

Chapter 1

Computational Knowledge and Ontology

Abstract. The definition of knowledge is an augmentative topic in philosophy. We do not try to find out an explicit meaning of philosophical knowledge, but the most important thing is that we should know about what is knowledge in computer system, as called computational knowledge. In this chapter, we introduce some researches and definitions related to knowledge and computational knowledge. Ontology is a word used in both philosophy and computer system to describe the formalization of knowledge. We shall look into the definition of ontology in brief and also introduce its formalization methods in computer system.

1.1 What Is Knowledge?

“Knowledge” has been discussed by many philosophers since the Greek ancient times. It is not an easy task to find out a high abstraction of definition about “knowledge”. However, in very generally speaking, knowledge can be said as a meaningful resource that makes us know about the world. Theories of knowledge define what is about the world, how is it encoded, and in what way we reason about the world. Similar definition can be applied in computer and information system, unless we are defining those in the aim of computer processing instead of human understanding, and it is called computational knowledge.

1.2 Computational Knowledge in Information Systems

Computational knowledge in computer system has been represented as a hierarchy of data-information-knowledge in many knowledge management theories (Daft, 2004; Devenpart and Prusak 2000). Data refers to a string of bits, numbers or symbols that are only meaningful to a program. Data with meaning, such as computing words, texts, database records, etc, define information that is meaningful to human. Knowledge is the highest level of abstraction, which is encoded in some form inside information.

Creating computational knowledge is a study of artificial intelligent (AI) - an area of computer science focusing on making a computer to perform tasks with more intelligence (Genesereth and Nilsson 1987). Advanced information systems, such as information retrieval system, forecasting system, resource management system, online shopping system, personalization system, etc, always require computational knowledge to perform tasks with more intelligence. Traditional

information systems are lacking of intelligence because they process data and information without analyzing the knowledge behind. To enable a computer understand and process knowledge, we need to discover and represent the knowledge from raw data to a computable form for processing. Intelligent information system with the ability to process knowledge is so called a knowledge-based system.

1.2.1 Knowledge Engineering

Knowledge engineering grew out rapidly with the increased desire of knowledge-based system in the past decade. Knowledge engineering is a process to find out a way or approach to extract useful knowledge from computer data. It requires processes of analyzing and discovering patterns of data and transforming them to a format that is understandable to either human or computer, or both. Over the years, knowledge engineering researches have been focusing on the development of theories, methods, and software tools which aid human to acquire knowledge in computer. They use scientific and mathematical approaches to discover the knowledge. The approaches can be simply defined as an input-process-output system: Input – the set of computer data such as texts and database records; process – the method for the transformation of input data to knowledge; output – the desired knowledge in a specific form of knowledge representation (such as ontology).

1.2.2 Knowledge Representation

A general view of knowledge representation can be summarized in five basic principles (Randall et al. 1993):

1. A knowledge representation is a surrogate - a substitute of a thing (a physical object, event and relationship) itself for reasoning about the world.
2. A knowledge representation is a set of ontological commitments - an ontology describing existences, categories, or classification systems about an application domain.
3. A knowledge representation is a fragmentary theory of intelligent reasoning - a theory of representation that supports reasoning about the things in an application domain. An explicit axioms or computational logic may be defined for intelligent reasoning.
4. A knowledge representation is a medium for efficient computation - other than the knowledge represented logically. It also must be encoded in some sort of format, language, which enables a computer to process it efficiently.
5. A knowledge representation is a medium of human expression – a knowledge representation that can be understood by human. It is used by knowledge engineers or domain experts to study and verify the knowledge.

In the area of information system, knowledge representation defines a computable form of knowledge in Computer. It applies the theories and techniques from other fields including (Sowa 1999): 1. Logic, it defines a formal structure and rules; 2. Ontology, it defines the kinds of existence in a domain of interest; and 3.

