b. Division 6 - Wood and Plastics

- 06100 Rough carpentry
- 06110 Wood framing
- 06400 Architectural woodwork
- 06460 Wood frames

The following relations are identified for formalizing the above example :

1. **used_for** (class-class, human intention) : purpose.
2. **kind_of** (class-class, intrinsic): containment relation of attributes of instances.
3. **instance_of** (instance-class, intrinsic) : membership.
4. **made_of** (class-class, intrinsic): material component.

- Table 6.13.1 shows the mathematical properties of these relations that are used in the subsequent step for data normalization. They are also used for reasoning in knowledge extraction.

Table 6.13.1 : Mathematical properties of the relations

Relations	Transitive	Reflexive	Anti-symmetric
used_for	-	-	-
kind_of	+	+	+
instance_of	+	+	+
made_of	+	-	-

Step 2 : Relation statements construction

- This step is to construct simple statements using the relations defined in step one and all keywords in the taxonomy. The statements are then processed in subsequent steps for constructing ontology.
- There are two advantages using this bottom-up approach for formalizing taxonomies. One is that it can better address the dynamic nature of standards by enabling incremental updates and modifications of the statements and their resulting ontology.
- The other advantage is that domain experts who are not familiar with ontology can directly express their knowledge in the simple statements without communication overhead with knowledge modeling experts.
- The following are examples of relation statements that partially describe the example shown in previous step.
  1. Metals (D5), Wood (D6), Plastics (D6_1) are **instance_ of** Material (root) →(D5_root, D6_root,D6_1_root)
  2. Metals (D5) are **used_for** framing → 05100_1
  3. Structural is a **kind_of** "metal framing" (05100_1) : 05100
  4. Cold formed is a **kind_of** "metal framing" (05100_1) : 05400
  5. Studes are **made_of** Metals (D5) → (05410_1)

6. "Load bearing metal studs" are **kind_of** Metal studs (05410_1) : 05410

7. 05410 is **used_for** 05400 : (05400_05410)

Step 3 : Normalization

- It is likely that redundant or conflict statements are generated along the way when domain experts annotate their taxonomies in the above steps.
- Based on the mathematical properties of the relations, this step normalizes the statements by :
  1. **Redundancy elimination** (removing same or equivalent statements)
  2. **Conflict detection** (for example, A-r1-B, and B-r1-A statements are conflict if r1 has asymmetric property)
  3. **Implication detection** (for example, A-r1-B, and B-r1 C statements imply A-r1-C through transitive property).

Step 4 : Semi-automatic generalization

- This step is to generalize the resulting statements from step 3 into higher-level concepts connected by the same set of relations.
- Human being intervention is required in this step due to the complexity of the process.
- For example, if there exist A-r1-C, A-r1-D, B-r1-C, and B-r1-D, they can be generalized to concept1 {A, B}-r1-concept2 {C, D} by union. However, it becomes difficult when the above example is extended to include concept1 {A, B}-r1-E and concept2 {C, D}-r2-F.
- One cannot conclude concept1 {A, B}-r1-concept2 {C, D, E} unless an exception indicating no E-r2-F is added. Alternatively, it can be generalized to concept1 {A, B}-r1-concept3 {E, concept2 {C, D}}. The system interacts with users by prompting the dilemmas for resolutions along the process of a whole taxonomy.
- Fig. 6.13.2 shows the generalized view or ontology of the relation statements shown in previous steps.

