A multitude of Relations

- In the real world entities are related to multiple other entities.
- The question is how does one capture such relations?
- Consider the use of verbs to describe action i.e: here is an incident of kids fighting.

Abc hit Xyz with a stick yesterday afternoon in a park.

Hit(abc, xyz, stick, yesterday_afternoon, park)

But the semantics is completely arbitrary and user would have to keep track of what parameter represents what.

Breaking down complex Sentences:

The modern trend is to breakdown complex sentences into binary relations:

We can create an abstract type called event,

and describe the rest of the relations for an instance of that event.

Instance (Hitting_event, e45) : e45 is a (type_of) Hitting_event.

Actor (e45, abc) : the actor of e45 is abc

Object (e45, xyz) : the object of e45 is xyz

Date (e45, March_21) : the date of e45 is Mar_21

Loc (e45, park) : The location of e45 is the park

Instrument (e45, stick) : The instrument of e45 is the stick

Noticed, that each property is a named property we can flexibly choose what properties to talk about!

Every sentence uses a binary predicate
The Predicate (Subject, Object) relation can be expressed as a tipple

<sub, predicate, obj> or <sub, property, value>

So the first one 'Instance' is like;

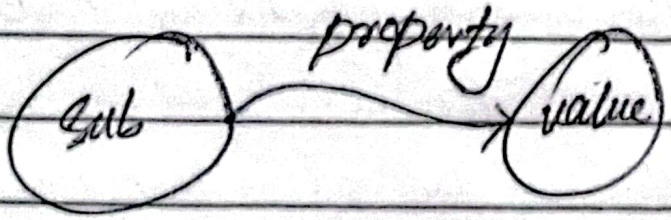
(Hitting_event, Instance, e45):
(e45, Actor, abc) :
(e45, Object, xyz) :

(e45, Date, nov-21):
(e45, Loc, bank):
(e45, Instrument, stick):

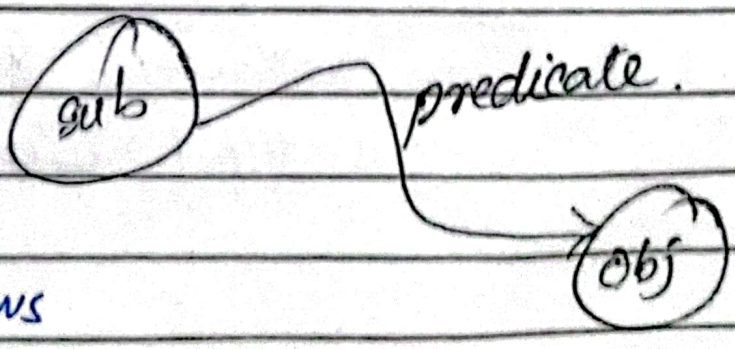
* Each triple says something about the subject, the first element.
* The second element characterize the property.
* The third element is the value of the property.

Triple Store

Since each triple is a binary relation b/w two entities we can depict the relation as an edge in a graph.



or



each sentence is stored as an edge b/w two identifiers

* the subject is a node in the graph
* The property is a labelled edge emanating from

Semantic Nets → KG

A semantic network is a graphical model in which nodes reasoning concepts are connected with labelled edges representing relations.

Kn Gr

A KG, also known as a semantic network, represents a network of real-world — i-e objects, events, situations, or concepts — and illustrates the relationship b/w them.

This information is usually stored in a graph database & visualized as a graph structure, ~~prometing~~ prompting the term knowledge "graph".

* A KG is made up of three main components: nodes, edges, & labels.
* Any object, place, or person can be a node.
* An edge defines the relationship b/w the nodes

Ontologies are also frequently mentioned in the context of KG.

Date

SEMANTIC NETS :-

The main idea behind Semantic nets is that the meaning of a concept comes from the ways in which it is connected.

Date

* Scripts & Frames allow us to represent the abstraction hierarchies as well as aggregation hierarchies explicitly.

Frames and Scripts

The idea of semantic Nets & KGs was to link concepts together with labelled edges to enable finding related elements quickly.

