Expert Systems

- Knowledge to solve real problems
  - Emulates the role of an expert
  - useability derived from knowledge and reasoning ability of ES
    - distinguished from number crunching and repetetive data processing tasks of standard information systems
- Definition (BCS):

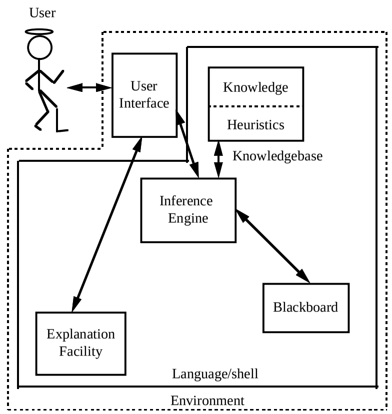
  An expert system is regarded as the embodiement within a computer of a knowledge-based component from a expert skill in such a form that the sytem can offer intelligent advice or take an intelligent decision about a processing function. A desirable additional characteristic, which many would consider fundamental, is the capability of the system to justify, on demand, its own line of reasoning in a manner directly intelligible to the enquirer. The style adopted to attain these characteristics is rule-based programming
- Feigenbaum

  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.

- Five desirable properties:
  - Reasons with domain specific knowledge
  - Uses domain specific methods
  - Performs well in its problem area
  - Explains or makes understandable what it knows and the reasons for its answers
  - Retains flexibility
- Emulates performance rather than simulates processes
- Designing an Expert System
  - Problem choice
  - Knowledge Engineering
  - Problem solving method
  - **has tended to be *ad hoc***

- Knowledge Engineering
  - Extracting the knowledge
    - heuristic (subjective)
  - Familiarity
    - Knowledge representation
    - Problem solving methods
    - Knowledge extraction methods
  - Unfamiliarity
    - Domain
      - learn a little (to a conversational level at least)
  - Knowledge Acquisition
    - Interviewing
      - Experts know more than they can tell
    - Problem solving exercises
      - results observed and abstracted into rules
    - Rapid prototyping
      - feedback and iteration
      - ease of inspection (modifications localised)
    - Bottleneck
      - labour intensive and time consuming
  - Initial implementation decisions

- Structure of a generic expert system



- Problem Solving Methods
  - how should the ES reason?
- Multiplicity of choices
  - Rule-based versus Model-based
    - If rule-based (most common)
      - forward or backward chaining?
    - If model-based
      - which ontology should be used?
  - Certain versus Uncertain
    - which type of uncertainty is it?
- Backward Chaining Rule-Based ES
  - Basic algorithm to prove a hypothesis G:
    - If G is in the current facts it is proved.
    - Otherwise, if G is askable, ask user, record their response as a new current fact, and succeed or fail according to the response.
    - Otherwise, find a rule which can be used to conclude G, and try to prove each of that rules preconditions
    - Otherwise, fail G

- A simple example
  - R1: IF *coughing* THEN ADD *smoky*
  - R2: IF *wet* AND NOT *raining* THEN ADD *burst_pipe*
  - R3: IF NOT *coughing* AND alarm_rings THEN ADD *burglar*
  - R4: IF *smoky* AND *hot* THEN ADD *fire*.
- Possibilities
  - *fire*, *burst_pipe* and *burglar*
- Can ask:
  - *hot?*, *coughing?*, *wet?*, *raining?* and *alarm?*

- Prove each hypothesis in turn:
- start with *fire* (R4)
  - set new goals *smoky* and *hot*
    - try to prove *smoky* (R1)

    Are you coughing?
      - reponse = no means hypothesis fails.
- Try *burst_pipe (R2)*
  - set new goal *wet*

    Are you wet?
      - response = no means this hypothesis fails too
- Try *burglar* (R3)
  - set new goals NOT *coughing* and *alarm_rings*
    - NOT *coughing* known

    Is there an alarm ringing?
      - response = yes

    I conclude there has been a burglary.

