Hybrid Intelligent Systems

Lecture 4

Knowledge representation and inference

Kinds of knowledge representations

• Logical	
– 1-order logic	symbolic
– Modal logics	symbolic
– Pseudo-physical logics	symbolic
– Description logics	symbolic
• Heuristic (empirical)	
– Rules	symbolic
– Semantic nets	graphic
– Frames	graphic, symbolic
• Neural networks	subsymbolic
• Probabilistic representation	subsymbolic

Rules

Rule - (I, A, P, A->B, F)

- I – identifier of rule (number or name)
- A – area of using of rule
- P – condition using of rule
- A – condition of rule
- B – conclusion of rule
- F – tali conditions (any comments or additional actions)
- A->B – core of rule, may be different kinds of interpretation

Kinds of interpretation of core

- Logical
  - A – logical function with &,V, not
  - If one is true then rule are executing
- Probabilistic
  - A – logical function with &,V, not
  - Rule are executing with any probability
- Threshold
  - A - set of features, which are adding with weights and rule are executing if addition is more then any threshold (as in model of neuron)

Examples.
Fragment of expert system for advice in development of expert system (in ESWin)

```
Rule 1
  EQ(Parameters.Area; Medicine)
  EQ(Parameters.Task; Diagnostics)
Do
  EQ(Knowledge representation method; Rules with Fuzzy) 90
  EQ(Knowledge representation method; Frames) 95
  EQ(Tool for Developer; ESWin) 95
EndR

Rule 2
  EQ(Parameters.Area; Computer Science)
  EQ(Parameters.Task; Monitoring)
Do
  EQ(Knowledge representation method; Rules) 100
  EQ(Tool for Developer; C++) 100
EndR
```

Kinds of inference

- Backward chaining
  - From goal to facts (as in Prolog or as in top-down method of grammatical analyzing)
- Forward chaining
  - From facts to goal (as in bottom-up method of grammatical analyzing)

Forward chaining inference

match-resolve-act cycle

The match-resolve-act cycle is the algorithm performed by a forward-chaining inference engine.

It can be expressed as follows:

loop

1. match all condition parts of condition-action rules against working memory and
   collect all the rules that match;
2. **if** more than one match, *resolve* which to use;
3. perform the action for the chosen rule

until action is STOP **or** no conditions match

Step 2 is called conflict resolution. There are a number of conflict resolution strategies.

Conflict resolution strategies

Specificity Ordering

If a rule's condition part is a superset of another, use the first rule since it is more specialized for the current task.

Rule Ordering

Choose the first rule in the text, ordered top-to-bottom.

Data Ordering

Arrange the data in a priority list. Choose the rule that applies to data that have the highest priority.

Size Ordering

Choose the rule that has the largest number of conditions.

Recency Ordering

The most recently used rule has highest priority. The least recently used rule has highest priority. The most recently used datum has highest priority.

The least recently used datum has highest priority

Context Limiting

Reduce the likelihood of conflict by separating the rules into groups, only some of which are active at any one time. Have a procedure that activates and deactivates groups.

Backward chaining inference

In backward chaining, we work back from possible conclusions of the system to the evidence, using the rules backwards. Thus backward chaining behaves in a goal-driven manner.

Backward chaining uses stack for store current goals (order of searching of tree) for possibility to select alternative path in case fail.

When backward chaining is better?

- It is needed to prove one goal, and what is goal is known preliminary
- Initial number of facts is enough large
- Number of query of facts during inference is enough small

When forward chaining is better?

- We preliminary don’t know what will be decision from several possible (its may be strongly differ between them)
- Part of time for dialog (query of facts) is relatively small in differ with part for generation of facts from other sources
- During inference some hypothesis may be generated
- It is needed to make decision in real time as answer on appearance of facts

