



DRONACHARYA
College of Engineering

INTELLIGENT SYSTEMS (CSE-303-F)

Section B

ISSUES IN KNOWLEDGE REPRESENTATION

ISSUES IN KNOWLEDGE REPRESENTATION

- Issues that arises while KR techniques:
 1. Important Attributes
 2. Relationships among Attributes.
 3. Choosing the granularity of Representation.
 4. Representing Sets of Objects.
 5. Finding the Right structures as Needed

Baseball knowledge

- *isa* : show class inclusion
- *instance* : show class membership

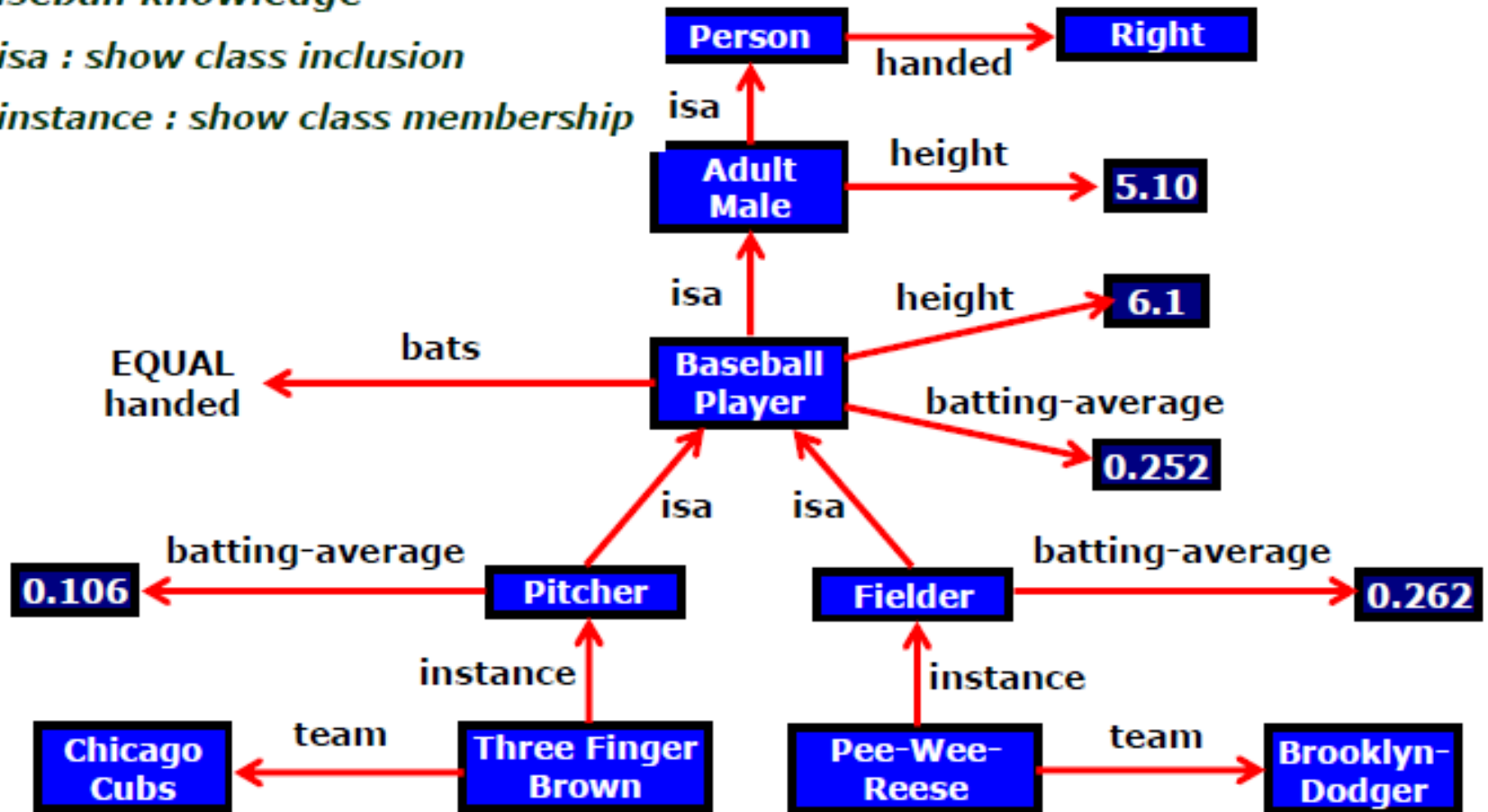


Fig. Inheritable knowledge representation (KR)

- **Important Attributes** : (Ref. Example - Fig. Inheritable KR)

There are attributes that are of general significance.