The idea of FRAMES; first espoused by Marvin Minsky is to take this physical proximity one level closer. To reify aggregations & encapsulate the related elements into one knowledge structure.

FRAMES:

* A Frame is a data structure that encapsulates a number of attributes of an entity.
* It is like a script in that it packs related "sentences" together in one place.
* It is different from a script in that it is a structured representation with a collection of attribute-value or slot-filler pairs rather than chains of events conceptualizations.

Signature No.

Date

- The Slot is a named attributes the filler can be anything including another frame.

* It is also different from scripts because there are explicit links to other frames.
  - ... that captures both abstractions & aggregations hierarchies.

* It is like a set of triples organized together with the "subject" being the header with properties nested together inside.

* the structure of the frames mirrors the structure of 'objects' they represent.
  - the basis for OOP.

* A frame is represented by

```
(Frame-name
        (slots-name-1 , filler-1)
        (slot-name-2 , filler-2)
        :
        (slot-name-n , filler-n))
```

Signature

* Frames may be Generic or individual frames
* Generic frames:
  represent a frame type or a class
* Individual frames:
  - They are instances of generic frames
  - They are like UNITs in RBS.

SLOTS in Frames

The slots in Frames can be of different types
- they may have procedures attached to them

<: TotalCost [IF-NEEDED ... procedure to compute total cost] >

<: TravelStep [IF-ADDED ... create frame for next leg if not at home] >

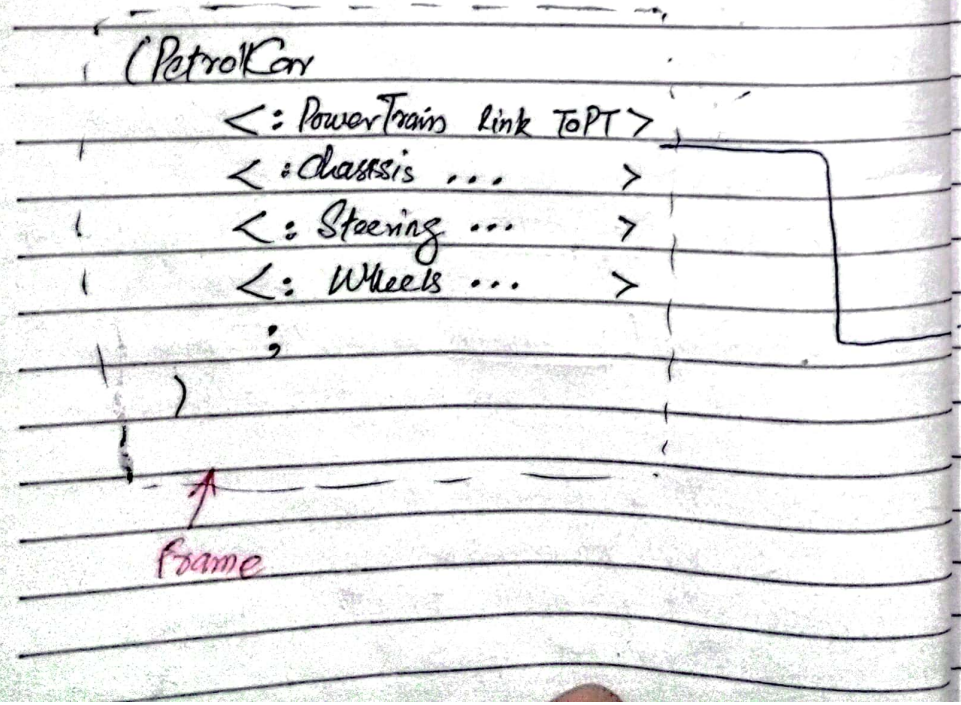
- Slots with a nominal value
  <: Age In Years 25 >
- Slots with pointers to other frames
  <: Owner socrates >

