lisp, and there are quite a few manuals. Proceed to next stage: NO

Soar
http://sitemaker.umich.edu/soar/home
It is available, but the syntax used looks rather complex with limited functionality. Proceed to next stage: YES

TyRuBa
http://tyruba.sourceforge.net/
Was generated as part of a Ph. D. thesis for generating Java code. Proceed to next stage: NO

XpertRule
http://www.xpertrule.com
Enterprise rapid application development environment. Not publicly available, and it looks very limited. Proceed to next stage: NO

Tmycin
Not available. Only a research project a stanford, and then expanded to University of Texas. Proceed to next stage: NO

Prototype and Implementation

Evaluate the those programs that are marked proceed to next stage “yes.” This involves downloading and testing the programs for basic features, ease of use, and amount of readily available documentation.

Evaluation Plan

According to our overall plan, we evaluated all the expert system shells available. We have identified those programs that need to be downloaded and evaluated. We have met the Week 4 milestone and are up to date.