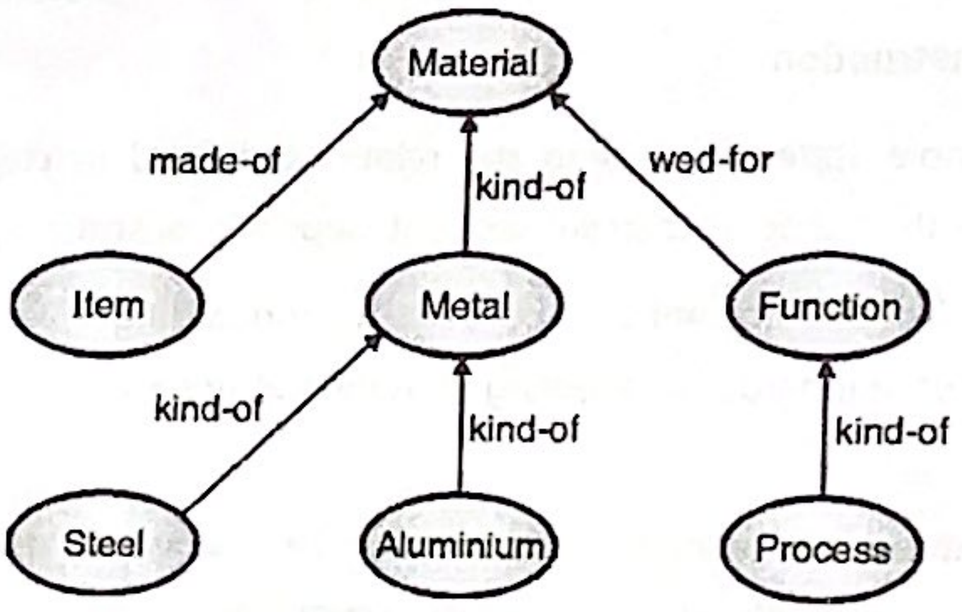

Fig. 6.13.2 : Ontology Example

6.13.4 Ontology Extraction from Text

The most widely used approaches at the ontology extraction from text are as shown in Fig. 6.13.3.

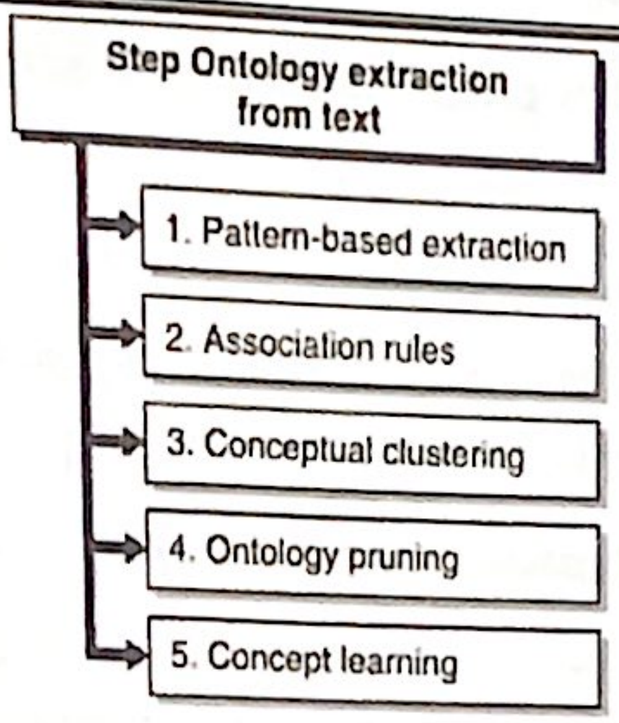

Fig. 6.13.3 : Steps for ontology extraction from text

1. Pattern-based extraction

This approach usually uses heuristic methods that examine the text with distinctive lexicon-syntactic pattern. A relation is recognized and extracted if a sequence of words within the text matches a pattern. The basic idea of this approach is very simple: to define a regular pattern that is able to capture the expression presented by the text and able to map the results of the matching into a semantic structure such as taxonomy of relations among concepts.

2. Association rules

Initially defined to extract information from databases (in the data mining field, has been used to discover non taxonomic relations between concepts using a concepts hierarchy as knowledge base).

3. Conceptual clustering

In this approach the concepts are grouped according to their semantic similarity to build hierarchies. The semantic similarity can be calculated with different methods. For example it may be calculated according to the distributional approach : less distance between the linguistic distributions of two words means more similar concept.

4. Ontology pruning

The aim of this approach is to build a domain ontology based on various sources. It includes the following steps :

1. Generic core ontology is used as basic infrastructure for a domain specific ontology.
2. Dictionary with important domain terms is used for the domain concept acquisition and these concepts are classified into the generic core ontology.
3. Domain specific and general corpora of text are used for the process of non domain specific concepts removal, following the heuristics that domain specific concepts should be more frequent in a domain-specific corpus than in a generic one.

5. Concept learning

With this approach a given taxonomy in incrementally enriched acquiring new concepts from textual documents.