Date

Aggregation

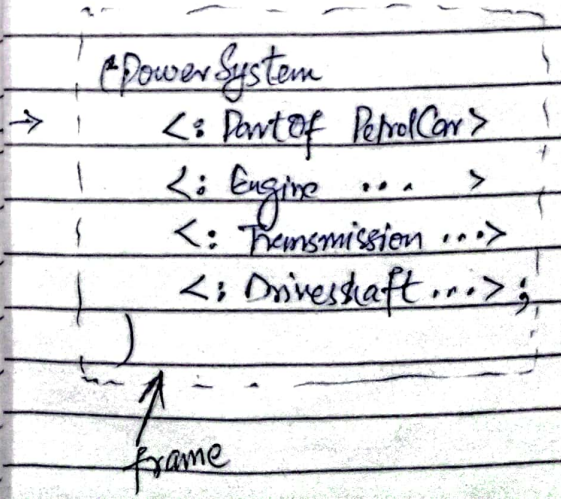
- A frame can organize information into a packaging or aggregating hierarchy
- there are usually no special keyword slot like HasPart.

- However, the value of a slot may be a pointer to a sub-part frame, which could have a PartOf slot.



Date

- As individual elements come together to form a large entity, hierarchy and aggregation happens naturally.
  - ant colonies
  - animal bodies
  - flocks of birds
- Aggregation is natural when the aggregate acquires agency.
  - most life forms have some form of ~~symbols~~ simple hierarchy.

