Frame Based ES

- **what is a frame**
  - data structure with typical knowledge about a particular object
  - first used by Marvin Minsky in the 70’s
  - capture and represent knowledge
- **example**
  - boarding passes
    - same details regardless of the airline
    - carrier, name, flight, date, seat, from (airport), to (airport), boarding, gate

Frame Based ES

- **frame**
  - a natural way for structured and concise representation of knowledge
  - holds all necessary knowledge about an object or concept
  - means of organising knowledge in slots to describe various attributes
- **frames are an application of object oriented programming for expert system**

Frame Based ES

- **object oriented programming**
  - uses objects as a basis for analysis, design and implementation
- **object**
  - defined as a concept or abstraction
  - a thing with crisp boundaries and meaning for the problem at hand
- **OOP**
  - offers a natural way of representing the real world in a computer

Frame Based ES

- **define the object**
  - assign attributes describing characteristics
  - write procedures to specify behaviour
- **in Expert Systems**
  - object is referred to as a frame (thanks to Minsky)
- **Frame**
  - collection of slots - each of which is a particular attribute or operation
  - may contain default value, pointer to another frame, set of rules or procedure by which slot value is obtained

Frame Based ES

- **Slots generally include**
  - Frame Name
  - Relationship of frame to other frames
    - IBM Aptiva S35 is member of class Computer
  - Slot Value
    - symbolic, numeric, Boolean
    - can be assigned when created or during session with ES
  - Default Slot Value
    - true when no evidence to contrary available
- **Range of Slot Value**
  - does the object comply with the stereotype
  - cost of computer - £350 - £5000
- **Procedural Information**
  - executed if the slot value is changed or needed
  - two types of procedure
    - WHEN CHANGED executed when new information placed in the slot
    - WHEN NEEDED executed when information needed for problem solving
Frame Based ES

- Procedural attachments - often called demons

- Facets
  - provide an extension to slot value structure
  - used to
    - establish attribute value
    - control end user queries
    - tell inference engine how to process an attribute
  - three types - value, prompt and inference

Frame Based ES

- value facets
  - specify default and initial values

- prompt facets
  - enable user to enter attribute value during online session

- inference facets
  - allow user to stop the inference process when value of specified attribute changes

Frame Based ES

- decomposition of problem into frames, frames into slots and facets
  - depends on the problem and the judgement of knowledge engineer
  - no predefined correct representation

- instance frame
  - refers to particular object

- class frame
  - group of similar objects
  - common to include class as a slot

Frame Based ES

- frames support inheritance
  - use the class to define the essential properties
    - things that are typically true for all objects in the class
  - can declare attributes to be compound attributes
    - take only one value from the available set
  - consider class PassengerCar
    - sub class is Mazda
    - sub class is Mazda 626
    - instance would be Mazda DR-1216

Frame Based ES

- each subclass inherits attributes of parent class
  - instance will set the attribute values to required
    - eg. Colour has choice of white, blue, green, black - one will be chosen in the instance (compound attribute)
  - subclass can overwrite values inherited from parent
    - class Mazda 626 has mileage of 22 mpg
    - instance Mazda DR-1216 has value 28
  - each subclass is a further specialisation of the parent

Frame Based ES

- object relations

- generalisation
  - denotes “a-kind-of” or an “is-a” relation
  - car is a vehicle

Damien Costello, Dept of Computing & Maths, GMIT
Frame Based ES

- **aggregation**
  - denotes "a-part-of" or "part-whole" relation
  - engine is a part of car

```
Class: Car
  a part of
Class: Chassis  Class: Engine  Class: Clutch
  Superclass: car  Superclass: car  Superclass: car
```

- **association**
  - semantic relation between different otherwise unrelated classes
  - Mr Black owns a house, car & dog

```
Class: Mr Black
  owns
Class: house  Class: car  Class: dog
  Superclass: Mr Black  Superclass: Mr Black  Superclass: Mr Black
```

- **inheritance**
  - essential feature of FBES
  - common use
    - impose default features on all instance frames
  - multiple inheritance
    - applies multiple parents for hybrid types of instance frame
    - example: muscle-solar-electric vehicle inherits from electric vehicle, solar vehicle and muscle vehicle
    - parents must have unique attribute names

```
Class: Car
  a part of
Class: Chassis  Class: Engine  Class: Clutch
  Superclass: car  Superclass: car  Superclass: car
```

- **disadvantages**
  - may be difficult to distinguish between
    - essential properties
      - those an instance must have to be considered a member of the class
    - accidental properties
      - those all the instances of a class just happen to have
  - frames are a powerful tool for combining declarative and procedural knowledge
    - does leave the knowledge engineer with difficult decisions

- **inheritance means code reuse**
  - must group similar classes together and reuse common code
  - important advantage of multiple inheritance
    - conceptual simplification
    - or reducing the number of independent and specific features in an expert system
    - to achieve this, may be necessary to redesign an entire structure of the system
      - removing some attributes which are not unique

- **“typical” properties do not always work**
  - may lead to unexpected conclusions
  - may use frames to represent ostrich as a bird but it is not a typical bird
  - there are no safeguards against creating incoherent structures
    - but they are more suitable for simulation of human reasoning than conventional programming language
  - modern systems combine frames and rules for more powerful knowledge representation
Frame Based ES

• methods and demons
  – used to add actions to the frames
  – because we expect not just a storage facility but a reasoner
• method
  – procedure associated with a frame attribute
  – determine a value or execute a series of actions when value changes
  – two types - CHANGE WHEN and WHEN NEEDED