6.14 Searching Across Ontologies

University Question

Q. Write a note on "Ontology based information sharing". **SPPU : May 17, 5 Marks**

- This is the information era. Information society needs to gain access of complete information which can be heterogeneous as well as distributed.
- But it is not so much simple because number of technical problems need to be handled.
- The problem of bringing together heterogeneous and distributed computer systems is known as interoperability problem. Interoperability problem need to be handled at technical level as well as information level.
- The data which is accessed by heterogeneous or distributed system must be available in such a way that remote system can use it for its own purpose.
- The problems in interoperability are structural heterogeneity and semantic heterogeneity.
- Structural heterogeneity means different structures are used by different systems to store the information. In semantic heterogeneity conflict may occur because of different interpretation of same data. Thus both the issues need to be resolved while sharing the information.
- The use of ontologies for the explication of implicit and hidden knowledge is a possible approach to overcome the problem of semantic heterogeneity.
- There are various approach to use ontology. These approaches difference based on :
  1. **Use of Ontologies :** The role and the architecture of the ontologies influence heavily the representation formalism of ontology.
  2. **Ontology Representation :** Depending on the use of the ontology, the representation capabilities differ from approach to approach.

Role of ontologies

- Ontologies are initially considered as explicit specification of a conceptualization.
- Thus ontologies can be used defining semantic of information sources.
- But for heterogeneous scenario, ontology can be used as identification and association of semantically corresponding information concepts. Following are the roles of ontology.

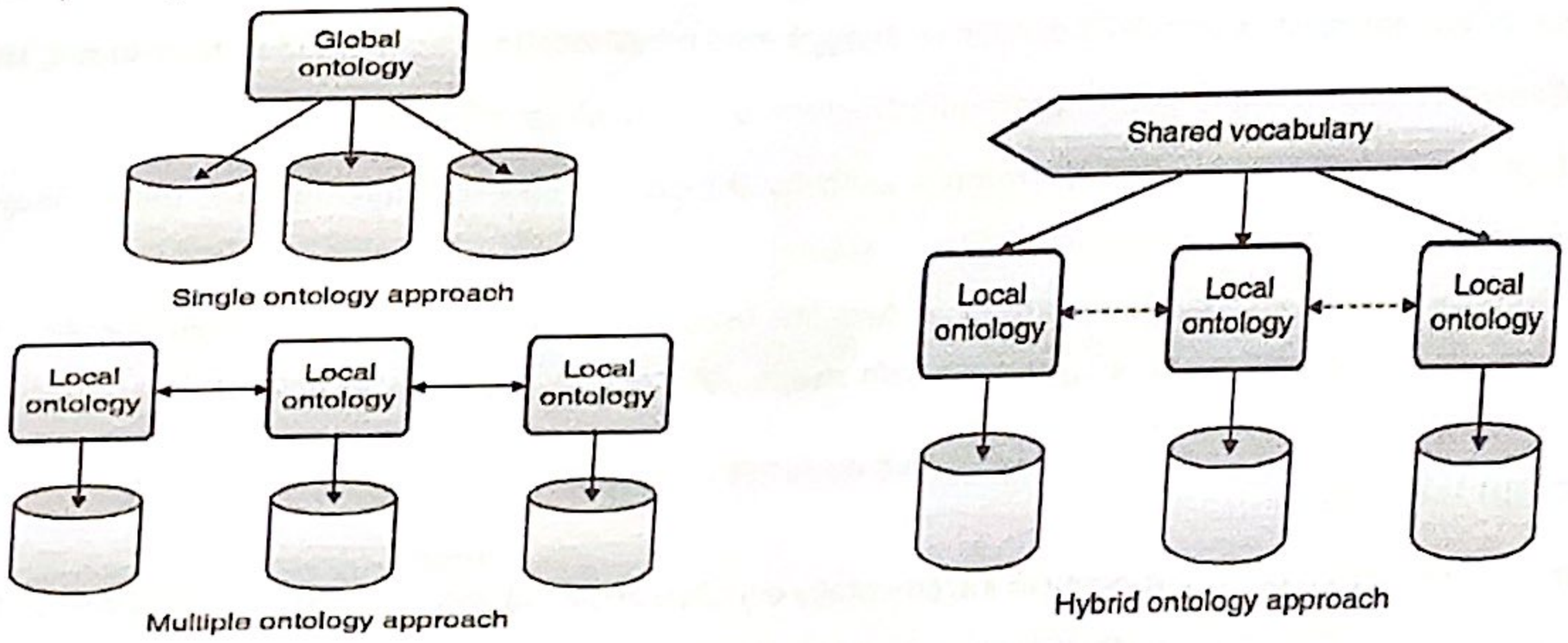

Fig. 6.14.1 : The three possible ways for using ontologies for content explication

6.14.1 Content Explication

Ontology works related to semantic. Various approaches are present to use ontology as shown in Fig. 6.14.2.

