Artificial Intelligence
Knowledge Representation Forms

Dr. Hrishikesh Bhaumik
Associate Professor
Department of Information Technology
RCC Institute of Information Technology

Metaclasses

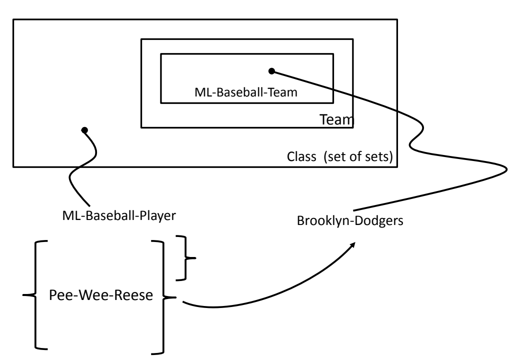
- These are special classes whose **elements are themselves classes**.
- Thus a class is an **element of some class** or may be a subclass of **one or more classes**.
- A class inherits properties from the class of which it is an instance and also passes inheritable properties down from its **super classes to its instances**.

Representing the class of All Teams as a Metaclass

Class
  instance : class
  isa : class
  *cardinality :
Team
  instance : class
  isa : class
  Cardinality: {the number of teams that exit}
  *team-size: {each team has a size}
Cricket-Team
  instance : class
  isa: Team
  cardinality: 10 {the number of Cricket teams that exist}
  *team-size: 14
  *manager:
  *coach:
Indian Cricket – Team
  instance: class
  isa : Cricket-Team
  team-size : 14

Sachin-Tendulkar
instance: Batsman
Uniform-color: Blue
Batting avg: 50.29

Classes and Metaclasses



Class Relationships

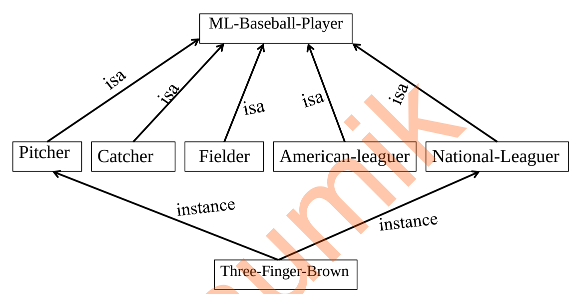
Mutually-disjoint-with:

This relationship relates a class to one or more classes that are guaranteed to have **no elements in common** with it.

Is-covered-by:

It **relates** a **class to a set of classes**, the union of which is equal to it. If a class is-covered-by a **Set S** of **mutually disjoint classes**, then S is called a **partition of the class**.

Example Of <u>Mutually-disjoint-with</u> and <u>Is-covered-by</u>

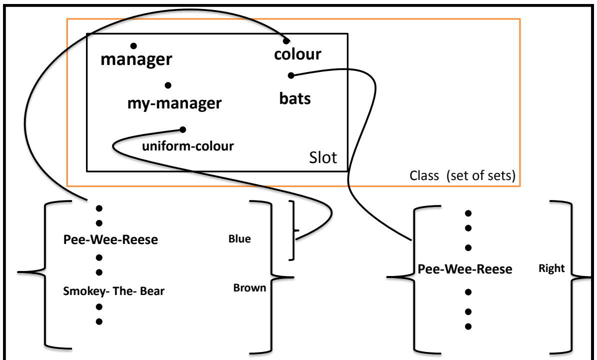


ML-Baseball-Player	
is-covered-by :	{Pitcher, Catcher, Fielder}, {American-Leaguer, National-Leaguer}
Picher	
isa :	ML-Baseball-Player
mutually-disjoint-with:	{Catcher, Fielder}
Catcher	
isa:	ML-Baseball-Player
mutually-disjoint-with:	{Pitcher, Fielder}
Fielder	
isa:	ML-Baseball-Player
mutually-disjoint-with:	{Pitcher, Catcher}
American-Leaguer	
isa:	ML-Baseball-Player
mutually-disjoint-with:	{National-Leaguer}
National-Leaguer	
isa:	ML-Baseball-Player
mutually-disjoint-with:	{American-Leaguer}
Three-Finger-Brown	
instance:	Pitcher
instance:	National-Leaguer

Representing Relationships among classes

Slots

A slot is a **relation** which maps from elements of its **domain**, i.e. the classes for which it makes sense to the elements of its range i.e., its possible values. Since each slot is a relation, it has a **domain** and a **range**.