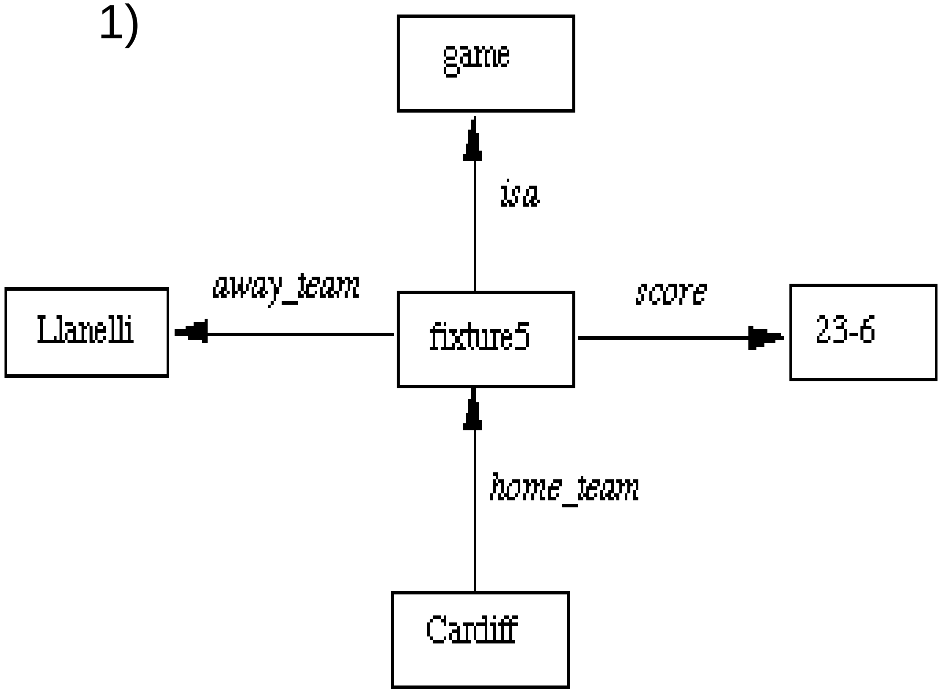
Representation of uncertainty in rules

- Facts with confidence
  - Confidence may be (0,1), (-1,1), (0,100), (0,10)
  - Are processing (during checking of condition) in compliance with formulas of fuzzy logic
- Rules with confidence
  - Confidence *Conf* is corresponding to any conclusion
  - It means that if confidence of condition is 1 (100%), then fact-conclusion is appending to base of facts with confidence *Conf*

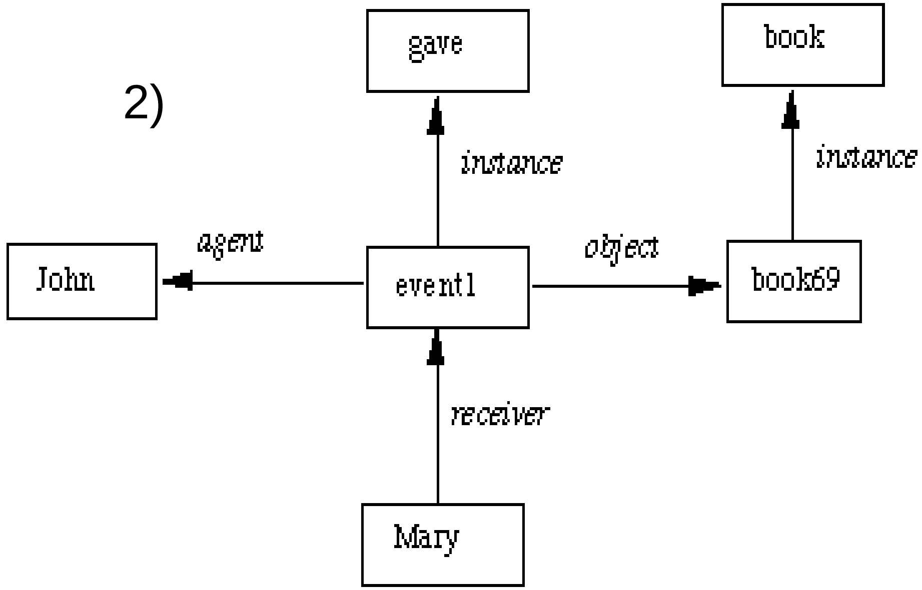
- Advantages of rules as method for knowledge representation
  - Flexibility
  - Possibility of nonmonotonic reasoning
  - Easy understandability
  - Easy appending of knowledge base
- Disadvantages
  - Low level of structuring, so it is difficult to explore of knowledge base
  - Orientation on consistent solving of task
  - Without special program support may be problems with knowledge integrity during its expanding