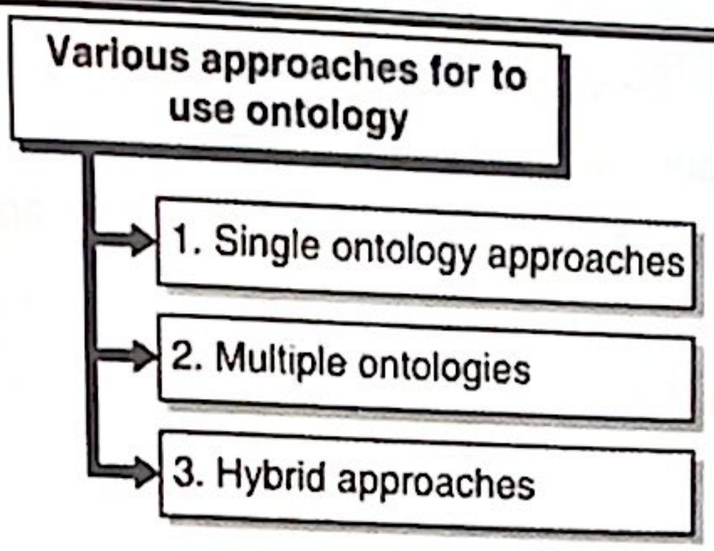

Fig. 6.14.2 : Approaches for use ontology

1. Single ontology approaches

- In this approach, a single ontology is shared and works as global ontology. All the objects will refer the same semantic while interpreting the information.
- A global ontology can be combination of several specialized ontologies. Single ontology approaches can be applied to integration problems where all information sources to be integrated provide nearly the same view on a domain.
- But if one information source has a different view on a domain, e.g. by providing another level of granularity, finding the minimal ontology commitment becomes a difficult task.
- Also, single ontology approaches are susceptible to changes in the information sources which can affect the conceptualization of the domain represented in the ontology.
- Depending on the nature of the changes in one information source it can imply changes in the global ontology and in the mappings to the other information sources. Because of the limitations of single ontology approach, multiple ontology approaches are introduced.

2. Multiple ontologies

- In multiple ontology approaches, each information source is described by its own ontology. For example, in OBSERVER. The semantics of an information source is described by a separate ontology. In principle, the "source ontology" can be a combination of several other ontologies but it cannot be assumed that the different "source ontologies" share the same vocabulary.
- Multiple ontologies overcome limitations of single ontology. The advantage of multiple ontology approaches seems to be that no common and minimal ontology commitment about one global ontology is needed. Each source ontology could be developed without respect to other sources or their ontologies - no common ontology with the agreement of all sources are needed.
- Thus it simplifies the task of changes in information source or adding or removing information sources. But in reality the lack of a common vocabulary makes it extremely difficult to compare different source ontologies. Thus to gain advantages of both, single and multiple ontologies, hybrid ontologies are introduced.

3. Hybrid approaches

- To overcome the drawbacks of the single or multiple ontology approaches, hybrid approaches were developed. Similar to multiple ontology approaches the semantics of each source is described by its own ontology.

- But in order to make the source ontologies comparable to each other they are built upon one global shared vocabulary. The shared vocabulary contains basic terms (the primitives) of a domain. In order to build complex terms of a source ontologies the primitives are combined by some operators.
- Because each term of a source ontology is based on the primitives, the terms become easier comparable than in multiple ontology approaches. Sometimes the shared vocabulary is also an ontology.

6.14.2 Query Model

- Integrated information sources normally provide an integrated global view. Some integration approaches use the ontology as the global query schema. For example, in SIMS the user formulates a query in terms of the ontology.
- Then SIMS reformulates the global query into sub-queries for each appropriate source, collects and combines the query results, and returns the results.
- Using an ontology as a query model has the advantage that the structure of the query model should be more intuitive for the user because it corresponds more to the user's appreciation of the domain.
- But from a database point of view this ontology only acts as a global query schema. If a user formulates a query, he has to know the structure and the contents of the ontology; he cannot formulate the query according to a schema he would prefer personally.
- Therefore, it is questionable where the global ontology is an appropriate query model.