Computation, it supports and distinguishes knowledge representation from philosophical knowledge. A knowledge representation defines different types of knowledge, typically ontologies, facts, rules, and constraints (Chein and Mugnier 2008). Knowledge is represented independently of programming logic, which means it should be defined generic enough for use with different kinds of program and system. Therefore it requires a formalized and structuralized approach to develop a valid knowledge representation.

Knowledge representation language is used to express knowledge in information system. It can be classified according to the kinds of primitives used by user (Guarino 1995), as summarized in Table 1.1. They are also described in five different levels:

1. The logical level contains the basic primitives including predicates and functions. It is level of formalization allowing for a formal interpretation of the primitives.
2. The epistemological (Brachman 1979) level is a knowledge structure to fill the gap between logical levels, which are general and abstract primitives, and the conceptual level, which is a model of specific conceptual meaning.
3. The ontological level is an ontological commitment including ontological relations, associated to an explicitly specified language primitive.
4. The conceptual level contains primitives with definite cognitive interpretation, corresponding to conceptual meaning which is language independent.
5. The linguistic level contains primitives of linguistic terms of nouns and verbs, which is language dependent.

Table 1.1 Knowledge representation formalisms (Guarino 1995)

Level	Type	Primitives	Interpretation	Main feature
1	Logical	Predicates, functions	Arbitrary	Formalization
2	Epistemological	Structuring relations	Arbitrary	Structure
3	Ontological	Ontological relations	Constrained	Meaning
4	Conceptual	Conceptual relations	Subjective	Conceptualization
5	Linguistic	Linguistic terms	Subjective	Language dependency

A visualized form of knowledge representation such as graph-based form is highly adapted to model knowledge (Chein and Mugnier 2008). Conceptual Graph is an example of graph-based knowledge representation introduced by Sowa in 1976 (Sowa 1976, 1984). Graph-based approach to knowledge modeling has the advantage of easy understanding by human. Since a graph is easy to be visualized on screen and to be understood by human, it takes advantage for control and maintenance, also for human verification and validation. Logic defined on graph-based knowledge benefits computational processing and calculation. From many researchers' opinion (Sowa 2000, Artale 1996, Guarino 1998, Harmelen et al. 2008), ontology is a relevant logical and graphical model for knowledge representation, and sometime it is also said to be a category or classification system to represent knowledge.

1.3 What Is Ontology?

“Ontology” originates from philosophy, and it has been growing into popular research in computer science and information system. In the philosopher’s perspective, for examples, Aristotle and Kant, ontology is the study of existence. It refers to a system of categories to describe the existence of the real word, or the classification of being. Although Aristotle’s ontology has been developed for more than two thousand years, his classification system is still relevant for defining nowadays ontological classification systems. Table 1.2 shows the Aristotle’s ten categories (Aristotle, Categories, 1990) to express things or existence:

Table 1.2 Aristotle’s categories

	Categories	Descriptions
1	Substance	What, or being
2	Quantity	How much
3	Quality	What kind
4	Relation	With respect to
5	Place	Where
6	Time	When
7	Position	To lie
8	State	To have
9	Action	To do
10	Affection	To undergo

Table 1.3 Kant’s categories

	Categories	Sub-categories	Descriptions
1	Quantity	Unity	Universal
		Plurality	Particular
		Totality	Singular
2	Quality	Reality	Affirmative
		Negation	Negative
		Limitation	Infinite
3	Relation	Inherence and Subsistence	Categorical
		Causality and Dependence	Hypothetical
		Community	Disjunctive
4	Modality	Possibility or Impossibility	Problematical
		Existence or Non-Existence	Assertoric
		Necessity or Contingence	Apoditic

Immanuel Kant presented a new successful categories system in 1781. The system is divided into four categories and further into three sub-categories in each main category as shown in Table 1.3. This classification system and categories

are also relevant to nowadays ontology development, especially the ontology is highly dependent on relation that describes an entity or a being. The sub-categories of this reference can be seen as different types of object, properties and relation of ontology.

