32002: AI (Knowledge Representation:Ontological engineering)	Spring 2014

Lecture 18: February 18, 2014

Lecturer: K.R. Chowdhary *: Professor of CS*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

18.1 Introduction

Ontology is a particular theory about the nature of being or the kinds of existent. The task of intelligent systems in computer science is to formally represent these existents. A body of formally represented knowledge is based on conceptualization. Conceptualization consists of a set of objects, concepts, and other entities about which knowledge is being expressed and of relationships that hold among them. Every knowledge model is committed to some conceptualization, implicitly or explicitly.

An ontology is a systematic arrangement of all of the important categories of objects or concepts which exist in some field of discourse, showing the relations between them. When complete, an ontology is a categorization of all of the concepts in some field of knowledge, including the objects and all of the properties, relations, and functions needed to define the objects and specify their actions. A simplified ontology may contain only a hierarchical classification (a taxonomy) showing the type subsumption relations between concepts in the field of discourse. An ontology may be visualized as an abstract graph with nodes and labeled arcs representing the objects and relations. Figure 18.1 shows these relations.

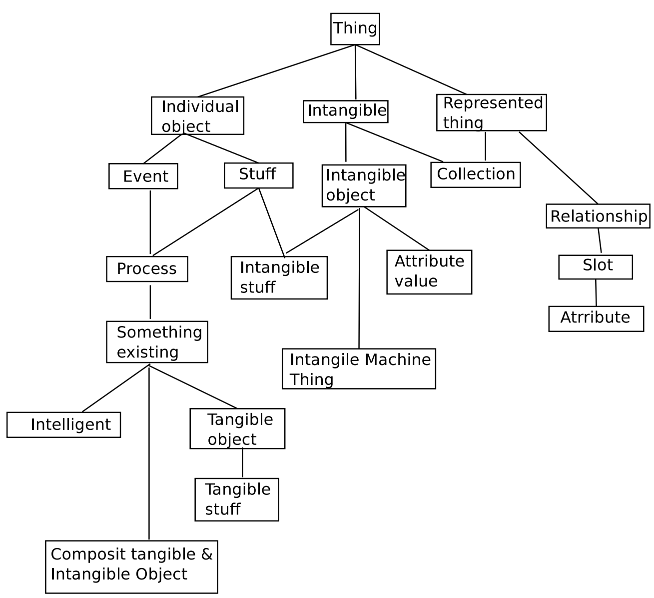

Figure 18.1: World Ontology.

The concepts included in an ontology and the hierarchical ordering will be to a certain extent arbitrary, depending upon the purpose for which the ontology is created. This arises from the fact that objects are of varying importance for different purposes, and different properties of objects may be chosen as the criteria by which objects are classified. In addition, different degrees of aggregation of concepts may be used, and distinctions of importance for one purpose may be of no concern for a different purpose.

The First-Order Predicate Logic (FOPL) for knowledge representation and reasoning, in the ontologies, we are particular for knowledge *organization* as well as *contents*. This approach has generalization with exceptions. For example, "all birds fly" can be a rule. But, there are some birds which do not fly, like penguin, and ostrich. These are exceptions. Thus, we should be able to add these concepts of knowledge, not as exceptions, but as extensions. This requires categorization of objects.

Ontologies provide explicit representations of domain concepts with a structure around which knowledge bases can be built (see figure 18.2). Each ontology is a system of concepts and their relations, in which all concepts are defined and interpreted in a declarative way. The system defines the vocabulary of a problem domain and a set of constraints on how terms can be combined to model the domain.

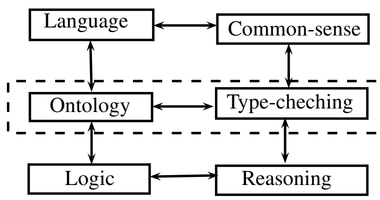

Figure 18.2: Language, Logic, and Ontology.

In comparing various ontologies, they can be viewed at three different levels: (1) a *is-a* taxonomy of concepts, (2) the internal concept structure and relation between concepts, and (3) the presence or absence of explicit axioms (figure 18.3). Taxonomy is central part of most ontologies, and its organization can vary greatly, for example, all concepts can be in one large taxonomy, or there can be number of smaller hierarchies, or there can be no explicit taxonomy at all.