6.14.3 Verification

- During the integration process several mappings must be specified from a global schema to the local source schema. The correctness of such mappings can be considered ably improved if these can be verified automatically.
- A sub-query is correct with respect to a global query if the local sub-query provides a part of the queried answers, i.e. the sub-queries must be contained in the global query (query containment).
- Since ontology contains a (complete) specification of the conceptualization, the mappings can be validated with respect to the ontologies. Query containment means that the ontology concepts corresponding to the local sub-queries are contained in the ontology concepts related to the global query.

6.15 Semantic Web Search

University Question

Q. Write a note on : "Ontology languages for semantic web". **SPPU : Dec. 16, May 19, 5/8 Marks**

- Many ontology languages are developed during the last few years and these are developed in the context of semantic web.
- Following are some ontology languages :
  1. Ontology Exchange Language (XOL) : based on XML syntax
  2. 3 SHOE4 : previously based on HTML
  3. Ontology Markup Language (OML)
  4. Resource Description Framework (RDF)6
  5. RDF Schema7 : RDF 6 and RDF Scheme7 are created by World Wide Web Consortium (W3C) working groups.

6. Ontology Inference Layer (OIL)8
7. DAML+OIL9

- Ontology Inference Layer (OIL)8 and DAML+OIL9 languages are built on top of RDF(S) i.e. the union of RDF and RDF Schema to improve its features.

