Frame-Based Expert Systems

- Introduction, or what is a frame?
- Frames as a knowledge representation technique • Inference in frame-based experts
- Methods and demons
- Interaction of frames and rules
- Summary

what is a frame?

- A frame is a data structure with typical knowledge about a particular object or concept.

- Frames were first proposed by **Marvin Minsky** in the 1970s.

Overview

what is a frame?

what is a frame?

- Each frame has its own name and a set of **attributes** associated with it. *Name*, *weight*, *height* and *age* are slots in the frame *Person*. *Model*, *processor*, *memory* and *price* are slots in the frame *Computer*. Each attribute or slot has a value attached to it.
- Frames provide a natural way for the structured and concise representation of knowledge.

Boarding pass frames

- A frame provides a means of organising knowledge in **slots** to describe various attributes and characteristics of the object.
- Frames are an application of **object oriented programming** for expert systems.

QANTAS BOARDING PASS ***Carrier***: *QANTAS AIRWAYS* ***Name***: *MR N BLACK* ***Flight***: *QF 612*
Date: *29DEC*
Seat: *23A*
From: *HOBART*
To: *MELBOURNE* ***Boarding***: *0620*
Gate: *2*

AIR NEW ZEALAND BOARDING PASS ***Carrier***: *AIR NEW ZEALAND* ***Name***: *MRS J WHITE*
Flight: *NZ 0198*
Date: *23NOV*
Seat: *27K*
From: *MELBOURNE*
To: *CHRISTCHURCH* ***Boarding***: *1815*
Gate: *4*

Frames as a knowledge representation technique

- The concept of a frame is defined by a collection of **slots**. Each slot describes a particular attribute or operation of the frame.
- Slots are used to store values. A slot may contain a default value or a pointer to another frame, a set of rules or procedure by which the slot value is obtained.

Typical information included in a slot

- **Frame name.**
- **Relationship of the frame to the other frames.** The frame *IBM Aptiva S35* might be a member of the class *Computer*, which in turn might belong to the class *Hardware*.
- **Slot value.** A slot value can be **symbolic**, **numeric** or **Boolean**. For example, the slot *Name* has symbolic values, and the slot *Age* numeric values. Slot values can be assigned when the frame is created or during a session with the expert system.
- **<u>Default slot value.</u>** The default value is taken to be ~~true when no eviden~~ce to the contrary has been found. For example, a car frame might have four wheels and a chair frame four legs as default values in the corresponding slots.
- **<u>Range of the slot value.</u>** The range of the slot ~~value determines whether~~ a particular object complies with the stereotype requirements defined by the frame. For example, the cost of a computer might be specified between $750 and $1500.
- **<u>Procedural information.</u>** A slot can have a ~~procedure attached to it, w~~hich is executed if the slot value is changed or needed.

Computer class

What are the class and instances?

- The word *frame* often has a vague meaning. The frame may refer to a particular object, for example the computer *IBM Aptiva S35*, or to a group of similar objects. To be more precise, we will use the *instance-frame* when referring to a particular object, and the *class-frame* when referring to a group of similar objects.
- A **class-frame** describes a group of objects with common attributes. *Animal*, *person*, *car* and *computer* are all class-frames.
- Each frame "knows" its class.

CLASS: ***Computer*** [Str] ***Item Code*:**
[Str] ***Model*:**
[Str] ***Processor*:**
[Str] ***Memory*:**
[Str] ***Hard Drive*:**
[Str] ***Floppy*:** [Default] [Str] ***CD-ROM*:**
[Str] ***Mouse*:**
[Str] ***Keyboard*:**
[Str] ***Power Supply*:** [Default] [Str] ***Warranty*:** [Default] [N]
***Cost*:**
[Str] ***Stock*:** [Initial]
3.5"; 1.44MB
145 Watt
3 years
In stock

Computer instances

INSTANCE: ***IBM Aptiva S35*** ***Class*:** *Computer*
[Str] ***Item Code*:** *SY7973* [Str] ***Model*:** *IBM Aptiva S35* [Str]
***Processor*:** *Pentium 233MHz* [Str] ***Memory*:** *48MB*
[Str] ***Hard Drive*:** *6.4GB*
[Str] ***Floppy*:** *3.5"; 1.44MB* [Str] ***CD-ROM*:** *24X*
[Str] ***Mouse*:** *Cordless Mouse* [Str] ***Keyboard*:** *104-key*
[Str] ***Power Supply*:** *145 Watt* [Str] ***Warranty*:** *3 years* [N]
***Cost*:** *1199.99* [Str] ***Stock*:** *In stock*

INSTANCE: ***IBM Aptiva S9C*** ***Class*:** *Computer*
[Str] ***Item Code*:** *SY7975* [Str] ***Model*:** *IBM S9C* [Str]
***Processor*:** *Pentium 200MHz* [Str] ***Memory*:** *32MB*
[Str] ***Hard Drive*:** *4.2GB*
[Str] ***Floppy*:** *3.5"; 1.44MB* [Str] ***CD-ROM*:** *16X*
[Str] ***Mouse*:** *2-button mouse* [Str] ***Keyboard*:** *104-key*
[Str] ***Power Supply*:** *145 Watt* [Str] ***Warranty*:** *3 years* [N]
***Cost*:** *999.99*
[Str] ***Stock*:** *In stock*

Class inheritance in frame-based systems

- Frame-based systems support **class inheritance**.
- The fundamental idea of inheritance is that attributes of the class-frame represent things that are *typically* true for all objects in the class. However, slots in the instance-frames can be filled with actual data uniquely specified for each instance.