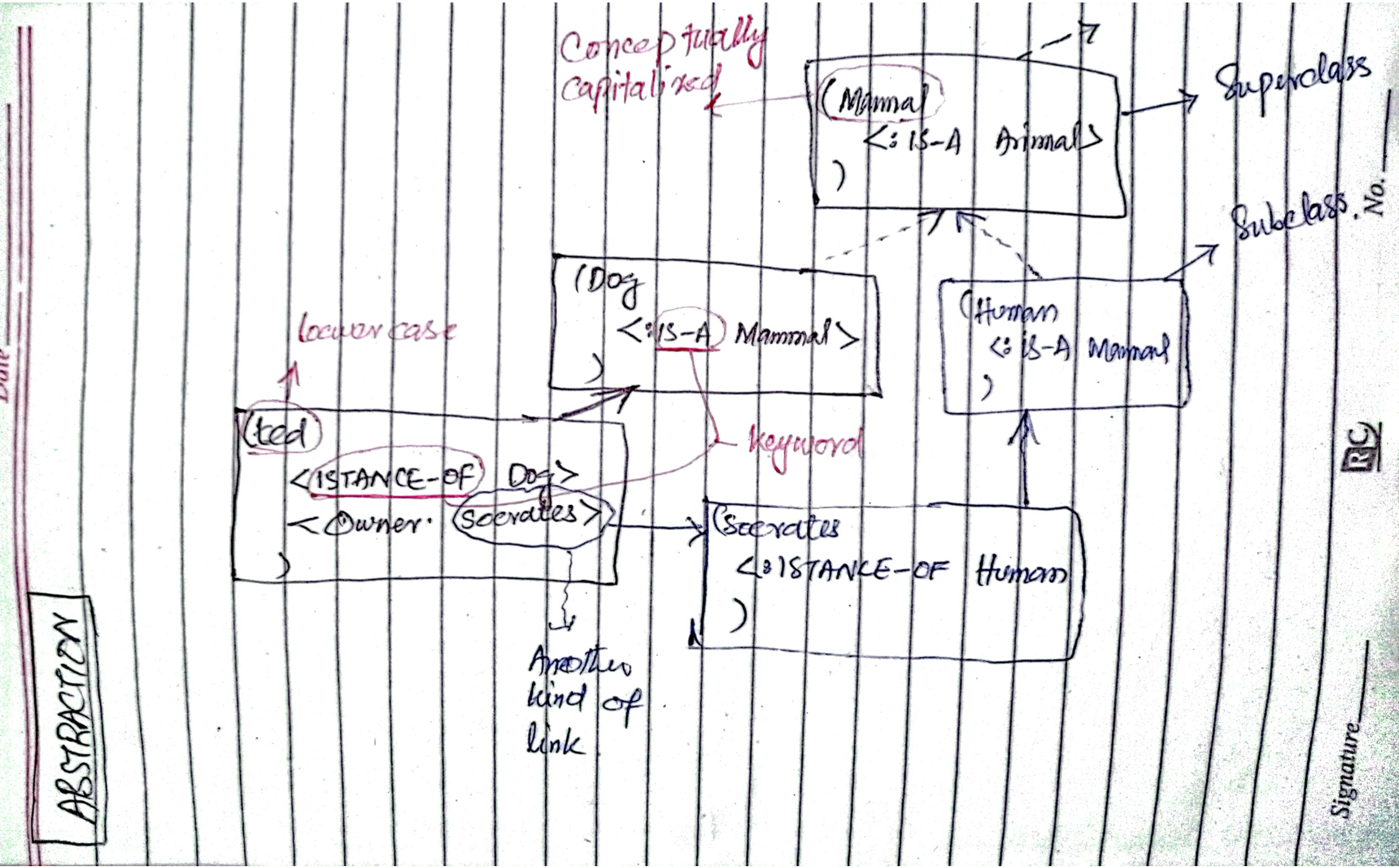


Date

Aggregation ...

Humans are organized into various structures with multiple levels.

- Political :- countries, states, cities, villages, families, - with individual governing structures.
- Geographic :- continents, countries, cities towns, villages.
- Corporate :- companies, divisions, units, - design, manufacturing, marketing


Conceptually capitalized
(Mamal
<: IS-A Animal>
)
Superclass
Subclass. No.
(Dog
<: IS-A Mammal>
)
(Human
<: IS-A Mamal
)
lower case
(ted
<ISTANCE-OF Dog>
<Owner. Socrates>
)
keyword
(Socrates
<: ISTANCE-OF Human
)
Another kind of link
ABSTRACTION
RC
Signature

Date

The Abstraction Hierarchy:

- Abstraction in frames is implemented by two special kind of slots.
  - > :IS-A when a generic frame is a specialization of another frame.
  - > :ISTANCE-OF when an individual frame is an instance of a generic frame.

- Note, that these links are handcrafted
  - o unlike in description logics.
  - o where they are defined logically.

?

- In frames there is nothing to stop one from saying.

(Whale
<: IS-A LargeFish>
<: IS-A Mammale>)

- A frame, like a concept in DL, can be a subclass of two frames
- Consistency issues:- (Resolving)

Date

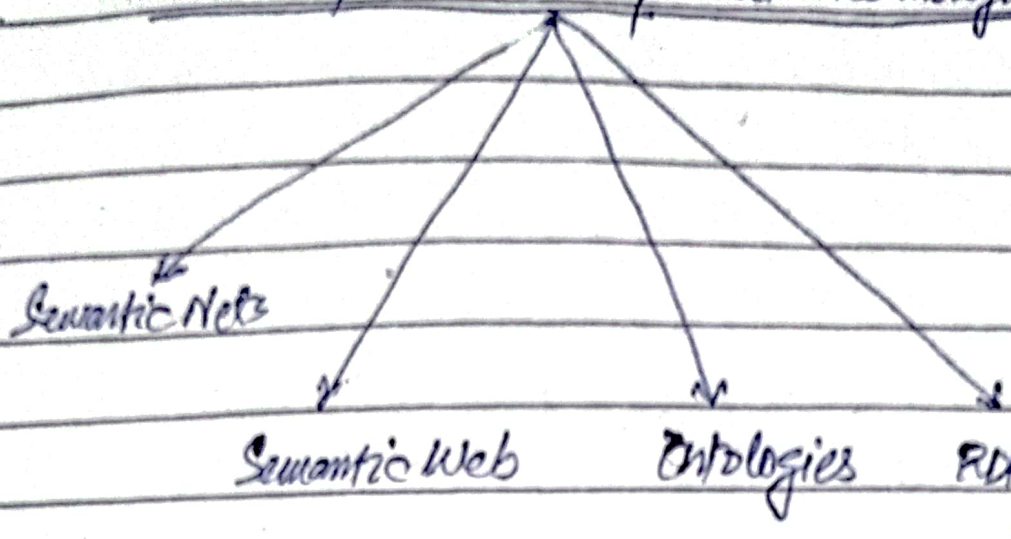
Default Reasoning

* Should Inheritance be Eager or Lazy?
- A property value stored in a superclass is inherited by a subclass or by an instance
- In a data driven or forward chaining mode:
  * the property is added to all subclasses when added.
  * Will have to be careful not to override properties stored in the subclass
  * Any subsequent change will have to be propagated.
  * but the value would be readily available in the subclass or instance.

