We’re going down a familiar road

**KR trends**
- 55-65: arbitrary data structures
- 65-75: semantic networks
- 75-85: simple frame systems
- 85-95: description logics
- 95-?: logic

**Web trends**
- 95-97: XML as arbitrary structures
- 97-98: RDF
- 98-99: RDF schema as a frame-like system
- 00-01: DAML+OIL
- 02-?: DAML-L

Only much faster!

---

**Semantic Networks**

A semantic network is a simple representation scheme which uses a graph of labeled nodes and labeled, directed arcs to encode knowledge.

- Usually used to represent static, taxonomic, concept dictionaries
- Semantic networks are typically used with a special set of accessing procedures which perform "reasoning" in e.g., inheritance of values and relationships
- Semantic networks were very popular in the 60's and 70's and enjoy a much more limited use today.
- Often much less expressive than other KR formalisms
- The **graphical depiction** associated with a semantic network is a big reason for their popularity.
Nodes and Arcs

- arcs define binary relationships which hold between objects denoted by the nodes.

Semantic Networks

- The ISA (is a) or AKO (a kind of) relation is often used to link a class and its superclass.
- And sometimes an instance and its class.
- Some links (e.g., haspart) are inherited along ISA paths.
- The semantics of a semantic net can be relatively informal or very formal.
  - often defined at the implementation level.

Reification

- Non-binary relationships can be represented by “turning the relationship into an object”
- This is an example of what logicians call “reification”
  - reify v: consider an abstract concept to be real
- We might want to represent the generic give event as a relation involving three things: a giver, a recipient and an object.
  - give(john,mary,book32)

Individuals and Classes

- Many semantic networks distinguish
  - nodes representing individuals and those representing classes
  - the “subclass” relation from the “instance-of” relation

### Diagrams

- **Nodes and Arcs**
  - Objects are connected by arcs indicating relationships.

- **Semantic Networks**
  - ISA and AKO relations link classes.

- **Reification**
  - Relationships are reified into objects.

- **Individuals and Classes**
  - Distinguishes between individuals and classes.
Inference by Inheritance

- One of the main kinds of reasoning done in a semantic net is the inheritance of values along the subclass and instance links.
- Semantic Networks differ in how they handle the case of inheriting multiple different values.
  - All possible values are inherited
  - only the "lowest" value or values are inherited
**Multiple inheritance**

- A node can have any number of superclasses that contain it, enabling a node to inherit properties from multiple "parent" nodes and their ancestors in the network.
- These rules are often used to determine inheritance in such "tangled" networks where multiple inheritance is allowed:
  - If $X \prec A \prec B$ and both A and B have property P then X inherits A's property.
  - If $X \prec A$ and $X \prec B$ but neither $A \prec B$ nor $B \prec Z$ and both A and B have property P with different and inconsistent values, then X will not inherit property P at all.

**Nixon Diamond**

- This was the classic example circa 1980.

**From Semantic Nets to Frames**

- Semantic networks morphed into Frame Representation Languages in the 70's and 80's.
- A Frame is a lot like the notion of an object in OOP, but has more meta-data.
- A **frame** has a set of **slots**.
- A **slot** represents a relation to another frame (or value).
- A slot has one or more **facets**.
- A **facet** represents some aspect of the relation.

**Frame languages**

**Typical characteristics include**

- **OO representation languages**
- **Class - subclass taxonomies**
- **Prototype descriptions of class instances**
- **Frame KR languages perform standard inferences:**
  - Inheritance of attributes, constraints and values
  - Type checking of attribute values
  - Checking number of attribute values
**Historical Perspective**

- Frame based KR systems were developed ~ 1975-1985
- [http://web.media.mit.edu/~minsky/papers/Frames/frames.html](http://web.media.mit.edu/~minsky/papers/Frames/frames.html)
- Inspired by Minsky's vision as well as OO ideas in the wind (e.g., from Simula, Smalltalk) many researchers developed new OO AI representation systems.

**Facets**

- A slot in a frame holds more than a value -- e.g., metadata, attached procedures, etc.
- Other facets might include:
  - current fillers (e.g., values)
  - default fillers
  - minimum and maximum number of fillers
  - type restriction on fillers (usually expressed as another frame object)
  - attached procedures (if-needed, if-added, if-removed)
  - salience measure
  - attached constraints or axioms
- In some systems, the slots themselves are instances of frames

**Description Logic**

- There is a family of Frame-like KR systems with a formal semantics.
  - E.g., KL-ONE, LOOM, Classic, ...
- An additional kind of inference done by these systems is automatic classification.
  - Finding the right place in a hierarchy of objects for a new description.
- Current systems take care to keep the language simple, so that all inference can be done in polynomial time (in the number of objects).
  - Ensuring tractability of inference.

---

*UMBC*
Summary

- Real knowledge representation and reasoning systems come in several major varieties.
- These differ in their intended use, degree of formal semantics, expressive power, practical considerations, features, limitations, etc.
- Some major families are
  - Logic programming languages
  - Theorem provers
  - Rule-based or production systems
  - Semantic networks
  - Frame-based representation languages
  - Databases (deductive, relational, object-oriented, etc.)
  - Constraint reasoning systems
  - Description logics