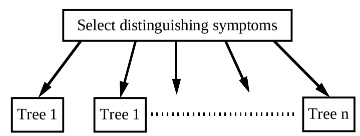
- Explanation facilities

System: Are you coughing?
User: No
System: Are you getting wet?
User: No
System: Is there an alarm ringing?
User: Why?
System: I'm trying to prove that there has been a burglary. It has been established that it is not the case that you are coughing. Therefore, if there is an alarm ringing, we can conlude that there has been a burglary.
Is there an alarm ringing?
User: Yes.
System: I conclude there has been a burglary.
User: How?
System: This follows from rule R3:
IF NOT *coughing* AND *alarm_rings* THEN ADD *burglar*
You told me that it is not the case that you coughing.
You told me that there is an alarm ringing.

- A more realistic (but still simple) example
- Diagnostic Systems as Fault Trees
  - Engineering not Medical diagnosis
    - can remove parts to test :-)
  - Effective Diagnostic Systems
    - Efficient construction
    - Reasonable maintainability
    - Adequate coverage
    - Correct answers
    - Minimum effort from the user
    - Appropriate response time
    - Good cost/benefit ratio for diagnosis
  - Low technology solution
    - construct on paper
    - easily understood
    - used by engineers to describe troubleshooting procedures
    - represent wide range of diagnostic problems
    - easily translated to ES shell

- Advantages of the Expert System
  - keeps track of how far it has got
  - handles complexity better than shuffling between pieces of paper
  - more test information can be included
  - more repair information can be included
  - more complex nodes can be included
- Advantages of the shell
  - Easy implementation
  - Reasonable maintainablity
  - Extendibilty
  - Clear order for asking questions

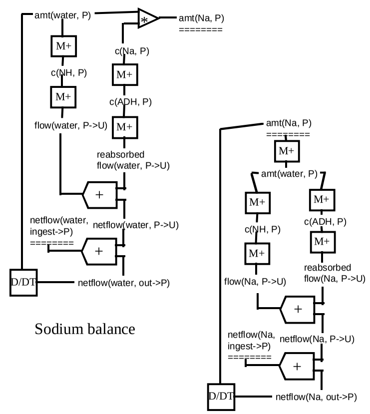
- Problems with the Fault Tree approach
  - Trees can get very large
  - Fault tree is new and empty
  - Fault tree changes frequently
  - Fault tree is really a small forest
  - Continuous diagnosis is needed



- Model-based Reasoning
- Motivations
  - Problems with RBS
    - Reasoning from First Principles
    - Dangers with “nearest approximation”
  - Modellers requirements
  - Second Generation Expert Systems
    - Use deep knowledge
    - Provide explanations of reasoning process
  - Commonsense reasoning
    - Capture how humans reason
    - Enable use of appropriate causality
  - Model reuse
    - Improved ease of ES maintenance

- Model-based Systems
  - Natural Systems
    - Physical: Fluid behaviour, Chemical reactions
    - Biological: Drug uptake, Cardiac performance, Renal operation, Photosynthesis
    - Ecological
  - Artificial Systems
    - Physical: Electrical circuits, Mechanical systems, Chemical plant
    - Economic: Housing markets, Organisations

- Example of a model



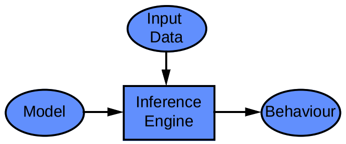
Sodium balance

Water balance

- Domains of Application
  - Modelling of ecological systems
  - Diagnosis of industrial plant
  - Training of process operators
  - Control of process plant
- Industrial Investment
  - Number of large collaborative projects involving industry (e.g. Unilever, Siemens, BG) and academia
- Eye to the future
  - Industrial rollout
  - Focus on the essence of ‘Modelling’
- Development methods
  - KADS - Expert Systems development
  - ARTIST, PRIDE - Model-based Diagnosis

What is a Model?

- Assume knowledge
    - You've all come across them.
- Physical
    - E.g. Doorlock mechanism
- Mathematical
    - Declarative Structure
        - Representation
            - Executable but distinct from inference mechanism.