- In a goal driven or backward chaining mode.
  * the value in the superclass will be accessed when needed
  * the upward search can be terminated when found.
  * Access time will increase.

Date

AI Representations parallel Technologies



Semantic Web

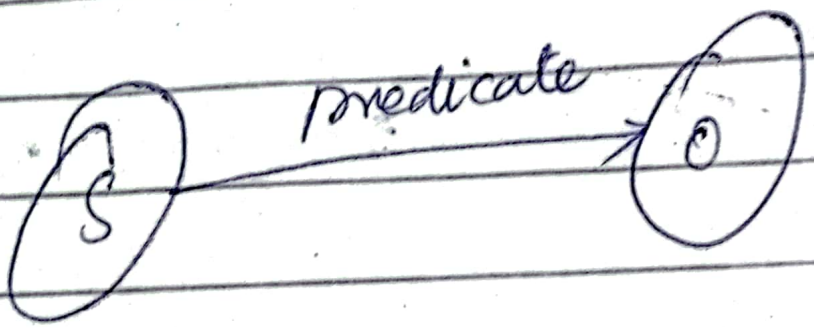
started in 1990s by Tim Berners-Lee
published seminal article in 2001.
Later W3C, World wide web consortium.
- published standards for Semantic Web formats.
- These standards facilitate exchange of semantically rich information.
- RDF:
- OWL
- SPARQL protocol & RDF Query language.
- RIF: Rule Interchange Format.

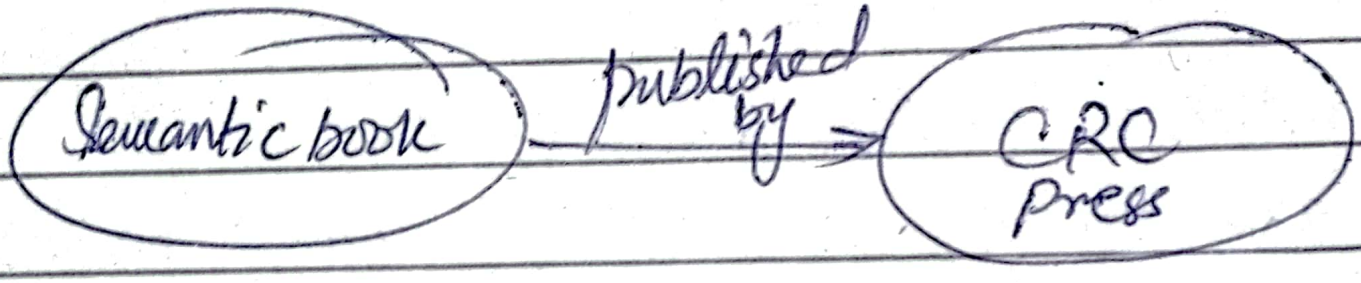
RDF a formal language for describing structured information
=> RDF consequently is often viewed

Date

as the basic representation format for developing the Semantic Web.

RDF stores, - also called triple stores for reason





= RDF is a framework
= RDF Schema is a language.
= RDF stores data in graphs.
= XML :- is encoded in Tree structures

Date

* Web 1.0 , 2.0 , 3.0 → aims to make data machine readable.

Semantic Web

uses standards, languages, and protocols i-e RDF , OWL , & SPARQL

Automated Agents

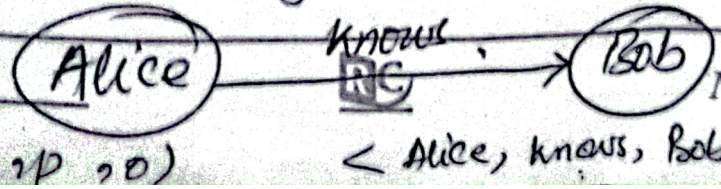
are SW programs for complex tasks.

Tim. Berners-Lee's Vision

web of data, instead of web of docs & pages.