There are two attributes "instance" and "isa", that are of general importance. These attributes are important because they support *property inheritance*.

- **Relationship among Attributes** : (Ref. Example- Fig. Inheritable KR)

The attributes to describe objects are themselves entities they represent.

The relationship between the attributes of an object, independent of specific knowledge they encode, may hold properties like:

Inverses, existence in an isa hierarchy, techniques for reasoning about values and single valued attributes.

◆ **Inverses :**

This is about *consistency check*, while a value is added to one attribute.

The entities are related to each other in many different ways. The figure shows attributes (*isa, instance, and team*), each with a directed arrow, originating at the object being described and terminating either at the object or its value.

There are two ways of realizing this:

- ‡ first, represent two relationships in a *single representation*; e.g., a logical representation, *team(Pee-Wee-Reese, Brooklyn-Dodgers)*, that can be interpreted as a statement about Pee-Wee-Reese or Brooklyn-Dodger.
- ‡ second, use attributes that focus on a *single entity but use them in pairs*, one the inverse of the other; for e.g., one, *team = Brooklyn-Dodgers* , and the other, *team = Pee-Wee-Reese,*

This second approach is followed in semantic net and frame-based systems, accompanied by a knowledge acquisition tool that guarantees the consistency of inverse slot by checking, each time a value is added to one attribute then the corresponding value is added to the inverse.

◇ **Existence in an "isa" hierarchy :**

This is about *generalization-specialization*, like, classes of objects and specialized subsets of those classes. There are attributes and specialization of attributes.

Example: the attribute *"height"* is a specialization of general attribute *"physical-size"* which is, in turn, a specialization of *"physical-attribute"*. These generalization-specialization relationships for attributes are important because they support inheritance.

◇ **Techniques for reasoning about values :**

This is about *reasoning values of attributes* not given explicitly.

Several kinds of information are used in reasoning, like,

height : must be in a unit of length,

age : of person can not be greater than the age of person's parents.

◇ **Single valued attributes :**

This is about a *specific attribute* that is guaranteed to take a unique value.

Example : A baseball player can at time have only a single height and be a member of only one team. KR systems take different approaches to provide support for single valued attributes.

● Choosing Granularity

What level should the knowledge be represented and what are the primitives ?

- Should there be a small number or should there be a large number of low-level primitives or High-level facts.
- High-level facts may not be adequate for inference while Low-level primitives may require a lot of storage.

Example of Granularity :

- Suppose we are interested in following facts
John spotted Sue.
- This could be represented as
Spotted (agent(John), object (Sue))
- Such a representation would make it easy to answer questions such are
Who spotted Sue ?
- Suppose we want to know
Did John see Sue ?
- Given only one fact, we cannot discover that answer.
- We can add other facts, such as
Spotted (x , y) → saw (x , y)
- We can now infer the answer to the question.



Set of Objects

Certain properties of objects that are true as member of a set but not as individual;

Example : Consider the assertion made in the sentences

"there are more *sheep* than *people* in Australia", and

"*English* speakers can be found all over the world."

To describe these facts, the only way is to attach assertion to the sets representing *people*, *sheep*, and *English*.

The reason to represent sets of objects is :

If a property is true for all or most elements of a set, then it is more efficient to associate it once with the set rather than to associate it explicitly with every elements of the set .

This is done in different ways :

- in logical representation through the use of *universal quantifier*, and
- in hierarchical structure where node represent sets, the *inheritance propagate* set level assertion down to individual.

Example: assert *large (elephant)*;

Remember to make clear distinction between,

- whether we are asserting some property of the set itself, means, *the set of elephants is large*, or
- asserting some property that holds for individual elements of the set , means, *any thing that is an elephant is large*.

Finding Right Structure

Access to right structure for describing a particular situation.

It requires, selecting an initial structure and then revising the choice.

While doing so, it is necessary to solve following problems :

- how to perform an initial selection of the most appropriate structure.
- how to fill in appropriate details from the current situations.
- how to find a better structure if the one chosen initially turns out not to be appropriate.
- what to do if none of the available structures is appropriate.
- when to create and remember a new structure.

There is no good, general purpose method for solving all these problems.

Some knowledge representation techniques solve some of them.