

26:198:722
Expert Systems

Dr. Peter R. Gillett

Associate Professor

**Department of Accounting & Information
Systems**

Faculty of Management

Rutgers University

Introduction to Expert Systems

- What are Expert Systems
- Overview of Artificial Intelligence
- Heuristic Classification
- Expert Systems Tools

Introduction to Expert Systems

■ What are Expert Systems?

- * “An intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution” (Feigenbaum 1982)

Introduction to Expert Systems

■ What are Expert Systems?

- * “Expert Systems are a class of computer programs that can advise, analyze, categorize, communicate, consult, design, diagnose, explain, explore, forecast, form concepts, identify, interpret, justify, learn, manage, monitor, plan, present, retrieve, schedule, test, and tutor. They address problems normally thought to require human specialists for their solution”
(Michaelson, Michie, & Boulanger 1985))

Introduction to Expert Systems

■ What are Expert Systems?

- * “An Expert System is a computer system which emulates the human decision-making ability of a human expert”
(Giarratano & Riley 1989)
- * “An expert system is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice”
(Jackson 1999)

Introduction to Expert Systems

- What are Expert systems?
 - * “Expert System =
Knowledge Base + Inference Engine”
(Cowell et al. 1999)
- *Note the important separation of the knowledge from the rest of the system*
- *Is this an academic view v. an implementor’s view?*

Introduction to Expert Systems

■ What are Expert Systems?

- * The most significant practical product to emerge from 30 years of Artificial Intelligence Research?
- * Knowledge-based systems NOT Artificial Intelligence?
- * Decision support systems (DSS) v. Expert Systems (ES)?

Introduction to Expert Systems

■ What are Expert Systems?

- * Characterization by function v. characterization by technology
- * Address higher level problems
- * Address unstructured problems
- * Performance comparable to level of human expert (not necessarily emulation)
- * Use a representation of knowledge and rules
- * Can apply heuristic reasoning
- * Can supply explanations of their reasoning
- * Are highly domain specific

Introduction to Expert Systems

■ Forms of knowledge used in ES:

- * **Descriptive knowledge (facts)**
 - ◆ overdrafts are a potential source of finance
- * **Prescriptive knowledge (rules)**
 - ◆ if long-term finance is needed and only limited security is available, overdrafts are not a likely source of finance
- * **Heuristics (rules-of-thumb)**
 - ◆ bank managers in urban centers do not provide large overdrafts to farmers

■ Facts v. factoids

Introduction to Expert Systems

■ Expert Systems Components

- * User interface
- * Knowledge acquisition module
- * Knowledge base
- * Inference engine
 - ◆ control strategy
 - ◆ working memory
- * Explanation facility
- * Other interfaces

Introduction to Expert Systems

■ Knowledge engineering

* Knowledge acquisition

- ◆ Knowledge elicitation

* Knowledge representation

- ◆ Production rules

- ◆ Semantic networks

- ◆ Frames

Introduction to Expert Systems

■ Production rules

- * Dominant paradigm for applications?
- * Especially where textbook knowledge or heuristics are applied, can appear a very natural representation
- * Can pose problems when the number of rules grows excessively large
- * A method for resolving rule conflicts is needed

Introduction to Expert Systems

■ Frames

- * “Object-oriented” approach
- * Knowledge represented by structured groups of shared properties
- * Useful when there is a good deal of default or hierarchical knowledge
- * Provide for inheritance of knowledge
- * Use “Is-A” links between frames

Introduction to Expert Systems

■ Semantic Networks

- * Represent knowledge as a network of nodes
- * Useful when knowledge less hierarchical
- * Emphasize relationships rather than nodes themselves
- * Preceded frames, but are now less used

Introduction to Expert Systems

■ Inference Engines

* Control Strategies

- ◆ Forward chaining
- ◆ Backward chaining

* Search strategies

- ◆ Depth first
- ◆ Breadth first

* Conflict resolution

- ◆ RETE algorithm

Introduction to Expert Systems

■ Managing Uncertainty

- * Unreliable sources of data and information
- * Abundance of irrelevant data
- * Imprecision of language and perception
- * Lack of understanding
- * Faulty equipment
- * Conflicting sources of data
- * Hidden or unknown variables
- * Unknown or poorly specified rules or procedures
- * Data difficult or expensive to obtain

Introduction to Expert Systems

■ Managing Uncertainty

- * Classical probabilities
- * Bayesian inference
- * Certainty factors
- * Belief functions
- * Fuzzy logic, fuzzy sets, possibility theory
- * Non-monotonic logic

