

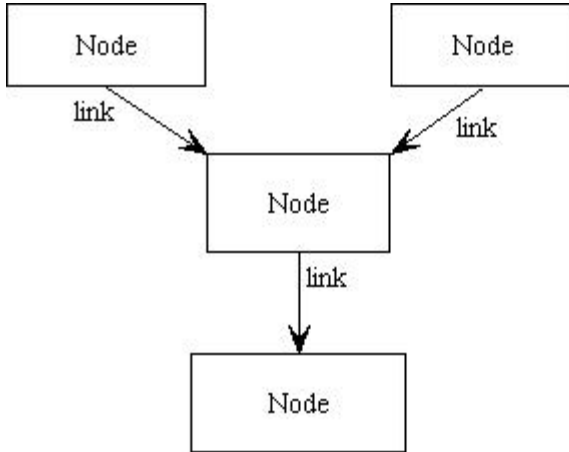
Knowledge Representation

Previous forms of knowledge structures covered:

Logic: man(Marcus)

Production Systems: if *tawny color* and *dark spots* then *giraffe*

Semantic Networks:



--With logic we receive little from memory upon input.

--With production systems we receive more

--And with semantic networks we receive even more inferences

--As seen with the above knowledge structures, if we can structure the knowledge in interesting ways, we can get more related information for free.

We mentioned 4 agent architectures at the beginning of the class:

- 1) Reflexive
- 2) State Based
- 3) Goal Based
- 4) Utility Based

--The amount of memory used by the agents increases as we go from reflexive to state based

After looking at the types of memory structures we have covered, the question becomes, are there alternate memory structures so as to receive even more inferences upon an input?

Frames

If we are given a story: The leader of Lower Slovinia declares there has been an earthquake. There was massive casualties and damage. The earthquake occurred on the Sandi Fault.

--In order for an agent to understand this story, it must have knowledge of earthquakes.

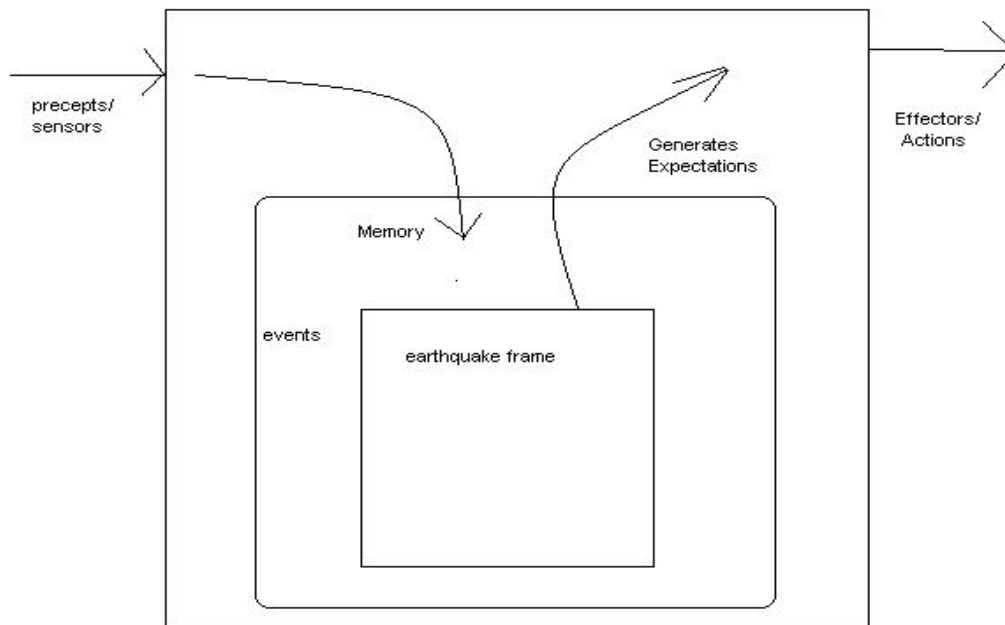
--In order to organize the knowledge about a particular thing we turn to frames.