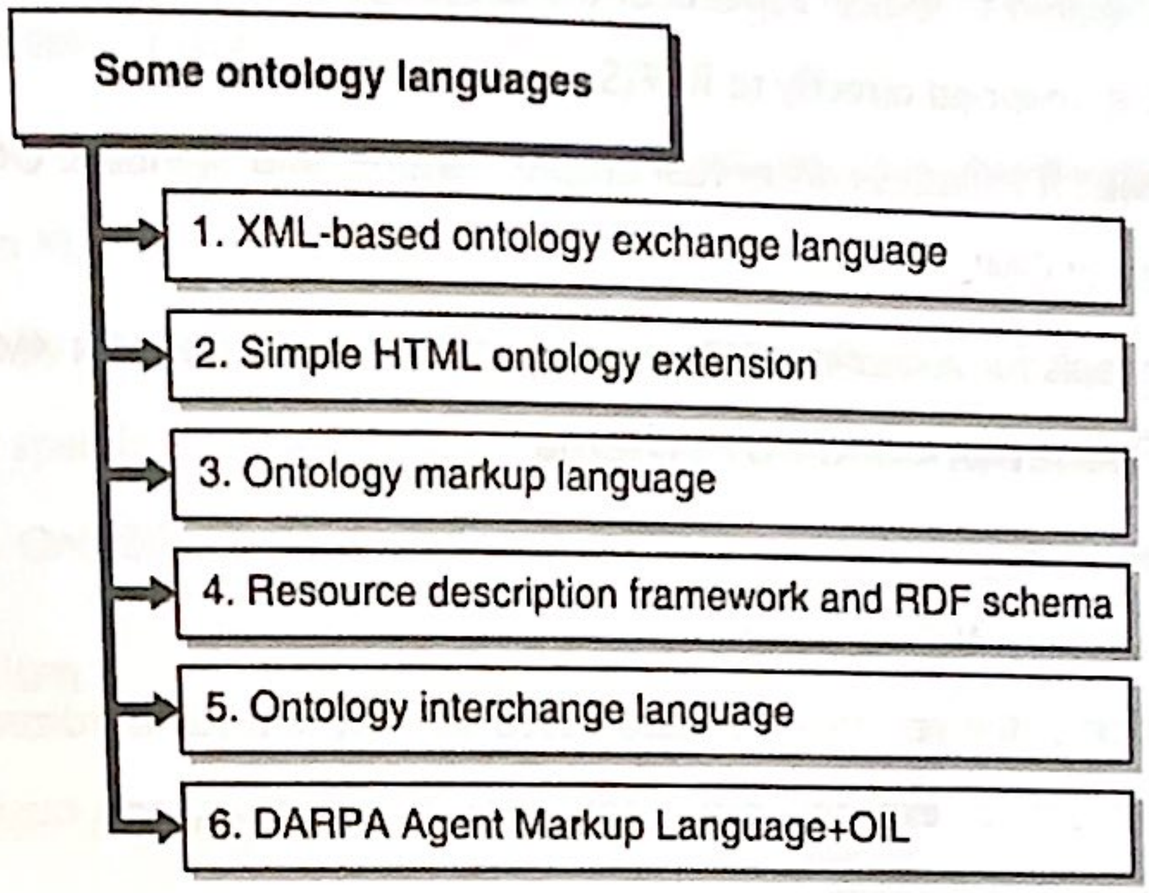

Fig. 6.15.1 : Some ontology languages

1. XML - based ontology exchange language

- The US bioinformatics community has designed XOL by studying the representational needs of experts in bioinformatics.
- This language is designed for the exchange of ontology definitions among a heterogeneous set of software systems in their domain.
- Ontolingua and OML are used as the basis for creating XOL, merging the high expressiveness of OKBC-Lite, a subset of the Open Knowledge Based Connectivity protocol, and the syntax of OML, based on XML.
- Tools are not available to develop XOL but XML editors are used to author XOL files.

2. Simple HTML ontology extension

- SHOE is developed at the University of Maryland and used to develop OML.
- SHOE was created as an extension of HTML and machine-readable semantic knowledge in HTML documents or other Web documents are incorporated into it.
- University of Maryland has adapted the SHOE syntax to XML.
- Using SHOE agents can gather meaningful information about Web pages and documents to improve search mechanisms and knowledge gathering.
- This process consists of three phases :
  1. Define an ontology.
  2. Annotate HTML pages with ontological information to describe themselves and other pages.
  3. Have an agent semantically retrieve information by searching all the existing pages and keeping information updated.
  4. The Knowledge Annotator annotates ontological information in HTML pages.

3. Ontology markup language

- OML is developed at the University of Washington and is partially based on SHOE.

- Initially it was considered an XML serialization of SHOE. So there are many common features between OML and SHOE.
- Four different levels of OML are
  1. **OML Core :** It is related to logical aspects of the language and is included by the rest of the layers;
  2. **Simple OML :** It is mapped directly to RDF(S);
  3. **Abbreviated OML :** It includes conceptual graphs features and Standard OML.
  4. Expressive version of OML.
- As there are no other tools for authoring OML ontologies so general purpose XML edition tools are used.

4. Resource description framework and RDF schema

- RDF is developed by the W3C.
- It is developed for describing Web resources.
- It allows the specification of the semantics of data based on XML in a standardized and interoperable manner.
- It also provides mechanisms to explicitly represent services, processes, and business models, while allowing recognition of non explicit information.
- The RDF data model is equivalent to the semantic networks formalism.
- It consists of three object types :
  1. **Resources :** These are described by RDF expressions and are always named by URIs plus optional anchor IDs.
  2. **Properties :** It define specific aspects, characteristics, attributes, or relations used to describe a resource.
  3. **Statements :** It assign a value(might be another RDF statement) for a property in a specific resource.
- The RDF data model does not provide any mechanisms for defining the relationships between properties (attributes) and resources.
- RDFS offers primitives for defining knowledge models that are closer to frame-based approaches.
- RDF(S) is widely used as a representation format in many tools and projects, such as Amaya, Protégé, Mozilla, SilRI, and so on.

5. Ontology interchange language

- OIL was developed in the Onto Knowledge project.
- It allows semantic interoperability between Web resources.
- Syntax and semantics of OIL are based on existing proposals like OKBC, XOL, and RDF(S).
- It provides modelling primitives and formal semantic and reasoning support.
- It also provides decidability and an efficient inference mechanism).
- OIL was built on top of RDF(S) and has the following layers :
  1. Core OIL groups the OIL primitives that have a direct mapping to RDF(S) primitives.
  2. Standard OIL is the complete OIL model, using more primitives than the ones defined in RDF(S).
  3. Instance OIL adds instances of concepts and roles to the previous model.
  4. Heavy OIL is the layer for future extensions of OIL.

- The tools lokeOILEd, Protégé2000, and WebODE can be used to author OIL ontologies.
- OIL's syntax is expressed in XML as well as ASCII.