RDF

* Standard framework used to represent information about resources on the web.
* Uses triplets (sub , predicate, object) to describe relationships b/w entities, enabling data to be linked & shared across different systems.
* provides structured, machine readable way to describe web-content.



Signature

No.

Triple (s , p , o)

< Alice, knows, Bob>

Date

OWL:

* Used to create & define ontologies. formal representations of knowledge within a specific domain,
* for complex representations.
* Complex class relationships only on Protégé
* Cardinality constraints: "Person can have

RDFS

is used for simple vocabularies in SW. by defining basic elements such as classes, properties & hierarchies

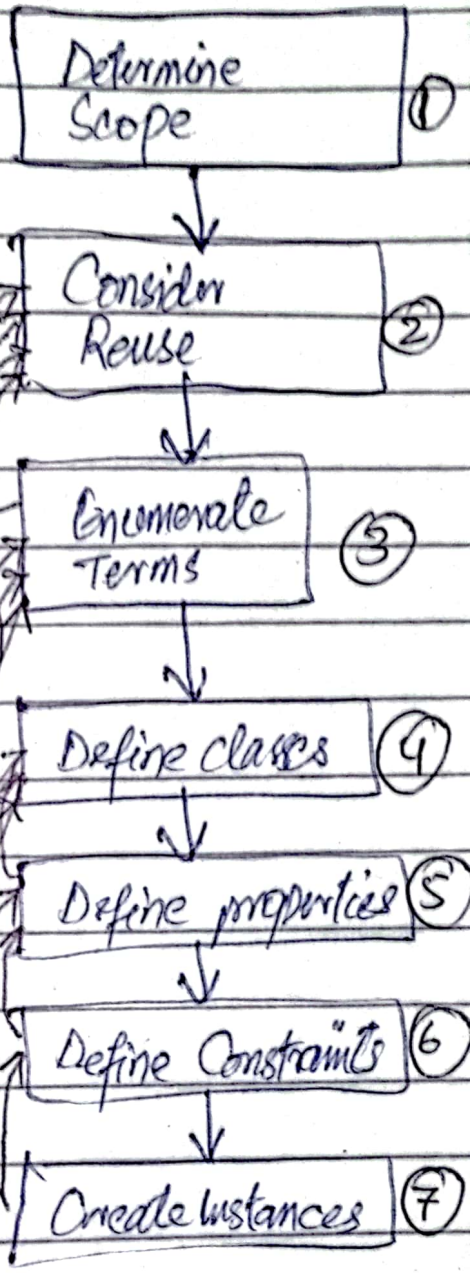
* Classes: Categories or types of things (e.g "Person", "Books")
* Properties: relationships "hasAuthor", "hasTitle"
* Hierarchies: Defining relationships b/w classes, or properties "Teacher" is a subclass of "Person"

RDFS is used for basic reasoning
⇒ limited in expressing complex relationships.

Date

Ontology Development

* In practice an iterative process that repeats continuously and improves the ontology
* there are always different approaches for modeling an ontology.
* In practice the designated application decides about the modelling approach.



Note:-
There is no correct way to model a domain. There are always viable alternatives.

1) Determine Scope

* Which domain should be covered by the the ontology.?
* what should the ontology be used for?
* what type of Questions should be

Signature_____ RC No._____

Date

answered by the knowledge represented in the ontology?

* Who will use and maintain the ontology?
* Formulation of Competence Questions

Competence Questions might change within the ontology life cycle.

Ofcource if some questions are you think are important, the definetly you must also vary the scope of the domain.

In some cases

2 Consider Reuse

* Why should we consider reuse?
  - In order to save cost
  - In order to apply tools that are applied for other existing ontologies also for our own ontology.
  - In order to reuse ontologies that have been validated by this application.

Note

If you don't find a suitable ontology or if the adaption is too complex then create a new ontology!

3 Enumerate Terms:

* About which concepts are we talking?
* Which properties have these concepts?
* What do we want to say about these concepts?
  - = Fajar, Namaz, Wazoo, etc.
  - = Rakats, Sujood, Qayam etc.
  - = Subtypes:
    - → Farz
    - → Nafal
    - → Sunnat
    - → Tahajjud
  - = etc.