Ontology is a fundamental form of knowledge representation about the real world. In the computer science perspective, ontology defines a set of representational primitives with which to model a domain of knowledge or discourse (Gruber 2008). The representational primitives of the ontology contain classes, attributes (properties) and relationships between classes. They are used to model knowledge of particular application domains.

Ontology sometimes is regarded as for conceptual analysis and domain modeling (Guarino 1998). It is used to analyze the meaning of an object in the world, of a particular domain, and provides a formal specification to describe the object. The object is being “conceptualized” in this case. Gruber (1992) provided a very short definition about ontology – “An ontology is a specification of conceptualization”. The formal specification is in support of some sort of knowledge representation model, being generated, analyzed, and processed by computer. The conceptualization has been defined in AI researches (Genesereth and Nilsson 1987, Nilsson 1991) as a structure of $\langle D, R \rangle$. The structure defines D as a domain and R as a set of relations on the domain D . This suggests that ontology and conceptualization process are created as domain dependent and relational based.

Ontology aids the development of knowledge-based system, enabling knowledge sharing and reuse. It enables intelligent communication between computers, such as the communication language used in software agents (Lee 2007). Formalized specification allows knowledge engineers to develop their own ontology by reusing and sharing with each other.

1.4 Ontology Modeling in Computer Systems

Ontology modeling in computer system, called computational ontology, is rather simpler than that in philosophy. It provides a symbolic representation of knowledge objects, classes of objects, properties of objects, and the relationships among objects to explicitly represent knowledge about an application domain. The ontology modeling is usually simplified into different kinds of mathematical definition, logical definition, or structural language.

1.4.1 Computational Ontology Representation

Computational ontology is generally represented in different kinds of abstraction: top-level ontologies, lexical ontologies and domain ontologies. They create conceptualization by defining vocabularies. The vocabularies are organized by formal relationships to create dependent linkages. Some of them are organized as a tree structure and some of them are in graph.

1.4.2 Top-Level Ontologies

Top-level ontologies (also known as upper ontologies) are limited to concepts that are universal, generic, abstract and philosophical. They are generic enough to deal with high-level abstraction and broad ranges of domain areas. Concepts defined in a top-level ontology are not specific to a particular domain (e.g. medical, science, financial), but it provides only a structure and a set of general concepts from which a domain ontology can be constructed. This top-level ontology promotes data interoperability, semantic information retrieval, automated reasoning and natural language processing.

The standard upper ontology working group (IEEE SUO WG 2003) develops a standard of upper ontology for computer application for data interoperability, information search and retrieval, natural language process (NLP), etc. Examples of existing upper ontologies include SUMO (suggested upper merged ontology) (SUMO Ontology 2004), the CYC ontology (OpenCyc 2003), and also SUO 4D ontology (SUO 4D Ontology 2005).

The SUMO has been proposed as a starter document for the SUO working group. It creates a hierarchy of top-level things as “Entities”, and subsumes “Physical” and “Abstract”. SUMO divides the ontology definition into three levels: the upper ontology (the SUMO itself), the mid-level ontology (MILO), and the bottom-level domain ontology. Mid-level ontology serves as a bridge between the upper abstraction and the bottom-level rich details of domain ontologies. Beside the upper and mid-level ontology, SUMO also defines rich details of domain ontologies including Communications, Countries and Regions, distributed computing, Economy, Finance, engineering components, Geography, Government, Military (general, devices, processes, people), North American Industrial Classification System, People, physical elements, Transnational Issues, Transportation, Viruses, World Airports A-K, World Airports L-Z, WMD (SUMO Ontology 2004).