Relationships among objects

- **Generalisation** denotes ***a-kind-of*** or ***is-a*** relationship between *superclass*s and its subclasses. For example, a car *is a* vehicle, or in other words, *Car* represents a subclass of the more general superclass

Vehicle. Each subclass inherits all features of the *superclass*

CLASS: *Vehicle*

***Superclass**:* *Vehicle* ***Superclass**:* *Vehicle*

CLASS: *Car*

***Superclass**:* *Vehicle*

CLASS: *Airplane*

CLASS: *Boat*

- **Aggregation** is ***a-part-of*** or ***part-whole*** relationship in which several subclasses representing *components* are associated with a superclass representing a *whole*. For example, an engine is *a part of* a car.

CLASS: *Car*

- **Association** describes some semantic relationship between different classes which are unrelated otherwise. For example, Mr Black owns a house, a car and a computer. Such classes as *House*, *Car* and *Computer* are mutually independent, but they are linked with the frame *Mr Black* through the semantic association.

CLASS: *Mr Black*

***Superclass**:* *Car* ***Superclass**:* *Car* ***Superclass**:* *Mr Black*

CLASS: *Engine* ***Superclass**:* **CLASS:** *House* ***Superclass**:* **CLASS:** *Computer*

Car *Mr Black* ***Superclass**:* *Mr Black*

CLASS: *Transmission* **CLASS:** *Car*

CLASS: *Chassis*

How does an inference engine work in a frame based system?

- In a rule-based expert system, the inference engine links the rules contained in the knowledge base with data given in the database.
- When the goal is set up, the inference engine searches the knowledge base to find a rule that has the goal in its consequent.
- If such a rule is found and its IF part matches data in the database, the rule is fired and the specified object, the goal, obtains its value. If no rules are found that can derive a value for the goal, the system queries the user to supply that value.

- Thus, the goal in a frame-based system can be established either in a method or in a demon.

Example:

Suppose we want to evaluate the

But:

- In a frame-based system, rules play an auxiliary role. Frames represent here a major source of knowledge, and both methods and demons are used to add actions to the frames.

credit request selected by the user. The expert system is expected to begin the evaluation when the user clicks the Evaluate Credit pushbutton on the input display.

This pushbutton is attached to the

attribute *Evaluate Credit* of the class *Credit Evaluation.*

pla ce

day

fataliti es

fault Sadie Hawkins

Earthquake Example cont.

Earthquake13

dama ge

magnit ude

Lower Slabovia Today

25

500,000,000 8.5

- An earthquake occurred in *value in location slot value in day slot*. There

Earthquake Summary Pattern

The Credit Evaluation class, WHEN CHANGED and WHEN NEEDED methods

CLASS: ***Credit Evaluation***

were *value in fatalities slot* fatalities and *value in damage slot* in property damage. The magnitude was *value in magnitude slot* on the Richter scale, and the fault involved

was the *value in fault slot*.

[S] ***Evaluate Credit*:** [WHEN CHANGED] [C] ***Collateral*:**
Excellent:
Good:
Moderate:
[C] ***Financial rating*:**
Excellent:
Good:
Medium:
Bad:
[C] ***Evaluation*:** [WHEN NEEDED] *Give credit*:
Deny credit:
Consult a superior:

```
WHEN CHANGED
 BEGIN
 PURSUE Evaluation OF Credit Evaluation  END
```

```
WHEN NEEDED
 BEGIN
 Evaluation OF Credit Evaluation IS Consult  superior := TRUE
 END
```

Property

The WHEN CHANGED method of the attribute Load Property

CLASS: ***Action Data***

Instances of the Class

INSTANCE: Property 1

Class: *Property*
[Str] ***Area***: *Central Suburbs* [Str] ***Suburb***: *New Town* [N] ***Price*:** 164000 [Str] ***Type***: *House* [N] ***Bedrooms*:** 3
[N] ***Bathrooms*:** 1
[Str] ***Construction***: *Weatherboard*
[Str] ***Phone*:** (03) 6226 4212 [Str] ***Pictfile***: *house01.bmp* [Str] ***Textfile***: *house01.txt* [N] ***Instance Number***: 1

INSTANCE: Property 2

Class: *Property*
[Str] ***Area***: *Central Suburbs*
[Str] ***Suburb***: *Taroona* [N] ***Price***: 150000 [Str] ***Type***: *House* [N] ***Bedrooms*:** 3
[N] ***Bathrooms*:** 1
[Str] ***Construction***: *Brick*
[Str] ***Phone*:** (03) 6226 1416
[Str] ***Pictfile***: *house02.bmp*
[Str] ***Textfile***: *house02.txt* [N] ***Instance Number***: 2

[S] ***Load Properties*:** [WHEN CHANGED]

INSTANCE: Action Data 1

Class: *Action Data*

[S] *Load Properties*: TRUE

```
WHEN CHANGED
 BEGIN
 Current Instance Number := 0
 FORGET Property
 FIND dB3 HOUSE 1
 WHEN FOUND
 Current Instance Number := Current Instance
Number + 1  MAKE Property
 WITH Area := area OF dB3 HOUSE 1
 WITH Suburb := suburb OF dB3 HOUSE 1  WITH
Price := price OF dB3 HOUSE 1  WITH Type := type
OF dB3 HOUSE 1  WITH Bedrooms := bedrooms
OF dB3 HOUSE 1  WITH Bathrooms := bathrooms
OF dB3 HOUSE 1  WITH Construction := construct
OF dB3 HOUSE 1  WITH Phone := phone OF dB3
HOUSE 1  WITH Pictfile := pictfile OF dB3 HOUSE
1  WITH Textfile := textfile OF dB3 HOUSE 1  WITH
Instance Number := Current Instance Number
FIND END
 Total Number of Instances := Current Instance
Number Goto First Property OF Action Data :=
TRUE END
```