Overview of Expert Systems

A. Farzanegan
Fall 2014

Norman B. Keevil Institute of Mining Engineering
Expertise and Human Expert

• Expertise is skill or knowledge in a particular area that has been acquired by someone during practicing his/her job for a relatively long period of time.

• Expert is a person with a high degree of skill or knowledge of a certain subject or simply a person with expertise in a particular field.

• A highly experienced mill operator, a medical doctor, or an experienced auto mechanic are all experts in their job.
Definition of Expert System

• A program that uses available information, heuristics, and inference to suggest solutions to problems in a particular discipline. (The American Heritage® Dictionary of the English Language)

• A computer program that can offer intelligent advice or make intelligent decisions using rule-based programs (Collins English Dictionary)

• A computer program that imitates the functions of a human expert in a particular field, as in diagnosing a problem, by using logical operations to draw inferences from a stored body of specialized knowledge. (Random House Kernerman Webster's College Dictionary)
Advantages of Expert System

Availability
Cheaper
Reduced danger
Permanence
Multiple expertise
Explanation
Fast response
Unemotional and response at all times
Limitations ES

Some of the limitations are:

• Knowledge is not always readily available.
• It can be difficult to extract expertise from humans.
• There are frequently multiple correct assessments.
• Time pressures.
• Users have cognitive limits.
• ES works well only within a narrow domain of knowledge.
• Most experts do not have an independent means to validate results.
• Vocabulary is often limited and difficult to understand.
• Help from knowledge engineers is difficult to obtain and costly.
• Potential for lack of trust on the part of the end-users.
• Knowledge transfer is subject to biases.
Success of ESs

• Level of knowledge must be sufficiently high.
• Expertise must be available from at least one expert.
• The problem to be solved must by fuzzy.
• The problem must be narrow in scope.
• The shell must be of high quality and naturally store and manipulate the knowledge.
• The user interface must be friendly to novice users.
• The problem to be solved must be difficult and important enough to justify the development of a system.
• Knowledgeable developers with good people skills are needed.
• The impact of the ES must be considered.
• The impact should be favorable.
• Management support is needed.
Some General Applications of Expert System

- Credit granting
- Information management and retrieval
- AI and expert systems embedded in products
- Plant layout
- Hospitals and medical facilities
- Help desks and assistance
- Employee performance evaluation
- Loan analysis
- Virus detection
- Repair and maintenance
- Shipping
- Marketing
- Warehouse optimization
Knowledge Acquisition Subsystem

• A subsystem to help experts build knowledge bases. Collecting knowledge needed to solve problems and build the knowledge base continues to be the biggest bottleneck in building expert systems.

• There are many ways to collect domain knowledge (represented by if-then rules) such as referring to books, journals, interviews with human experts, query forms etc.
What is Knowledge

Expert systems are also called Knowledge Based Systems (KBSs) as they are built based on the problem-solving knowledge of a human expert to manipulate available facts to conclude new facts.

Knowledge has been defined:

- Understanding of a subject area
- Statements for mapping between facts
Types of ESs

**Rule-based ES:**
Knowledge is represented by a series of rules

**Frame-based systems:**
Knowledge is represented as a series of frames (an object-oriented approach)

**Hybrid systems:**
Involve several approaches such as fuzzy logic and neural networks

**Model-based systems:**
Structured around a model that simulates the structure and function of the system under study

**Ready-made systems:**
Utilize prepackaged software

**Real-time systems:**
Systems designed to produce a just-in-time response
Expert System Structure

- User Interface
  - Working Memory (facts)
  - Explanation Subsystem
  - Inference Engine
  - Knowledge Base
  - Knowledge Acquisition Subsystem

- User
- Human Expert
- Knowledge Engineer

MINE 432 - Industrial Automation and Robotics
Knowledge Base

Knowledge base contains facts and rules.

contains the domain knowledge which is used by the inference engine to draw conclusions. The inference engine is the generic control mechanism that applies the axiomatic knowledge to the task-specific data to arrive at some conclusion. When a user supplies facts or relevant information of query to the expert system he receives advice or expertise in response. That is given the facts it uses the inference engine which in turn uses the knowledge base to infer the solution.
Simple Facts

In a Mortgage Application Review Expert System

Mike has a net income equal to $30000

Mike is working in a publishing company for 5 years.

...

In a Mineralogy Expert System

Mineral sample is yellow

Mineral sample has a metallic luster
A Simple Rule in A Mortgage Application Expert System

If the loan is between $100,000 to $200,000

If the are no previous credits problems, and

If month net income is greater than 4x monthly loan payment, and

If down payment is 15% of total value of property, and

If net income of borrower is > $25,000, and

If employment is > 3 years at same company

Then accept the applications
Inference Engine

The inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging capabilities.

An inference engine can be developed independent of any specific domain and can be applied to various knowledge bases.
Reasoning Methods

The reasoning is processing of facts based on computerized expert knowledge to arrive to a conclusion or a number of conclusions.