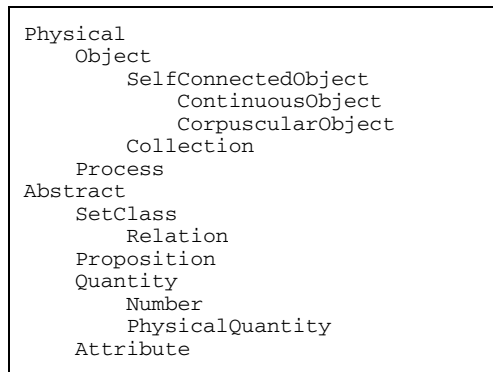


Fig. 1.1 SUMO top level

The OpenCyc is also upper-level ontology. It is some formalized common knowledge, and it models the general knowledge base and intended to solve commonsense problems. The entire Cyc ontology contains hundreds of thousands of

terms with relationship among the terms to model human consensus reality. It contains a knowledge server to serve for its Cyc knowledge base, an inference engine, and it also defines CyCL representation language for knowledge representation. It is an upper-ontology available for defining some lower level ontology knowledge such as domain specific knowledge, and domain specific facts and data as shown in the Figure 1.2 (OpenCyc 2003).

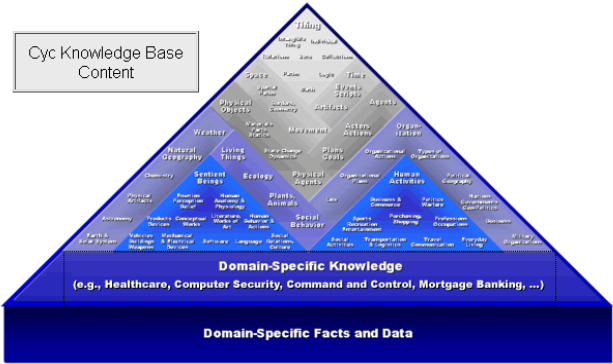


Fig. 1.2 OpenCyc upper-level ontology hierarchy (OpenCyc 2003)

1.4.3 Lexical Ontologies

A lexical ontology is an ontology describing linguistic knowledge, and it tries to model the word meaning by ontological structure. Examples of this type of ontology are WordNet (Miller 1998), and HowNet (Dong and Dong 1998). WordNet is an English based system which organizes words on the basis of lexical taxonomical semantic relationships, but the usage on WordNet is strictly limited to English based application only. HowNet is a Chinese-English bilingual lexical ontology describing the semantic relationship between concepts and the relationship between the attributes of concepts. It covers over 65,000 concepts in Chinese that are equivalent to about 75,000 concepts in English. Lexical ontology is useful for developing knowledge based system that may requires text analysis such as word sense disambiguation, word sense similarity calculation, words sense annotation and ontological annotation.

WordNet

WordNet is originally designed as a lexical database of English word (Miller 1998). It could be used as a lexical ontology to represent knowledge for computer text analysis and artificial intelligence application development, especially for many natural language related applications. Word-Net defines synsets to group English nouns, verbs, adjectives and adverb into sets of synonyms, and uses different grammatical rules of distinguishes between them (noun verbs, adjectives

and adverb). It is helpful to model concept of words and its semantic relationship. It has been used for various natural language text analyses such as word sense calculation and disambiguation.

WordNet research has been extended to ImageNet (Deng et al. 2009), a large-scale hierarchical image database. It uses the meaningful concept in WordNet connecting to image data. This is a practical example of using Word-Net as knowledge to build an intelligent information system, a concept-based image database.

HowNet

HowNet is lexical database of Chinese word developed by Dong (1998). It is a common-sense knowledge based for modeling inter-conceptual relations and inter-attribute relations of Chinese lexicons concepts and their English equivalents (Dong and Dong 1998, HowNet 2003). How-Net is aimed for Chinese language processing by using its constructed knowledge based of Chinese words. Similar to the synsets of WordNet, HowNet defines its specific Sememe-Network to model the inter-conceptual relations between Chinese lexicons concepts. HowNet is a fully computable electronic database. Knowledge of HowNet is structured by a graph. A graph based example is shown in Figure 1.3 for describing different concepts, property, attributes, and their inter-relationship. HowNet also defines a taxonomy (Figure 1.3) which serves as the upper ontology to model category for Chinese lexical concepts.