Semantic nets

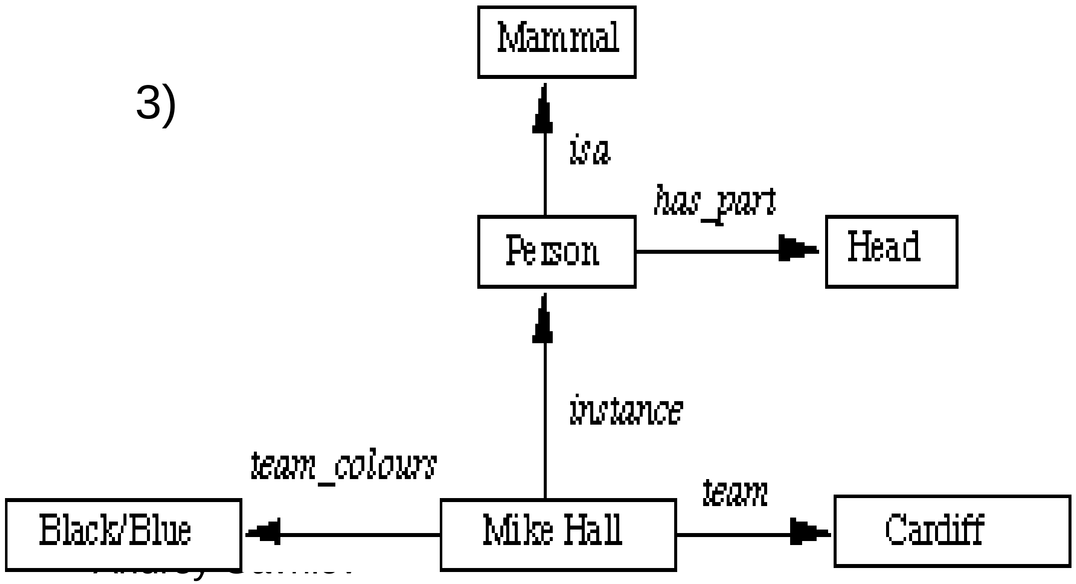
1)



2)

