Expert System Development Tools

Pattarachai Lalitrojwong

Choices

1. The methodology (expert system or other approaches)
2. The mode of knowledge representation within the model (production rules, frames, logic programming, or some combination of these methods)
3. The software development package
4. The hardware upon which the expert system is to be implemented

Some Expert System Tools

1. Expert system shells
2. High-level programming languages
3. Multiple-paradigm programming environments

1. Expert System Shells

- They may be implemented on inexpensive, general-purpose personal computers.
- They exhibit computational efficiencies.
- They (usually) permit multiple modes of chaining.
- They provide bridging features that permit access to external programs.
Differences among Rule-based Shells

- The language(s) by which extensions to the package may be developed
- The feature of the rule editor
- The use of free form prompts versus menus
- The supporting method(s) for consideration of uncertainty

Differences among Rule-based Shells

- Provisions for rule-base development through a rule-induction process
- Restriction on run-time versions of any expert systems developed
- Hardware requirements/restrictions

Advantages of Expert System Shells

- Simple (friendly and easy to use)
- Widely available
- Portable
- Inexpensive
- Training requirements are minimal

Disadvantages of Expert System Shells

- Control over the consultation / inferencing process is restricted
- The lack of frames may make the development of some knowledge bases awkward.
Disadvantages of Expert System Shells

- All shells are not suited to all tasks.
- It is difficult to be rigorous in one’s recommendations concerning what shell should be used for what problem.
- Most available shells provide facilities that are adequate only for small search spaces.
- “Inflexibility”

2. High-level Programming Languages

- Give a fast prototype tools so that more flexible designs can be explored and evaluated at relatively low cost in terms of time and effort.
- Provide a few more degrees of freedom than a shell to build an expert system.
- E.g., production rule languages, object-oriented programming languages, and logic programming languages.

4 General Categories of Programming Languages

- List processing language, e.g., LISP
- Logic programming languages, e.g., PROLOG
- Object-oriented languages, e.g., SMALLTALK
- Conventional programming languages, e.g., FORTRAN

Advantages and Disadvantages of Languages

- Advantage: “control”
- Disadvantage: Need a lot of efforts, compared to available shells.
3. Multiple-Paradigm Programming Environments (Hybrid Systems)

- Allow skilled programmers to experiment with novel problem-solving architectures by selecting and combining different software modules.
- Such environments provide the programmer with greater flexibility in the representation of knowledge and its procedural application.

Examples of Environments

- LOOPS combines procedure-oriented programming, rule-oriented programming, object-oriented programming, data-oriented programming.
- KEE also includes multiple inheritance of properties, message passing, better rule language, high-resolution graphic interface.
- KEE and Knowledge Craft have added a query language in the PROLOG style. It also allows procedure attached to objects

KEE

- A wide variety options that will operate with KEE.
  - KEEConnection: a KEE-based expert system that provides a bridge between SQL relational databases and KEE knowledge bases.
  - IntelliScope: this option accesses databases and formulates database queries, while displaying the analyses in graphic format.

KEE

- SimKit: an option that facilitates the development and operation of graphics-based simulations.
- PC-Host: provides the capability to develop applications on the LISP computer and then deliver them to PCs as linked to mainframe/mini-host computers.
Advantages of Environments

- They are considered to represent professional tools for serious implementation.
- They may be used for much more than just expert systems development, that is they are full-fledged environments for general AI research and development.
- Flexibility
- The ability to access mainframe resident databases.

Disadvantages of Environments

- High initial cost
- High overall costs (training, hardware support)
- Complexity and amount of training and experience required

Selecting a Software Tool

- Generality
- Selection
- Speed
- Testing