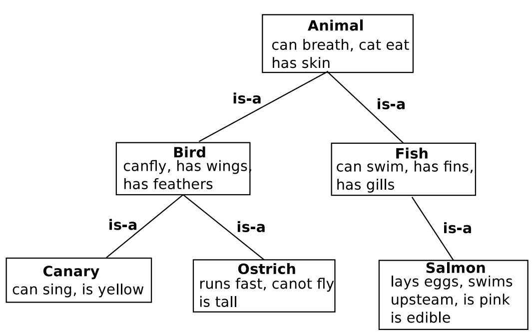

Figure 18.3: Top level ontology for Animal.

Although all general-purpose ontologies try to categorize the same world, they are very different at the top level. They also differ in their treatment of basic parts: things, processes, and relations.

The next level of comparison is internal concept structure, which can be realized by properties and roles. Concepts in some ontologies are atomic and might not have any properties or roles or any other internal structure associated with them.

An important test for any ontology is the practical applications it is used for. These can be applications in natural language processing, information retrieval, simulation and modeling, and so on, that use knowledge represented in the ontology.

The CYC Project is an example of ontology. CYC contains more than 10,000 concept types used in the rules and facts encoded in the knowledge base. The upper level of the CYC hierarchy is presented in figure 18.1. At the top of the hierarchy is the Thing concept, which does not have any properties of its own. The hierarchy under *Thing* is quite tangled. Not all the subcategories are exclusive. In general, Thing is partitioned in three ways: First is *Represented Thing* versus *Internal Machine Thing*. Every CYC category must be an instance of one and only one of these sets. Internal-Machine-Thing is anything that is local to the platform CYC is running on (strings, numbers, and so on). *Represented-Thing* is everything else.

18.1.1 Sowa's Ontology

John Sowa (1997, 1995a) states his fundamental principles for ontology design as "distinctions, combinations, and constraints" (1995a, p. 175). He uses philosophical motivation as the basis for his categorization. There are three top-level distinctions: First is *Physical versus Information*, or *Concrete versus Abstract*. This is a disjoint partition of all the categories in the ontology.

Second is *Firstness* versus *Secondness* versus *Thirdness*, or *Form* versus *Role* versus *Mediation*. These categories are not mutually exclusive. For example, *Woman* is considered to be a form (Firstness) because it can be defined without considering anything outside a person. As a mother, a teacher, or an employee, the same individual would be an example of a role (Secondness). These roles represent an individual in relation to another type (a child, a student, an employer). Marriage is a mediation (Thirdness) category because it relates several (in this case, two) types together.

Third, *Continuant versus Occurrent*, or *Object versus Process*. Continuants are objects that retain their identity over some period of time; occurrents are processes "whose form is in the state of flux" (Sowa 1995a, p. 179). For example, *Avalanche* is a process, and *Glacier* is an object. Note that this distinction depends on the time scale. On a grand time scale of centuries, Glacier is also a process. These distinctions are combined to generate new categories (figure 18.4). At a lower level, for example, Script (for example, a computer program, a baking recipe) is a form that represents sequences and is thus defined as *Abstract*, *Form*, *Process*. Also, History (an execution of a computer program), which is a proposition that describes a sequence of processes, is then Abstract, Form, Object.

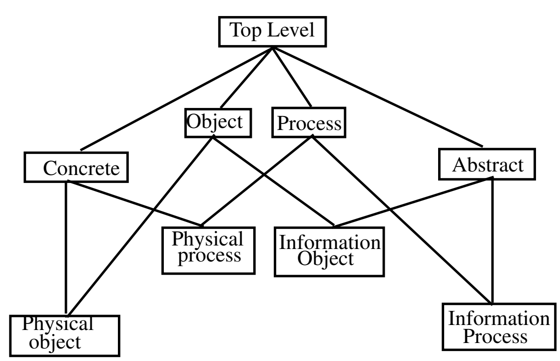

Figure 18.4: John Sowa's Ontology.

At the top level, there is a category for every possible combination of distinctions. To avoid too many combinations, however, constraints are used at the lower levels to rule out categories that cannot exist. Logical constraints, for example, would rule out triangles with four sides, and empirical constraints would rule out talking birds. Constraints are represented as axioms and are inherited through the hierarchy to lower levels.

References

[1] Chowdhary K.R. (2020) Logic and Reasoning Patterns. In: Fundamentals of Artificial Intelligence. Springer, New Delhi. https://doi.org/10.1007/978-81-322-3972-7_6