Representing Slots as Frames

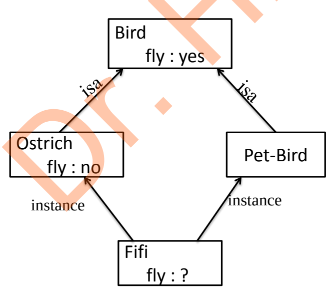
Slot

isa: Class
instance: Class
*domain:
*range:
*range-constraint:
*definition:
*default:
*transfer-though:
*to-compute:
*inverse:
*single-valued:

manager
instance: Slot
domain: Indian-Cricket-Team
Range: Person
Range-constraint: λx(Cricket-experience x manager)
default:
inverse : manager-of
Single-valued : TRUE

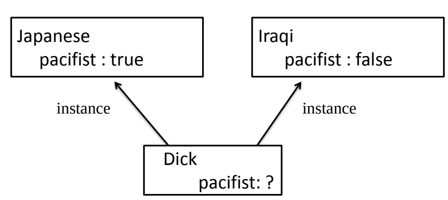
Representing Slots as Frames

Tangled Hierarchies



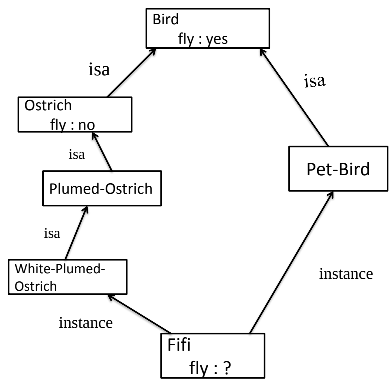
(a)

Tangled Hierarchies



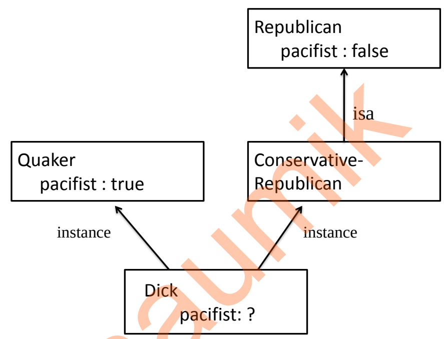
(b)

Tangled Hierarchies



(a)

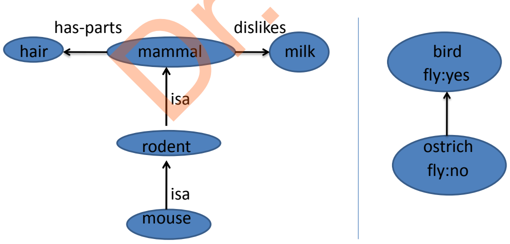
Tangled Hierarchies



(b)

Property Inheritance

It is a form of inheritance in which the nodes which are **members or subsets** of other nodes may inherit properties from their higher level ancestor nodes. When an object does not inherit certain properties, it would be assigned values of its own which override any inherited ones.



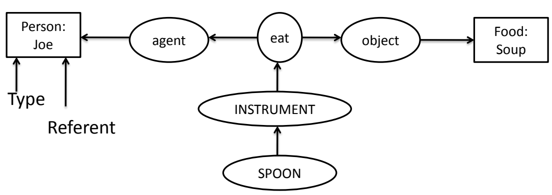
Conceptual Graphics

It is a **graphical portrayal** of a mental perception which consists of **basic or primitive concepts** and the **relationships** that exist between the concepts.

A **single conceptual graph (CG)** is roughly equivalent to a **graphical diagram** of a **natural language sentence** where words are depicted as **concepts and relationships**. A **CG** may be regarded as a **formal building block** for associated networks which are **linked** together to form a **complex knowledge structure.**

Conceptual Graphics: Example

" Joe is eating soup with a spoon"



Modifying and combining graphs

Operators required are:

COPY: Produces a duplicate copy of a **CG**.

RESTRICT: Modifies a graph by replacing a type label of a concept with a subtype or a specialization from generic to individual by inserting a referent of the same concept type.

Join :

Combines two identical graphs C1 & C2 by attaching all relation arcs from C2 to C1 and then erasing C2.

Simplify :

Eliminates one of two identical relations in conceptual graph when all connecting arcs are also the same.

Conceptual Dependencies:

It is based on the use of a limited number of primitive concepts and rules of formation to represent any natural language statement. It states that different sentences which have the some meaning should have the same unique CD representation and representations for any sentence should be unambiguous. Thus CD is a mechanism for representing and reasoning about events.