4- Define Classes

* Classes are concepts in the designated domain

* Classes are collection of objects with similar properties.
* Choose a top-down / bottom-up / middle out approach to model class hierarchies.

5- Define Properties:

* Properties in a Class definition describe

Date

attributes of instances
- Every flower has a color, fragrance, size.

6- Define Property Constraints
* Property constraints (restrictions) describe or restrict the set of possible property values
  - The name of the flower is a "String"
  - etc.

7- Create Instances:
* Create instances for the class.
  - Every class directly becomes the type of its instance
  - Every superclass of a direct type is also type of its instances
* Create instances for properties, i-e assignment of property values for the instances according to the given constraints.

Date

When designing a new ontology following points are to be very clear:

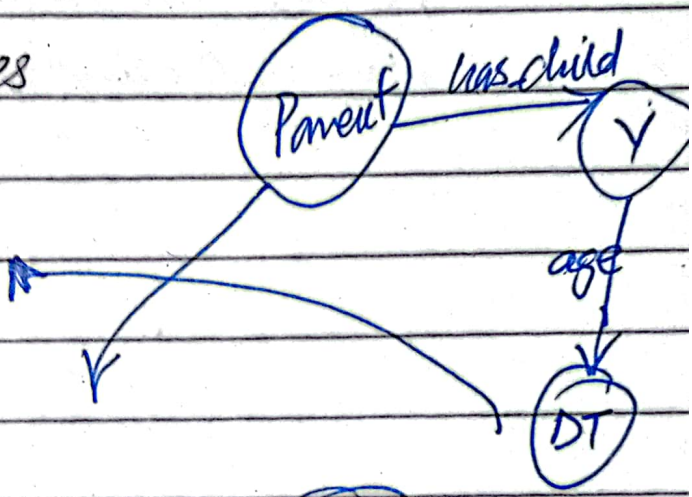
1- Classes (What and how many classes are to be)

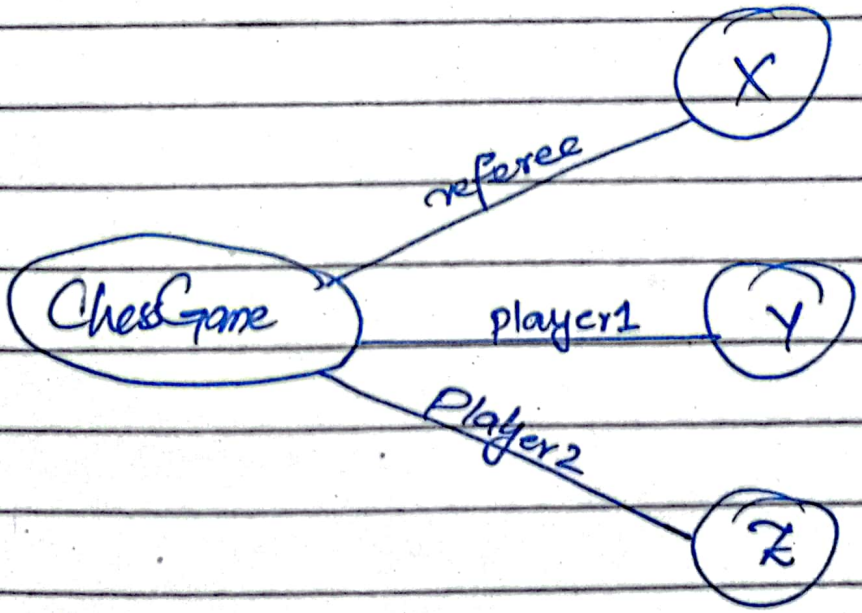
2- Object Properties-

3- Datatype Properties

4- Builtin Datatypes

& Instances.





referee (X, Y, Z)

X is the referee in a chess game b/w player Y & Z.

Let's introduce new auxiliary resource

Date

ChessGame , % binary predicates ref, player1
% player2 . Now we can represent
referee (X, Y, Z)

ref (ChessGame, X)
player1 (chessGam, Y)
player2 (chessGame, Z)