6. DARPA agent markup language + Oil

- A joint committee from the US and the European Union (IST) has developed DAML+OIL i.e. DARPA. And it is developed in the context of DAML.
- As DARPA allows semantic interoperability in XML it shares the same objective as OIL.
- DAML+OIL is built on RDF(S).
- Its name implicitly suggests that there is a tight relationship with OIL.
- It replaces the initial specification, which was called DAML-ONT, and was also based on the OIL language.
- The tools like OILEd, OntoEdit, Protégé2000, and WebODE are used to author DAML+OIL ontologies.

6.16 Ontology Creation

An ontology should contain following terms :

- **Classes** that represent concepts (either physical/specific or abstract/conceptual).
- They could be organized in taxonomies to define superclass- subclass hierarchy.
- Relations that represent association between concepts. They are usually binary.
- Attributes (also called properties, slots. . .) to describe the features of the concepts.
- Formal axioms to model sentences those are always true.
- Functions are special case of relations.
- Instances that represent elements or individuals in an ontology.

Domain ontology

- There are different types of ontologies. For example, we can create domain ontology i.e.
- A conceptualization of concepts, relations and attributes belonging to a particular field of our interest.

6.16.1 Methodology : General Ideas

To develop ontology we should,

- Define concepts, i.e., classes.
- Organize them somehow in a taxonomy (remember inheritance issues among superclass- subclass).
- Define relations among the classes.
- Define the attributes and which values they can take.
- Define instances, i.e., "real" elements in our domain.
- If possible, axioms and functions.
- There is not a correct way to model a domain: you can have an idea on the domain that will for sure differ from mine. But by understanding, we can finalize.
- The ontology development is an iterative process.

- For simplicity, initially we can start with nouns and verbs from your knowledge sources (i.e. the documents you are using and your own knowledge on the domain) : It A noun will be a class, attribute, instance. A verb will be a relation. Obviously, refinement and iterations will be needed to further clarify this.