Scripts:

A script is a structure that describes a **stereo-typed sequence of events** in a particular context. It **consists of a set of slots**. Associated with each slot is some information about what kinds of values it may contain and a default value to be used, if no other information is available.

Scripts are constructed using **basic primitive concepts** and rules of formation similar to conceptual graphs. A script may represent **commonly occurring experience** such as going to movies, shopping in a supermarket, eating in a restaurant etc.

ATRANS: Transfer of an abstract entity
PTRANS: Physical transfer from one location to another.
Primitive actions.

SCRIPTNAME	: food market
TRACK	: supermarket
ROLES	: shopper daily attendant checkout clerk sacking clerk other shoppers
ENTRY CONDITIONS	: shopper needs groceries food market open
PROPS	: shopping cart display aisles market items checkout stands cashier money

contd...

SCENE 1	: Enter Market shopper PTRANS shopper into market shopper PTRANS shopping-carts to shopper
SCENE 2	: shop for Items shopper MOVE shopper through aisles shopper ATTEND eyes to display items shopper PTRANS items to shopping cart.
SCENE 3	: Check Out shopper MOVE shopper to checkout stand shopper WAIT shopper turn shopper ATTEND eyes to charges shopper ATRANS money to cashier sacker ATRANS bags to shopper.
SCENE 4	: Exit Market
RESULTS	: shopper has less money shopper has grocery items market has less grocery items. market has more money.

Reasoning using scripts

Scripts contain **knowledge** that people use for common every day activities. Thus they provide an **expected scenario** for a given **situation**.

Reasoning in a script begins with **creation of a partially filled script** named to meet the current situation. Then a known script which **matches the current solution** is recalled from **memory**. The script name, pre-conditions, other keywords provide index values with which to search the appropriate script. Inference is accomplished by filling in slots with inherited and default values that satisfy certain conditions.

Dempster-Shafer Theory of Evidence

It considers **sets of properties** and assigns to each of them an **interval** [**belief**, **plausibility**] within which the degree of belief for each proposition must lie. This **belief** measure, denoted by **bel**, ranges from **zero**, indicating **no evidence of support** for a set of propositions, to **one** denoting **certainty**. The **plausibility** of a proposition p, **pl(p)** is defined as:

pl(p)=1-bel(not(p))

The **plausibility** also ranges between 0 and 1 and reflects **how evidence of not (p)** relates to the **possibility for belief in p**.

If there are two belief functions m_1 & m_2 and X be the subset of θ (universal set) to which m_1 assigns a nonzero value and Y be the corresponding set for m_2 then, m_3 denotes the combination of m_1 & m_2 .

Such that:

$$m_3(Z) = \frac{\sum_{X\cap Y=Z} m_1(X).m_2(Y)}{1-\sum_{X\cap Y=\emptyset} m_1(X).m_2(Y)}$$

Dempster-Shafer Theory of Evidence

Example:

A medical diagnosis problem where

θ= {All, Flu, Cold, Pnew}

We consider a Probability Density Function m which is defined for not only the elements of θ but for all subsets of it.

We define m as {θ} (1.0)

After an evidence let the updates value of m be m_1:

{Flu, Cold, Pnew} (0.6)

{θ} (0.4)

Evidence → running nose

m_2: {All, Flu, Cold} (0.8)

{θ} (0.2)

contd..

Dempster-Shafer Theory of Evidence: Example

Combining m_1 & m_2 , we get m_3, which we derive using numerator of the comb rule: m_3:

	{A,F,C}	(0.8)	θ	(0.2)
{F,C,P}(0.6)	{F,C}	(0.48)	{F,C,P}	(0.12)
{θ} (0.4)	{A, F,C}	(0.32)	{θ}	(0.08)

Evidence: m_4 : patient goes on trip

{A} (0.9)

θ (0.1)

	{A}	(0.9)	θ	(0.1)
{F,C} (0.48)	ϕ	(0.432)	{F,C}	(0.048)
{A,F,C} (0.32)	{A, F,C}	(0.288)	{A, F,C}	(0.032)
{F,C,P}(0.12)	ϕ	(0.108)	{F,C,P}	(0.012)
θ (0.08)	{A}	(0.072)	θ	(0.008)

contd..

Dempster-Shafer Theory of Evidence: Example

Total belief associated with ϕ is 0.54

∴ we need to scale the values by a factor (1-0.54) =0.46

∴ m5:

{ F,C}	(0.104)
{A,F,C}	(0.696)
{F,C,P}	(0.026)
{A}	(0.157)
θ	(0.017)
	1.0