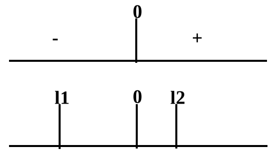


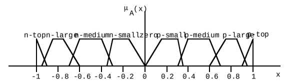
- Prediction:
    - What value will it have?
- Explanation
    - Why did it happen that way?
    - Facilitates understanding of system

- Terminology and Concepts
    - new(ish) field: proliferation of terms
    - underlying concepts basis for all QR
- Symbollically represents the important (qualitative) distinctions in a system
    - increasing, steady, decreasing
    - high, medium, low
- Scales of Measurement
    - nominal, ordinal, interval, ratio
- Qualitative versus Quantitative?

Qualitative Reasoning

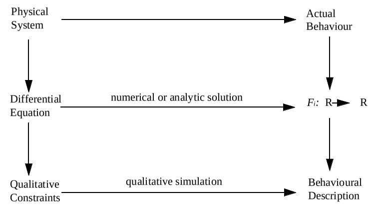
- Components of a Qualitative Model
    - Ontology (a way of looking at the world)
    - Variables (things that change)
    - Quantity space (values variables take)
    - Relations (what variables do to each other)
- Quantity Spaces





Qualitative Relations

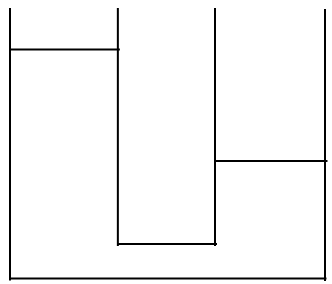
- Behavioural Abstraction



- Incompleteness
    - Not the same as "Uncertainty"
        - but is related to "Precision"
    - Known model structure (assumed)
    - Imprecise knowledge of system functional relations
- Operators
    - ADD, MULT, DERIV

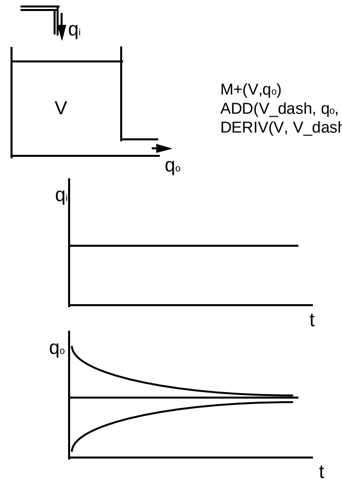
- Ontology
- A way of representing what there is in the world (closed)
- Two (main) perspectives:
  - Functional: focuses on purpose (design)
  - Behavioural: focuses on operation
- Three Behavioural Ontologies:
  - Devices (Components): pipes, tanks valves
  - Processes: heating, reacting, decomposing
  - Constraints: relations between variables

- What happens next?



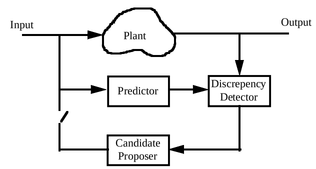
U-Tube

- The workings of the bathtub!

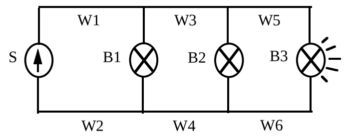


M+(V,q_o)
ADD(V_dash, q_o, q_i)
DERIV(V, V_dash)

- **Model-based Diagnosis**
  - Based on use a of “Normal” Model
  - Diagnostic Tasks
    - Fault Detection
      - *That* a fault has occurred
    - Fault Isolation
      - *Where* exactly the fault is located
    - Fault Identification
      - *How big* the fault is
    - Fault Explanation
      - How the fault *happened*
    - Fault Remediation
      - What should be done about it
  - Diagnostic Solutions
    - Diagnostic Strategist
    - Predictor
    - Candidate Proposer



- Example: Three Light Bulbs



Processes in the example

- {S, W1, W2, B1} ~ Conflict
- {S, W1, W2, W3, W4, B2} ~ Conflict
- {S, W1, W2, W3, W4, W5, W6, B3} ~ Exon.

Remove exonerated comp's from conflicts

Gives:

- {B1} ~ Conflict
- {B2} ~ Conflict

Candidates:

- {B1, B2} ~ Single diagnosis