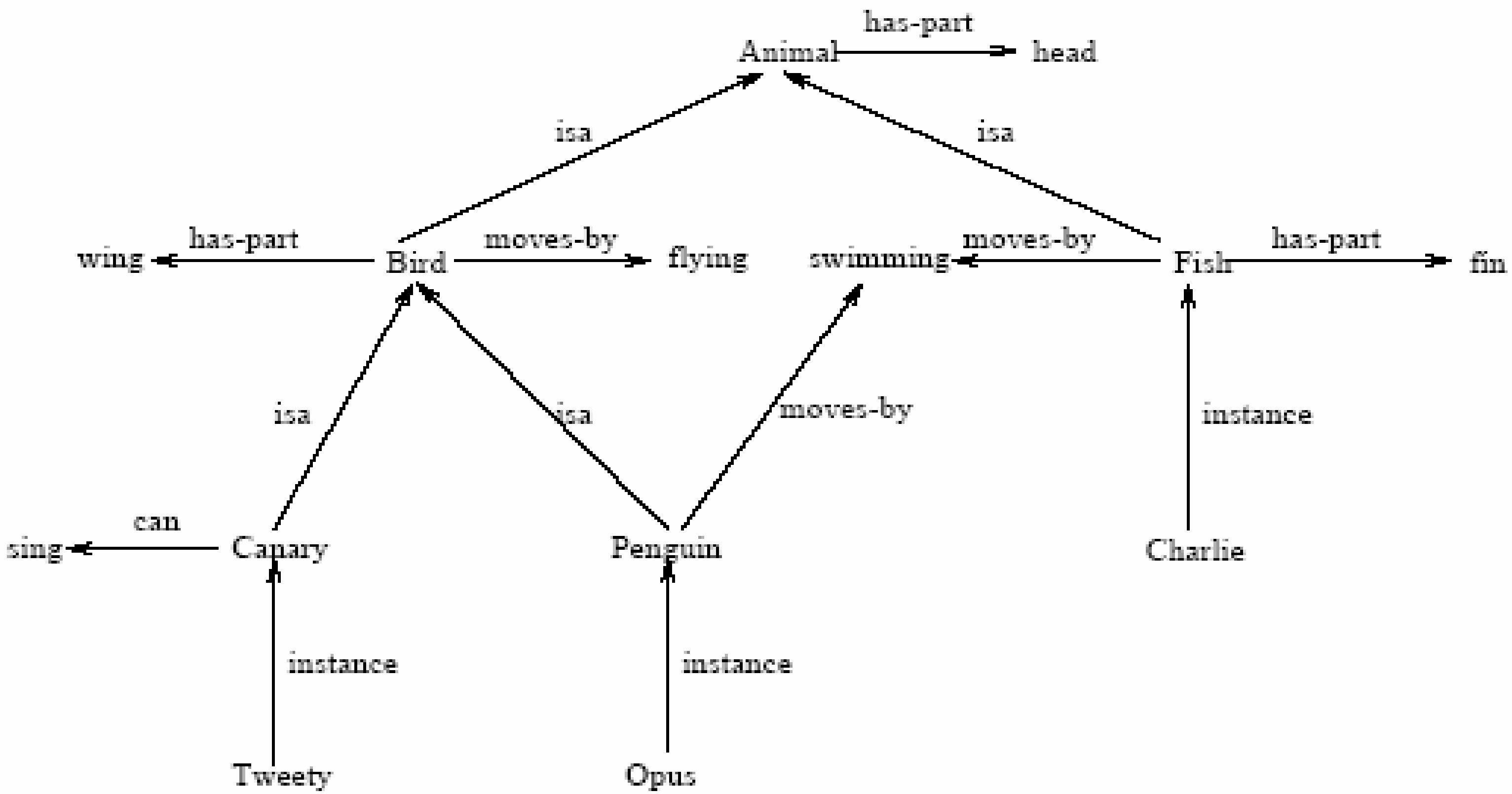


3)



isa(person, mammal),
instance(Mike-Hall, person)
team(Mike-Hall, Cardiff)