Frame Based ES

• demons
  – have IF-THEN structure in general
  – executed when IF statement changes value
  – similar to methods in this sense
• methods more appropriate for complex statements
  – demons limited to IF-THEN type
• WHEN CHANGED method
  – starts with BEGIN, ends with END
  – refer to attribute - <attribute> OF <class>

Frame Based ES

• Example
  – NEXT button on analysis form uses class called Action Data
  – contains information about loan requests
  – displayed using Value Boxes in an expert system shell
  – when NEXT button is pushed
  – value is set to TRUE, causing the WHEN CHANGED method to be activated
  – causes actions to occur

Frame Based ES

GOTO NEXT
WHEN CHANGED
BEGIN
Current Request Number = Request Number OF Request
FIND Request
LIMIT 1
WHERE Request Number OF Request > Current Request Number
FIND END
END

Frame Based ES

• what about WHEN NEEDED
  – used to obtain an attribute value only when needed
  – executed when information is required for solving a problem
  – but the value is currently undetermined
  – more on this later
• frames and rules
  – most frame based ES use a set of rules to evaluate the information in the frames

Frame Based ES

• rules are similar in structure to those in rules based systems
• rules based system
  – links rules in knowledge base with data in database
  – rule established for the rule base
• in frame systems
  – inference engine searches for a specified attribute until its value is obtained
  – goal set in either method or demon, rules used to help in evaluating a value for an attribute
### Frame Based ES

#### CLASS: Credit Evaluation

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>[E] Evaluate Credit</td>
<td>WHEN CHANGED: BEGIN PURSUE Evaluation Of Credit Evaluation END</td>
</tr>
<tr>
<td>[C] Collateral</td>
<td>Excellent: Good: Moderate:</td>
</tr>
<tr>
<td>[C] Financial Rating</td>
<td>Excellent: Good: Medium: Bad:</td>
</tr>
<tr>
<td>[C] Evaluation</td>
<td>WHEN NEEDED: BEGIN Evaluation Of Credit Evaluation IS Consult superior = TRUE END</td>
</tr>
</tbody>
</table>

**Frame Based ES**

- **user clicks on “Evaluate Credit” button**
  - ES expected to begin evaluation of credit
  - button attached to “Evaluate Credit” attribute with method
    - executed when value is changed (set to TRUE after button press)
  - PURSUE command
    - establish the value of the attribute Evaluation
    - part of the ES shell being used

**Frame Based ES**

- **how does the inference work**
  - goal set as “Give Credit”
  - find first rule which will draw this conclusion
  - same as with rules based system
  - with all valid rules and deductions, Evaluation will be set to Give Credit
  - if goal cannot be satisfied, try the next rule
- **after all goals tried and no conclusion is reached**
  - WHEN NEEDED will be called and a default conclusion will be drawn

**Frame Based ES**

- **allows setting of SEARCH ORDER**
  - if WHEN NEEDED called first every time
  - default conclusion - not good
  - specify rules first then WHEN NEEDED using a search order facet attached to each attribute

**Frame Based ES**

- **constructing a Frame System**
  - essentially the same as a rules based system
  - knowledge engineer
    - understanding of domain
    - create and test knowledge base
    - expand, test revise until complete
  - principle difference - the representation and viewing of the knowledge

**Frame Based ES**

- **in a rules based system**
  - rules represent knowledge useful for problem solving
  - adding new rules adds new knowledge
    - easily modified by changing, adding or subtracting rules
- **in a frame based system**
  - need to decide the overall hierarchical structure of the knowledge
    - classes, attributes, relationships between frames
    - architecture should provide natural description of problem
    - also allows the addition of actions to the frames through methods and demons
Frame Based ES

- typically involves the steps
  - specify the problem and define the scope of the system
  - collect relevant information
  - define the types of queries that will be answered
  - build a database which is easily modified and can be accessed by the ES
- determine classes and attributes
  - start with the conceptual, work to specifics
- define instances
  - using the data stored in the first database

Frame Based ES

- define instances
  - usually requires some type of MAKE function
  - each row could represent an instance object
  - frame receives the values of the current row
- define displays
  - screens may be linked together for continuity
  - use buttons which will be linked to methods
- define WHEN CHANGED and WHEN NEEDED methods
- define required rules
- evaluate and expand the system

Frame Based ES

- need to decide when to create instances of the classes
- two options
  - create all instances at once when user clicks a “continue” button, then remove inappropriate instances based on new preferences
  - involves using methods and demons
  - create relevant instances after all selections made for queries
  - involves using rules for decisions

Frame Based ES

- using the first option
  - when user makes a selection, the attribute is set to TRUE and the method or demon associated with that attribute is fired
  - demon can remove the inappropriate instances of the class
  - demon will only be fired if the user makes a choice (selects the corresponding button)
- using rules or methods
  - usually a preference of the designer
  - in general, methods and demons will be more effective

Frame Based ES

- Summary
  - frame is a data structure with typical knowledge about an object or concept
  - frame is used to represent knowledge
    - attributes are also called slots and are used to store values
    - may contain a default value or a pointer to another frame
  - frame may refer to group of similar objects
    - class frame describes group with common attributes - car, animal
  - frame based systems support inheritance
    - can have multiple parents
- frames communicate with each other through methods and demons
  - methods are usually more complex than the IF-THEN structure of demons
- difficult to design an effective and useful hierarchy for the knowledge