Introduction to Expert Systems

■ Advantages of Expert Systems

- * Preserving, reproducing, or disseminating expertise that is in short supply, is hard or expensive to obtain, or may otherwise be lost to an organization
- * Releasing true human experts from involvement in routine or straightforward decisions to concentrate on the more involved or doubtful cases, or to participate in research or training activities

Introduction to Expert Systems

■ Advantages of Expert Systems

- * Providing improved reliability by avoiding stress, fatigue or danger, responding more rapidly than humans, or combining expertise of multiple experts
- * Assisting in training of non-experts
- * Enabling non-experts to perform at a reasonable level of competence
- * Handling complex unstructured problems

Introduction to Expert Systems

■ Advantages of Expert Systems

- * Supporting a variety of decision-making styles
- * Improving timeliness by avoiding wait for human experts
- * Improving overall quality and consistency of decision making and improving decision consensus
- * Improving efficiency and saving money

Introduction to Expert Systems

■ Limitations of Expert Systems

- * Difficulties in identifying suitable human experts for development
- * Difficulties in eliciting expertise from humans, who may have problems in articulating their expertise
- * Disagreements among experts
- * Shortage of knowledge engineering professionals
- * Decisions and explanations may seem mechanical
- * Consultations time-consuming relative to perceived value

Introduction to Expert Systems

■ Pitfalls and problems

* Scope of implementation

- ◆ Willing, recognized experts available
- ◆ Scope not over-ambitious (task takes a few minutes to a few hours to complete)
- ◆ High payoff value for task
- ◆ Task requires expertise rather than common sense

* Prototyping

- ◆ Throw away the Mark 1 version!

Introduction to Expert Systems

■ Pitfalls and problems

* Development tools

◆ High level languages

➔ Lisp, Prolog

◆ Shells

➔ “Expert systems with the knowledge base removed”

– Often include additional support tools

– May be hard to find one that fits the actual problem

* Verification and validation

◆ Beware different uses of these terms!

Introduction to Expert Systems

■ Pitfalls and problems

* Gaining user acceptance

- ◆ User interface is friendly and inviting
- ◆ The system can explain the reasons for its conclusions or advice
- ◆ Shallow expertise does not give users the impression that the system is trivial
- ◆ Time taken for consultations is not disproportionate to the perceived value of the advice rendered

Introduction to Expert Systems

■ Golden-Oldies

- * (STRIPS)
- * DENDRAL
- * MYCIN (EMYCIN)
- * TEIRESIAS
- * PROSPECTOR
- * R1/XCON

Introduction to Expert Systems

■ Some business applications of ES

* Finance and banking

- ◆ Stock portfolio management
- ◆ Designing information systems for retail banks
- ◆ Asset-liability management
- ◆ Loan approvals and auditing

* Production

- ◆ Fault diagnosis in networks and equipment
- ◆ Complex bidding in the construction industry

Introduction to Expert Systems

■ Some business applications of ES

* Accounting and auditing

- ◆ Estate planning and tax advice
- ◆ Executing and analyzing internal auditing
- ◆ Charging back costs in computer time-sharing
- ◆ Auditing advanced EDP systems

* Marketing and sales

- ◆ Packaging insurance products

Introduction to Expert Systems

■ Some business applications of ES

* General management

- ◆ Advice on management by objectives
- ◆ Selection and use of forecasting techniques
- ◆ Analysis of failing companies
- ◆ Scheduling of business trips and meetings

* Human resources

- ◆ Matching personnel to jobs
- ◆ Arranging compensation packages

Introduction to Expert Systems

- Some business applications of ES
 - * Computers and Information Systems
 - ◆ Data center evaluation
 - ◆ Selection and maintenance of hardware and software

Introduction to Expert Systems

■ Accounting applications of ES

* Audit

- ◆ Audit program development
- ◆ Internal control evaluation
- ◆ Risk analysis
- ◆ Tax accrual and deferral
- ◆ Disclosure compliance
- ◆ Technical support (interpreting regulations)

* Computer support

- ◆ Software development
- ◆ Software selection
- ◆ Information transfer (e.g., file format conversion)

Introduction to Expert Systems

■ Accounting applications of ES

* Tax

- ◆ International tax planning
- ◆ Personal income tax planning
- ◆ Corporate tax planning
- ◆ Compliance checking
- ◆ Special issues (e.g., residency)

* Consulting

- ◆ Accounting expert systems for clients
- ◆ Personal financial planning

Introduction to Expert Systems

■ Artificial Intelligence

- * The part of computer science concerned with designing intelligent computer systems, that is, systems that exhibit the characteristics we associate with intelligence in human behavior - understanding language, learning, reasoning, solving problems, and so on. (Barr & Feigenbaum)