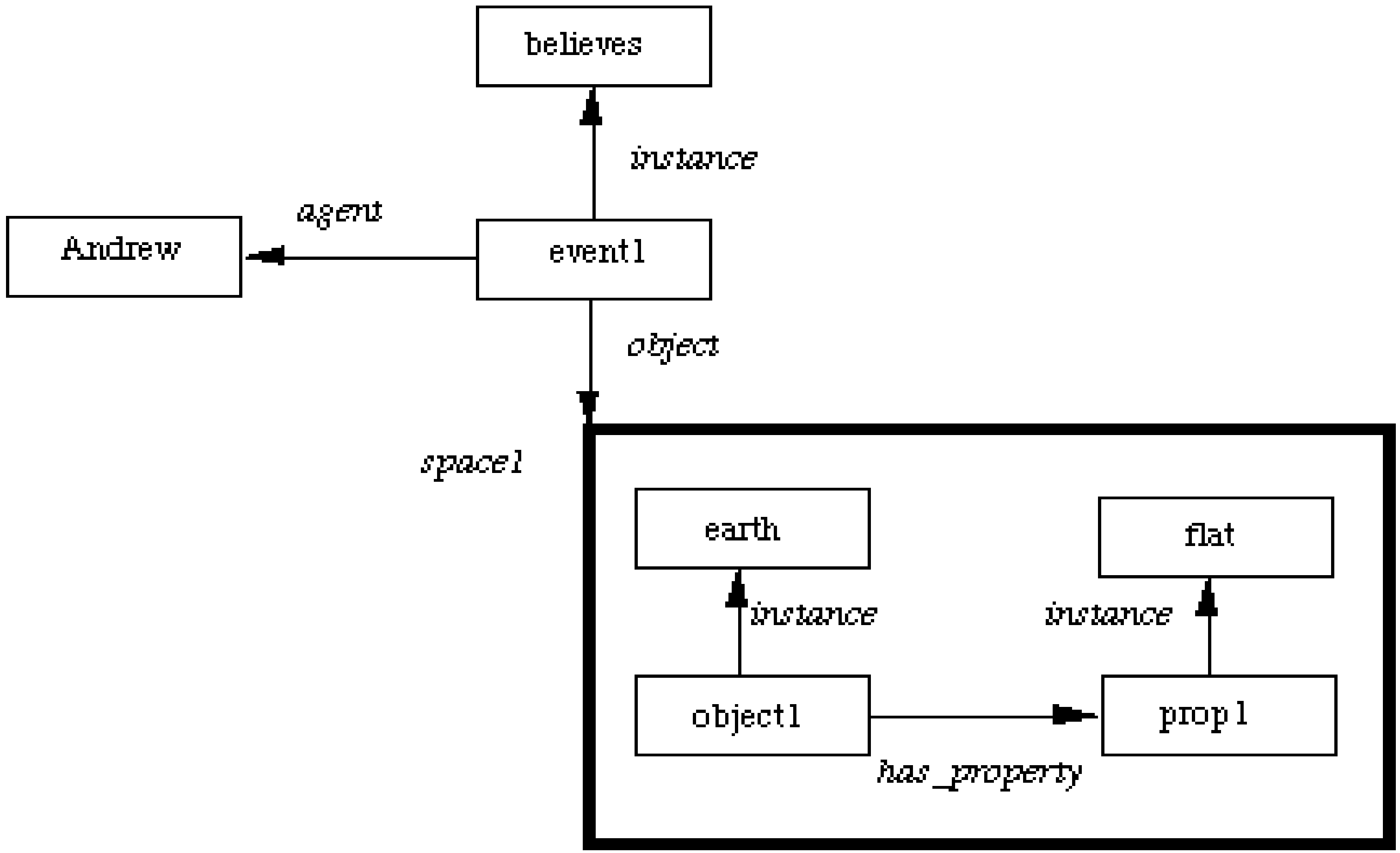
Semantic net



Extending semantic nets

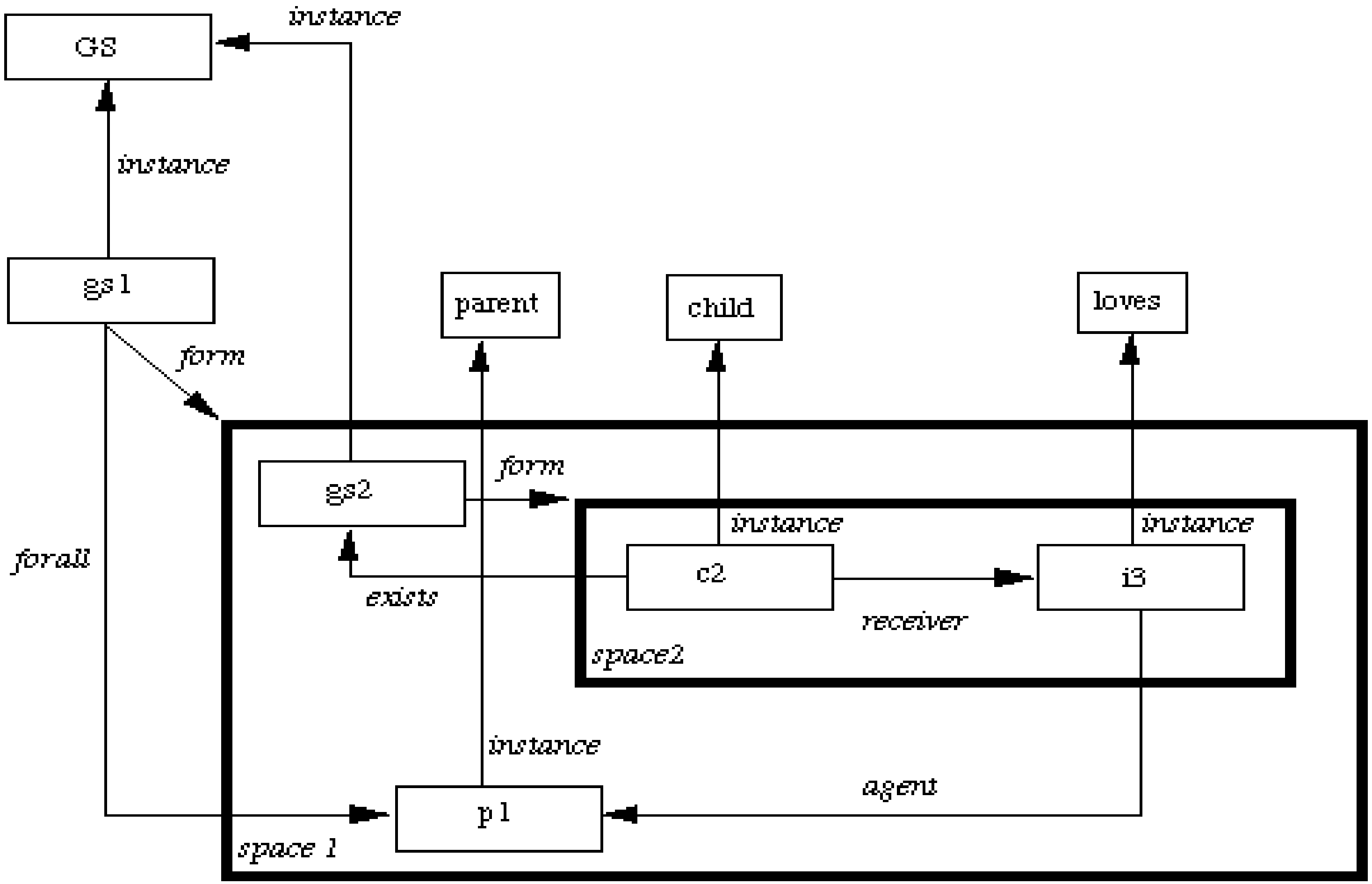
*Main idea: Break network into **spaces** which consist of groups of nodes and arcs and regard each **space** as a node.*