-- A frame has a number of slots referring to given concepts of that object. It also has fillers in which the data being analyzed for the thing is matched to each concept.

For example a frame for an earthquake may look like this:

Slots	Fillers
Time	<Unspecified>
Day	Today
Place	Lower Slovinia
Fault	Sandi
Damage	Massive
Casualties	Massive

An agent-centric illustration of frames:



-- There is a question of cost when dealing with frames however: When an agent has a large number of frames, how does it get the correct frame?

This can be addressed via frame organization:

We can add an additional field called the '*a kind of*' (ako) in order to take advantage of relationships of frames between each other.

For example:

Events

Slots	Fillers
Time	...
Place	...
AKO	...

Disasters

Slots	Fillers
Damages	...
Casualties	...

AKO	<i>Event</i>
-----	--------------

Celebrations

Slots	Fillers
Entertainment	...
AKO	<i>Event</i>

Earthquakes

Slots	Fillers
Fault	...
AKO	<i>Disaster</i>

To indicate a specific instance of a frame we introduce the is-a field into a frame.

For example:

1906 San Francisco Earthquake

Slots	Fillers
...	...
Is-A	<i>Earthquake</i>

Lower Slovinia Earthquake

Slots	Fillers
Fault	Sandi
...	...
Is-A	<i>Earthquake</i>

We have looked at two important properties of frames:

- 1) Stereotypicality: Namely that certain things have concepts in common which are represented via slots.
- 2) Inheritance: In which we can share frames and information between frames through 'Is-A' and 'A kind of' relationships

We will now look at a third important property: default values

For a given frame we can introduce default values for the slots.

For example, for the disaster frame we can say the default damages and casualties are massive.

Disasters

Slots	Fillers
Damages	Massive
Casualties	Massive
AKO	<i>Event</i>

These default values are especially useful to typify a concept. For example let's look at birds:

Under logic we would say: For all x: Bird(x) \rightarrow Fly(x)

With a production system: if 'small' and 'can fly' and 'has feathers' and 'has wings' then 'bird'

The problem with these methods is there often times is an exception to the rule. An ostrich, or example, can not fly. Retrofitting the logic and production system is costly. For a frame system, however, one of the slots of the bird frame is merely overridden by the ostrich frame.

For example:

Organism

Slot	Filler
...	...
AKO	...

Animal

Slot	Filler
...	...
AKO	<i>Organism</i>

Bird

Slot	Filler
movement	flies
...	...
AKO	<i>Animal</i>

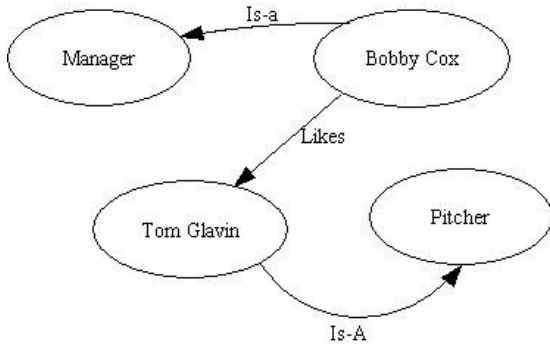
Ostrich

Slot	Filler
movement	walks
Is-A	<i>Bird</i>

Equivalence of Knowledge Systems

The types of knowledge structures we have covered are equivalent for expression purposes.

For example, here is a simple semantic network concerning the Atlanta Braves:



This can also be expressed via frames:

Bobby Cox

Slots	Fillers
Manager	Bobby Cox
Likes	<i>Tom Glavin</i>

Tom Glavin

Slots	Fillers
Pitcher	Tom Glavin
Likes	...

Or via logic:

Is-A(Manager, Bobby Cox)

Is-A(Pitcher, Tom Glavin)

Likes(Bobby Cox, Pitcher)

The different knowledge structures differ in how they are accessed for information