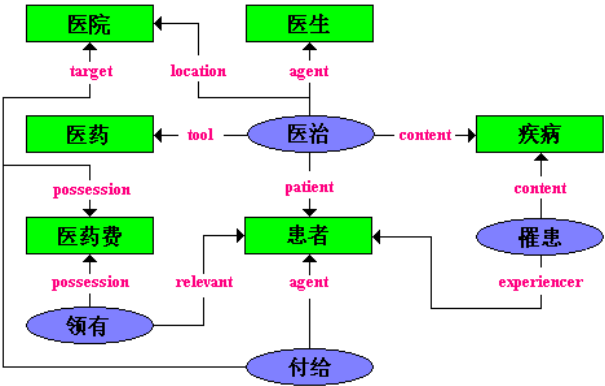


Fig. 1.3 Graphical expression of Chinese lexical concept in HowNet

Although there have been a lot of works done in these lexical ontologies, and they have also conceptualized lexical knowledge quite effectively, the main problem of these ontologies is that they are manually created in the entire process. A drawback of manual process is hard for maintenance, such as adding new knowledge, revising and updating existing knowledge. The concept and usage of words are changing all the time, so the defined knowledge of words is not permanently valid. Therefore, a continuous manually updating work and re-construction are required and thus make the process ineffective.


```

- {thing|万物} {entity|实体:{ExistAppear|存现:existent={~}}}

- {physical|物质} {thing|万物:HostOf={Appearance|外观},
  {perception|感知:content={~}}}

- {animate|生物} {physical|物质:HostOf={Age|年龄},
  {alive|活着:experiencer={~}}, {die|死:experiencer={~}},
  {metabolize|代谢: experiencer={~}},
  {reproduce|生殖:agent={~}, PatientProduct={~}}}

- {AnimalHuman|动物} {animate|生物:HostOf={Sex|性别},
  {AlterLocation|变空间位置:agent={~}}, {StateMental|精神
  状态:experiencer={~}}}

- {human|人} {AnimalHuman|动物:HostOf={Name|姓名}
  {Wisdom|智慧}{Ability|能力},
  {think|思考:agent={~}}, {speak|说:agent={~}}}

```

Fig. 1.4 Basic data - Taxonomy in HowNet

1.4.4 Domain Ontologies

A domain ontology is tied to a specific domain which can be extended from upper ontology. It should be defined for specific domains because even some huge ontology like Cyc, contains over ten thousands of concepts modeling the generic and high-level concepts, but is still not deep enough to express the conceptual and low-level of a specific domain (a domain such as medical, science, financial, etc.). In order to model a domain knowledge and make the information expressive and understandable by machines, domain ontology is developed based on the concept formation in the particular domain of interest. Domain ontology is preferably built based on an available upper ontology (e.g. SUMO, Cyc) for the ease of mapping and integration between different domain ontologies created by different specialists or researchers, as to enhance sharing and usability.

Unlike upper ontology which is usually built for reasoning commonsense knowledge, domain ontology is mainly built for reasoning a specific domain of knowledge. The domain ontology is boarder and more general for defining knowledge. In another words, domain ontology is less abstract but more specific. It is therefore more useful to build intelligent application because computer application is usually developed for particular target domains. Figure 1.5 shows an ontology tree specified for entertainment, a part of entertainment domain.

Most of the application ontologies are domain dependent, but they are shared among each other crossing over different domains. Ontology engineering usually aims to define and create domain ontologies rather than top-level ontologies and lexical ontologies, and the recent researches on domain ontology engineering will be reviewed in Chapter 2.

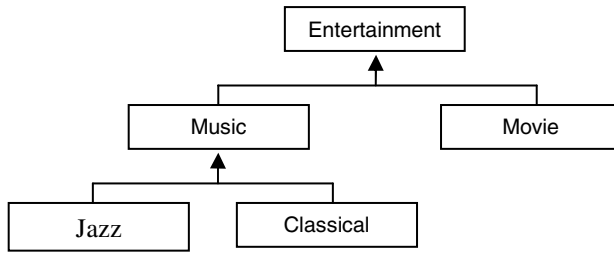


Fig. 1.5 Ontology sample of entertainment domain