Introduction to Expert Systems

■ Artificial Intelligence

* State space search

◆ Generate-and-test

- ➔ Depth first (can be faster, may not terminate)
- ➔ Breadth first (finds the shortest solution path)

* Theorem proving

* Heuristic search

- ◆ Hill climbing
- ◆ Best-first search (A^*)

Introduction to Expert Systems

■ Artificial Intelligence

- * SHRDLU
- * Knowledge representation schemes
- * Procedural v. declarative programming
- * Separation of inference and knowledge
- * 'Neat' v. 'scruffy' debate

Introduction to Expert Systems

■ Artificial Intelligence

* Some classic AI Applications

- ◆ Robotics
- ◆ Machine learning
- ◆ Computational linguistics
- ◆ Natural language processing
- ◆ Pattern recognition
- ◆ Computer vision
- ◆ Speech recognition
- ◆ Uncertain reasoning

Introduction to Expert Systems

■ Heuristic Classification

* The Hayes-Roth classification of ES

- ◆ Interpretation systems
- ◆ Prediction systems
- ◆ Diagnosis systems
- ◆ Design systems
- ◆ Planning systems
- ◆ Monitoring systems
- ◆ Debugging systems
- ◆ Repair systems
- ◆ Instruction systems
- ◆ Control systems

Introduction to Expert Systems

■ Heuristic Classification (Clancy)

* Analysis

◆ Interpret

➤ Identify

- Monitor
- Diagnose

➤ Predict

➤ Control

* Synthesis

◆ Construct

➤ Specify

➤ Design

- Configure
- Plan

➤ Assemble

- Modify

Introduction to Expert Systems

■ Classification problem solving

* Heuristic matching

◆ Data abstraction

➔ Definitional

➔ Qualitative

➔ Generalization

◆ Heuristic matching

◆ Solution refinement

* Solutions can be enumerated in advance

* Covers a wide range of real applications

Introduction to Expert Systems

- Classification problem solving
 - * Generic tasks (Chandrasekaran)
 - ◆ Hierarchical classification
 - ◆ Hypothesis matching
 - ◆ Knowledge-directed information passing

Introduction to Expert Systems

- Control search strategies
 - * Destructive modification
 - * Pruning
 - * Least commitment
 - * Propose and revise
 - * Backtracking

Introduction to Expert Systems

■ Knowledge acquisition strategies

- * Differentiation
- * Frequency conditionalization
- * Symptom distinction
- * Symptom conditionalization
- * Path division
- * Path differentiation
- * Test differentiation
- * Test conditionalization

Introduction to Expert Systems

■ Epistemological frameworks

* Structural knowledge

- ◆ different levels of abstraction through which one can view the problem domain

* Strategic knowledge

- ◆ knowledge about how to approach a problem by choosing an ordering on methods and sub-goals which minimizes search effort

* Support knowledge

- ◆ a causal model of the domain of discourse

Introduction to Expert Systems

■ Expert system tools

- * Support prototyping
- * Shells
- * High-level programming languages
- * Multiple-paradigm programming environments (e.g., KEE, ART)

Introduction to Expert Systems

- Expert system shells
 - * Need to be matched to the task
 - * May be inflexible
 - ◆ difficult to distinguish different kinds of knowledge
 - ◆ acquiring new knowledge difficult
 - ◆ difficult to generate comprehensible explanations
- High-level languages also impose constraints

Introduction to Expert Systems

- Common implementation pitfalls
 - * Knowledge inextricably intertwined with program
 - * Fundamental concepts missing
 - * Inadequate explanation facilities
 - * Too many rules make execution slow and unwieldy

Introduction to Expert Systems

■ Tool selection

- * Only the generality necessary to solve the problem
- * Let problem characteristics determine the tool
- * Built-in explanation facilities and user interface
- * Test early
- * Sophisticated tools are expensive
- * Time consuming to perform detailed comparative evaluations of tools
- * Terminology and notation differ

Introduction to Expert Systems

■ Maxims on expert system development

- * Task should not be too hard for human experts
- * Define the task clearly
- * Decide early how you will evaluate the system
- * Work intensively with representative problems
- * Separate domain-specific from general knowledge
- * Rules that look big are
- * If rules are similar, try to unify
- * Group rules into rule sets
- * Adopt a programming style
- * Sacrifice efficiency for maintainability

Introduction to Expert Systems

- Maxims on expert system development
 - * Throw away the Mark 1 prototype