6.16.2 Steps to Follow

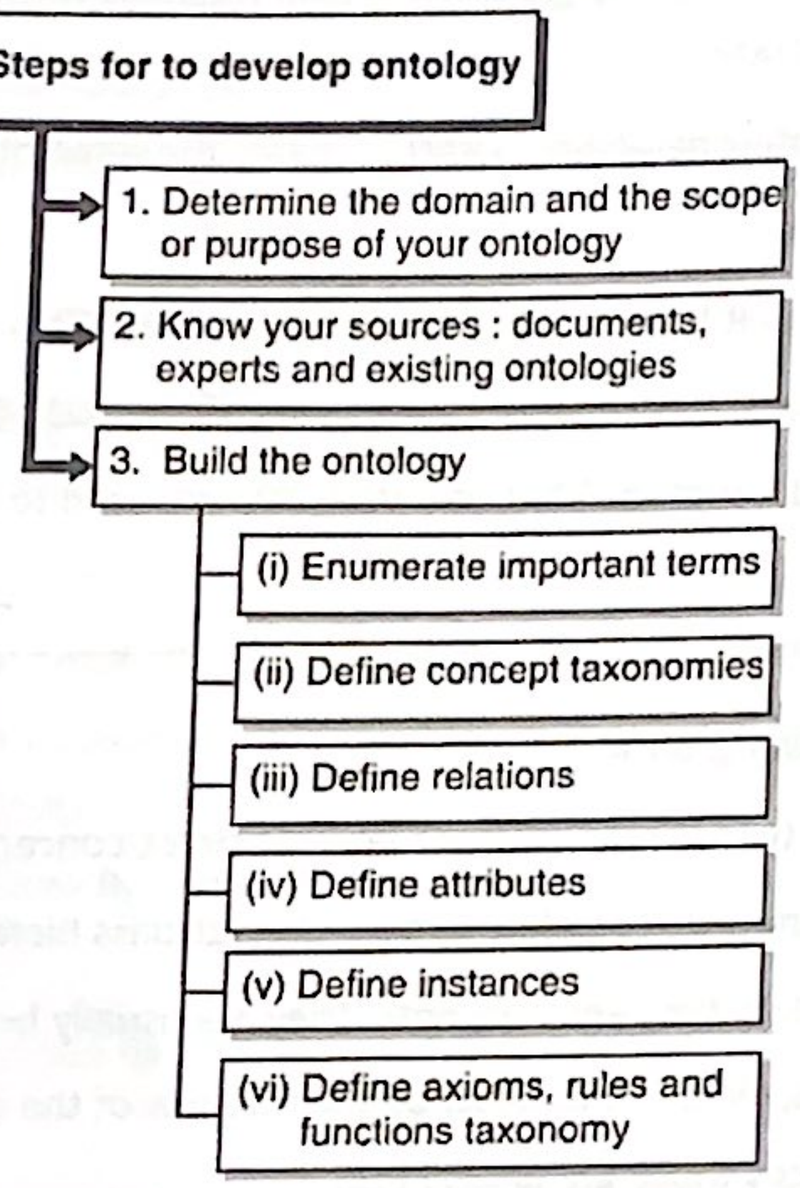

Fig. 6.16.1 : Steps for to develop ontology

1. Determine the domain and the scope or purpose of your ontology

Basically, try to find an answer to questions such as :

- Which domain are you thinking of?
- What will you use the ontology for?
- Is it going to be just one, or will you need different subontologies to make it clearer?
- Who will use the ontology?
- It is the hardest part to develop the ontology. It might be further clarified, but at least we need a good initial scope or purpose for your ontology.

2. Know your sources : documents, experts and existing ontologies

- Either you are an expert on the domain or more common you have a partial knowledge of the domain.
- In the first case, start with your thinking. In the second case, you will need more knowledge that can come from either experts or documents which are already available. We can take help of existing ontologies and based on these, we can develop our own.

3 Build the ontology

An ontology development usually encompasses several tasks. Different methodologies order them differently, but in general you should.

i. Enumerate important terms

Make a list or graph of all nouns and verbs. Follow a common tool while finalizing the list. For each term, try to write a name, synonym, a natural language description, type, source (to remember why you put it there) and comments. Finally decide whether a noun will be part of concept, attribute or instance.

ii. Define concept taxonomies

- In this step, classify the concepts in a hierarchy(taxonomy). Not all concepts will own a hierarchy, but as we list them, some nouns seem to be related as types (subclasses) of other superclasses).
- Traditionally, taxonomies/hierarchies are done following top-down (from general to specific), bottom-up (from specific to general) or combination processes.

1. **Subclass :** a concept C_1 is subclass of concept C_2, iff (if and only if) every instance of C_1 is also instance of C_2.
2. **Disjoint decomposition of C :** set of subclasses of C that do not have common instances and do not cover C.
3. **Exhaustive decomposition of C :** set of subclasses of C that may have common instances and subclasses and do cover C.
4. **Partition of C :** set of subclasses that do not share common instances but cover C.

iii. Define relations

Define each diagram and the relations in detail by giving a name, source concept, target concept, cardinality (how many instances of a concept are related with how many of the others), inverse name (we can read from A to B, but also from B to A. Sometimes, the distinction is important).

iv. Define attributes

At this stage, some of nouns in the list could have been considered attributes, i.e., terms used to describe others. Ontologists distinguish between class attributes (terms to describe concepts which take their values in the class they are defined, and they are not inherited in the hierarchy) and instance attributes (terms to describe concepts that take their values in the instance, and may be different for each instance). There is not a unique way to do this, but some guidelines could be :

- This step and the definition of taxonomies are intertwined: some classes might end up being attributes to describe the different classes and/or instances.
- Try to attach the attribute to the most general class/concept that can have that property.
- If it can have a well defined type (integer, string, float) it is an attribute, not a class.
- Try to define type attributes (integer, string, float, . . .). We can define our own types, or use the traditional ones.
- Try to define a range, value, precision, related classes. .

v. Define instances

An instance is an individual of a class, you can describe in detail relevant instances that may appear by giving them a name, concept to which they are related, attribute names and values.

vi. Define axioms, rules and functions taxonomy

Some require axioms and rules to be described before describing instances. It is up to you.

Review Questions

Q. 1 Explain basic concepts of XML.

Q. 2 Write a short note on : Challenges in XML Retrieval.

Q. 3 Explain vector space model for xml retrieval in detail.

Q. 4 Write a short note on : Evaluation of XML Retrieval.

Q. 5 Compare Text-Centric and Data-Centric XML Retrieval.

Q. 6 Write a note on : "Ontology languages for semantic web".

❑❑❑