There are two methods of reasoning used expert systems:

• Forward chaining
• Backward chaining
Forward Chaining

A method of reasoning that starts with the facts and works forward to the conclusions.
Backward Chaining

A method of reasoning that starts with conclusions and works backward to the supporting facts.
Explanation Subsystem

Most human experts can explain how they have reached to a particular conclusion, in other words they can explain their line of reasoning.

A subsystem that explains the system's actions. The explanation can range from how the final or intermediate solutions were arrived at to justifying the need for additional data.
User Interface

The means of communication with the user. The user interface is generally not a part of the ES technology, and was not given much attention in the past. However, it is now widely accepted that the user interface can make a critical difference in the perceived utility of a system regardless of the system's performance.
Expert Systems Development

- Determining domain requirements
- Identifying expert or panel of experts
- Construct expert system components
- System Verification & Validation
- Maintaining and reviewing system

Domain
- The area of knowledge addressed by the expert system.
CLIPS is an open source expert system shell which is available for free. Complete information, helps and manuals can be found here:

http://clipsrules.sourceforge.net/
Introduction

• Expert System Tool
• Complete environment for building rule/ object based Expert Systems
• Developed by Software Technology Branch, at NASA’s Johnson Space Centre (1985)
• Released 1986
• Developed to address shortcomings of LISP
  • Low availability of LISP on computers
  • High cost associated with LISP tools and hardware
• Poor integration with other languages
CLIPS Shell

FACT LIST
(CONTAINS DATA)

KNOWLEDGE BASE
(CONTAINS RULES)

INFERENC E ENGINE
(CONTROLS EXECUTION)
CLIPS Shell (Cont’d)

• Fact list and instance list is the global memory for data
  • Facts are data that designate relation or information such as (is-animal duck) or (this is a test) or (animals duck horse cow chicken)

• Knowledge base contains all the rules
  • Rules applied on facts in the form of IF-THEN rules

• Inference engine controls the execution of rules
  • Search in the Inference Engine uses forward-chaining and rule prioritization
CLIPS Shell (Cont’d)

• CLIPS has pattern matching abilities (the Rete Algorithm)
• Extended math functions
• Conditional tests
• Object Oriented programming (COOL: Clips Object-Oriented Language) with abstraction, Inheritance, Encapsulation, Polymorphism, Dynamic Binding
Key Features

• Designed using C programming language providing:
  • High portability
  • Low cost
  • Easy integration with external systems

• May be called from a procedural language, or may call procedural code

• Designed for integration with languages such as C, C++, FORTRAN, Java and Ada
Key Features

• Multi-paradigm language that supports rule-based, object-oriented and procedural programming.

• CLIPS supports only forward chaining rules

• Originally provided support for rule-based programming.

• Represents human knowledge in 3 ways:
  • Rules for experience based, heuristic knowledge
  • Deffunctions and generic functions for procedural knowledge
  • OOP also for procedural knowledge
Notation/Constructs

• Arithmetic Operations
  • Addition (+ 6 3 2)
  • Subtraction (- 6 3 2)
  • Multiplication (* 6 3 2)
  • Division (/ 6 3 2)
Notation/Constructs

• Facts – data or information to reason
  (person (name “John Q. Public”))
  (age 23)
  (eye-color blue)
  (hair-color black)}
Notation/Constructs

• Deftemplate
  (deftemplate person
    (slot name)
    (slot age)
    (slot eye-color)
    (slot hair-color))
Notation/Constructs

• Assert
  (assert (person (name "John Q. Public")
   (age 23)
   (eye-color blue)
   (hair-color black)))
Deffacts

(deffacts people
  (person (name "John Q. Public") (age 23) (eye-color blue) (hair-color black))
  (person (name "Jane Q. Doe") (age 26) (eye-color blue) (hair-color brown)))
• Defrule
  (deftemplate emergency (slot type))
  (deftemplate response (slot action))

  (defrule fire-emergency
   (emergency (type fire))
   =>
   (assert (response (action activate-sprinklers)))))
Notation/Constructs

General format for Defrule
(defrule <rule_name>
  <patterns>
  =>
  <actions>
Executing a CLIPS program

- Open Clips editor/ Notepad
- Add rules to knowledge base
- Add facts to global memory
- Load file
- Reset file
- Execute run command
(defrule ideal-duck-bachelor
    (bill big ?name)
    (feet wide ?name)
=>
    (printout t "The ideal duck is " ?name crlf))

(deffacts duck-assets
    (bill big Dopey)
    (bill big Dorky)
    (bill little Dicky)
    (feet wide Dopey)
    (feet narrow Dorky)
    (feet narrow Dicky))
A Few Variants of CLIPS

• FuzzyCLIPS
• AGENT CLIPS
• DYNACLIPS
• KnowExec
• CAPE
• PerlCLIPS
• wxCLIPS
• EHSIS
Questions?