Andrew believes that the earth is flat



Extending semantic nets

Every parent loves their child



Inference in a Semantic Net

Basic inference mechanism: *follow links between nodes*

Two methods to do this:

Intersection search

-- the notion that *spreading activation* out of two nodes and finding their intersection finds relationships among objects.

This is achieved by assigning a special tag to each visited node.

Many advantages including entity-based organisation and fast parallel implementation.

However very structured questions need highly structured networks.

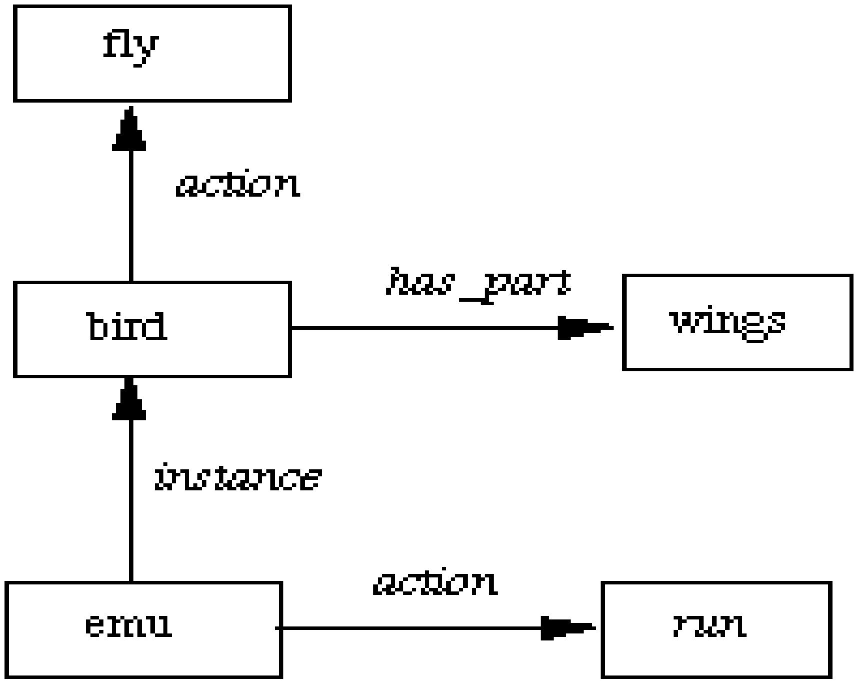
Inheritance

-- the *isa* and *instance* representation provide a mechanism to implement this.

Inheritance also provides a means of dealing with *default reasoning*. *E.g.* we could represent:

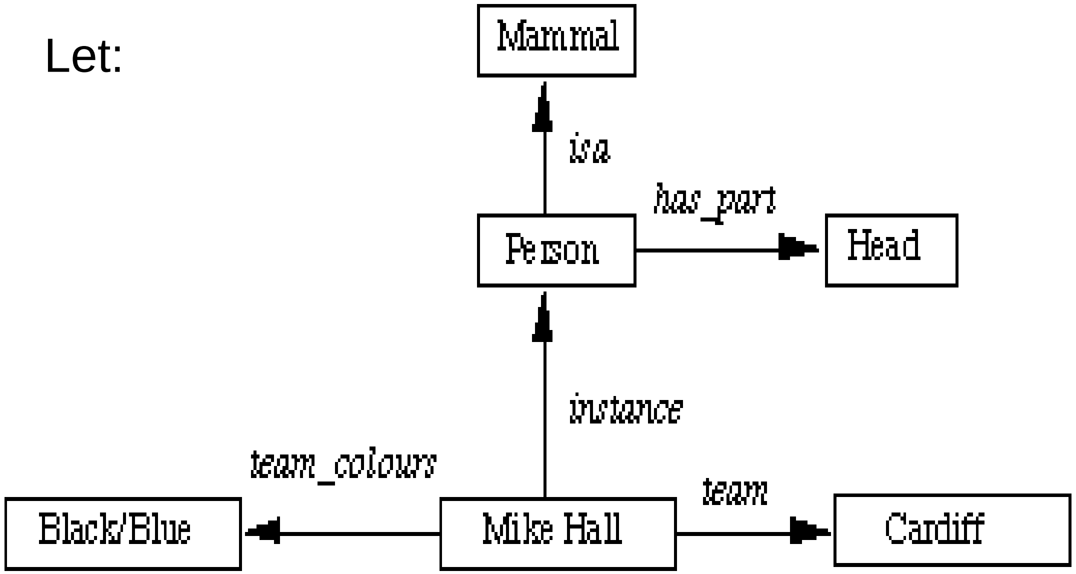
- Emus are birds.
- Typically birds fly and have wings.
- Emus run.

in the following Semantic net:

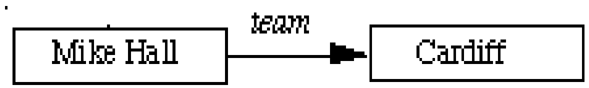


The basis of reasoning in semantic nets – the processing of query to find fragment of semantic net matching (equivalent) with query

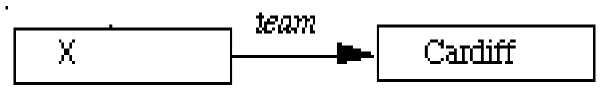
Let:



Example of query: is Mike Hall member of team Cardiff?



Example of query: Who are members of team Cardiff?



- Advantages of semantic networks as method of representation of knowledge:
  - Obviousness and understandability
  - Easy transformation to 1-order logic
- Disadvantages:
  - It is difficult to explore large semantic net
  - Not enough structuring of semantic nets
  - Capabilities of representation of procedural knowledge are absent

Frames

Author of idea – Marvin Minsky.
He invited frames for representation of scene in technical vision from different points of view.

Frames are a variant of nets that are one of the most popular ways of representing non-procedural knowledge in an expert system.

In a frame, all the information relevant to a particular concept is stored in a single complex entity, called a frame.
Superficially, frames look pretty much like record data structures.
However frames, at the very least, support inheritance.
They are often used to capture knowledge about *typical* objects or events, such as a typical bird, or a typical restaurant meal.

Example of frame “elephant”

Mammal
 subclass: Animal
 warm_blooded: yes

Elephant
 subclass: Mammal
* colour: grey
* size: large

Clyde
 instance: Elephant
 colour: pink
 owner: Fred

Nellie:
 instance: Elephant
 size: small

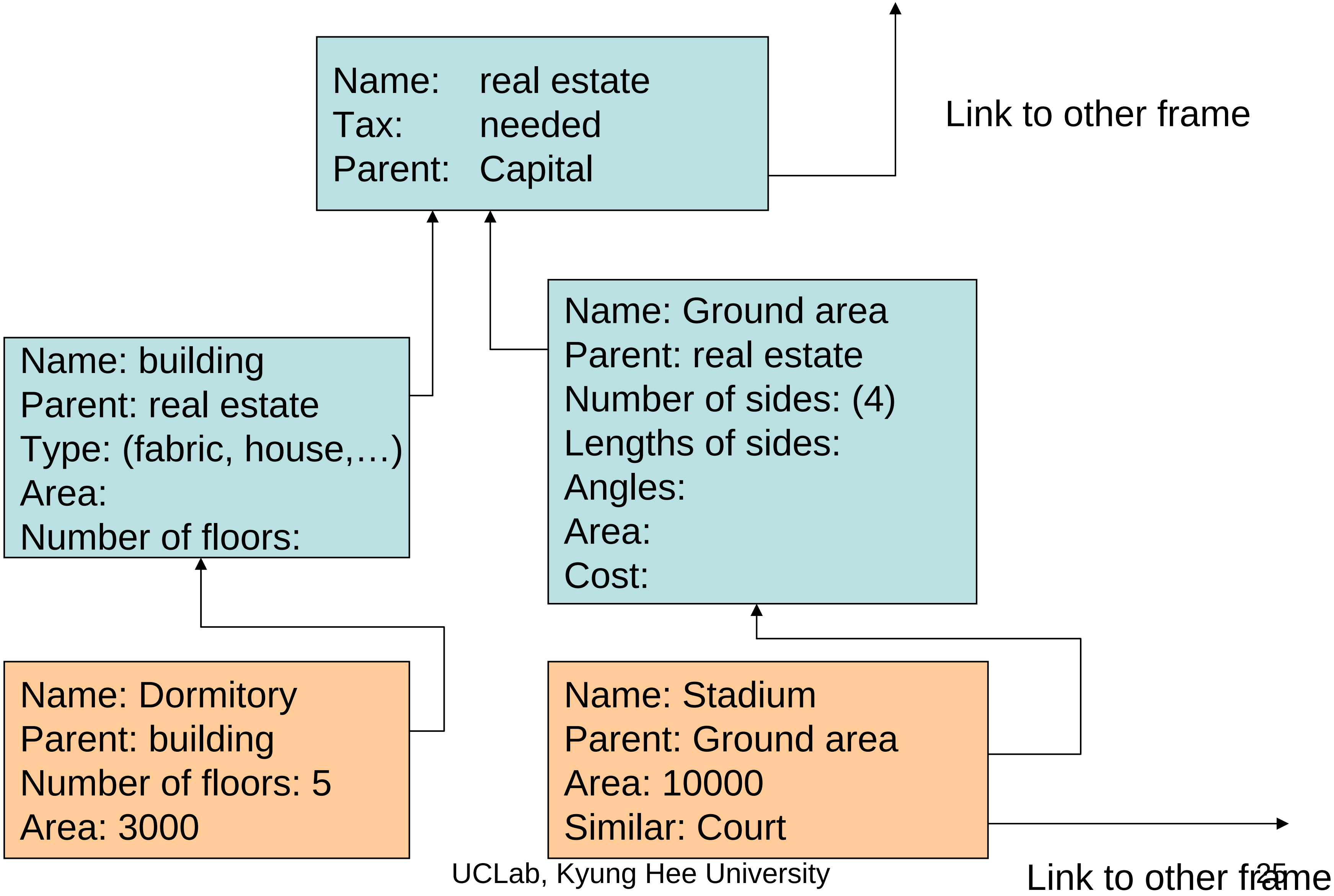
A particular frame (such as Elephant) has a number of *attributes* or *slots* such as colour and size where these slots may be filled with particular values, such as grey. We have used a ``*'' to indicate those attributes that are only true of a *typical* member of the class, and not necessarily every member.

Most frame systems will let you distinguish between typical attribute values and definite values that must be true.

Example of frame

- Name – Ground area
- Number of sides: (4)
- Lengths of sides:
- Angles:
- Area: IF_NEEDED: Calculation of area()
-                     IF_ADDED: Calculation of cost()
- Cost:

Example of net of frames:



Examples of frames in ESWin

Frame=Goal
 Parent:
 How to begin: ()
 Chances: ()
EndF

Frame=You
 Parent:
 Employment: (Unemployment; Engineer; Businessman)
 Old [How are You old ?]: (young; old)
EndF

Frame=She
 Parent: Women
 Old: (young; middle old)
EndF

Frame=Women
 Parent:
 Like flowers: (yes)
EndF

Frame=Goal
 Parent:
 Knowledge representation method: ()
 Tool for Developer: ()
EndF

Frame=Parameters
 Parent:
 Area: (Computer Science; Technology; Medicine)
 Task: (CAD; CAM; Monitoring; Diagnostics)
EndF

Methods of knowledge representation

- All methods of representations are similar
- It is relatively easy to transform from one to another (except neural networks)
- Part of them is oriented on suitable understanding of human (semantic nets, frames), part of them – on suitable of formalization and programming (different logics). Rules is such popular because ones are suitable for both goals
- Logics may be basis for building of more complex methods (many of rule-based or frame-based applied systems and tools was developed on languages of logic programming – Prolog and Lisp)

Combinations of knowledge representations

- Frames & rules – ESWin, ECLIPS, FLEX, XCON, many of medicine diagnostic expert systems (for example, CENTAUR)
- Semantic nets & rules – PROSPECTOR
- Rules & neural networks – in Hybrid Intelligent Systems (later)