Using Classified Interrelated Object Model to Represent Ontology

Abstract

Ontology representation is the essential part of the ontology learning progress. It requires a great amount of time and significant training to form this foundation stone. This paper aims to provide a straight-forward but efficient way to represent ontology. An effective data model technology – Classified Interrelated Object Model (CIOM) – is introduced and utilized to represent ontology. The main components of ontology development are elucidated and described, including ontology classes and their hierarchy, class attributes, and inter-classes relationships. This paper provides a general purpose methodology to facilitate the advanced ontology technologies.

Key words: ontology, ontology representation, semantic database model, classified interrelated object model

Author: Tao YANG
Date: Nov 23, 2008

Table of Contents

1. Introduction

In recent years the development of ontologies – explicit formal specifications of the terms in the domain and relations among them – has been moving from the realm of Artificial-Intelligenc laboratories to the desktops of domain experts (Noy, McGuinness 2005). Ontology learning greatly helps ontology engineers to construct ontologies (Maedche, Staab 2001). Ontology has been involved into a new generation – internet-based presentation – the Semantic Web. As ontology spreading to Internet, ontology becomes one of the core technologies for knowledge exchange and inter-system sharing purposes.

Ontologies are used to specify standard conceptual vocabularies in which to exchange data among systems, provide services for answering queries, publish reusable knowledge bases, and offer services to facilitate interoperability across multiple, heterogeneous systems and databases (Gruber, 2003).

Maedche and Staab proposed an effective ontology learing framework for the semantic web. Even in Noy and McGuinness' 101 guide to create an ontogoy, it still requires Protégé 2000, one of the most well-known ontology editor and knowledge-base framework, to implement this example. However, the evolvement of ontology learning is becoming so complicated and specialized that it requires a great amount of time and significant training to develop ontology with acceptable usability.

To resolve this contradiction, an effective data model technology – Classified Interrelated Object Model (CIOM) – is introduced and utilized to represent ontology. Why is a database modeling method used to represent ontology? Some of the advantages are:

- To eliminate the need to request a special designed development tools to deploy ontology applicationsTo provide a straight-forward but efficient way to represent ontology
- To take advantages of current well-developed database technologies
- To use existing database to store complicated and large-scale ontology information base.
- To offer a general purpose methodology to facilitate the advanced ontology technologies.

In the beginning of this paper, the major traits of ontology is described and discussed. Then the most impressing features of CIOM is briefed with a schema example. Finally, CIOM is catered to represent the core building blocks of ontology. Additionally, some CIOM schemas of ontology are provided for the purpose of better illustration.

2. Ontology

Sharing common understanding of the structure of information among people or software agents is the common purpose to develop ontologies (Musen 1992; Gruber 1993). People develop ontology to share common understanding under a specific context, to reuse the existing information, to explicitly express preconditions of this context, and to store this knowledge physically.

2.1 What is Ontology

In philosophy, ontology is a study subject that research on the nature of existence, their properties, and their relations. It has been applied to many other subjects. In the context of computer and information sciences, the commonly agreed definition of Ontology is proposed by Tom Gruber: an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The key role of ontologies with respect to database systems is to specify a data modeling representation at a level of abstraction above specific database designs (logical or physical), so that data can be exported, translated, queried, and unified across independently developed systems and services (Gruber, 2003).

Ontology and data models, such as Enhanced Entity-Relationship Model, are the same in terms of representing domain knowledge. But there are still some significant differences between them. First, Ontology represents a higher level of abstraction than data models. Data models are focus on current context that it elaborates to express while ontology can be shared and reused among various contexts. Second, as ontology stands at a higher abstract level it covers a larger scale of information than data models. Several data models for diverse domains might share the same ontology. Finally, the operation of ontology is usually isolated from its context while the data models are heavily related to the operation environments.

Successful applications include database interoperability, cross database search, and the integration of web services (Gruber, 2003).

2.2 Ontology Progress

Here is the step by step progress to create ontology with a given domain:

1. Define the context within the given domain.
2. Create classes and their hierarchy.
3. Identify the attributes of classes.
4. Connect classes with inter-relationships among them.

3. Classified Interacted Object Model

Semantic Database Model (SDM) is a high-level semantics-based database description and structuring formalism for database systems (Hammer, McLeod 1981). SDM captures more meaning of the real world. Comparing to other data model technologies, SDM outstands itself by its expressivity and effectiveness. Classified Interacted Object Model (CIOM) is a simplified subset of SDM, with basic structures, operations, and constraints of SDM.

The CIOM we use throughout this paper is loosely based on a SDM data model presented in “Database Description with SDM: A Semantic Database Model” (Hammer, McLeod 1981). CIOM primarily consists of classes, subclasses, and member attributes. A class is defined as a collection of entities, with a class name to identify itself from others. In CIOM, an oval with a class name written inside is denoted as this class. A subclass means specialization, viz. its membership is a subset of the members of its parent class. Member attributes are the common aspects of members of a class. In CIOM, an attribute is drawn as a pair of arrows pointing from one class to another with opposite directions. A simple CIOM schema for a class “Vehicle” is shown in Figure 1. As shown in this figure, a “Vehicle” has a unique “Vehicle Identification Number (VIN)”. The type of VIN is “String”, which implies that the VIN consists of alphanumeric characters. A “Vehicle” has two types of subclasses: “Car” and “Truck”.

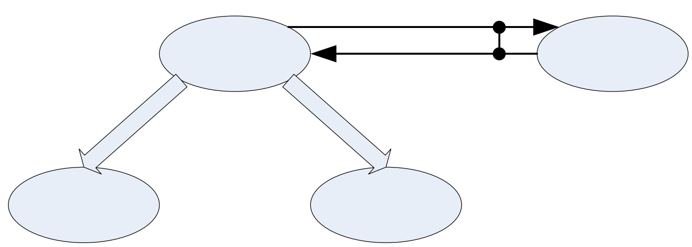

Figure 1: A CIOM schema for a class “Vehicle”

4. Represent Ontology

Ontology is a methodology to formula the definition of representational vocabulary for common sharing purpose. Ontology is a specification of conceptualization. It calls for a specific description of all kinds of entities, their properties and their relations. There are several kinds of ontology languages already developed to encode the ontology, such as Resource Description Framework and Web Ontology Language. These technologies are special designed for ontology representation. They might require familiarities of these technologies, exclusive development tools, and specially trained experts for ontology development tasks. Furthermore, ontologies developed on these tools are only able to share within the same platform, which limits their usability. In this paper, a more general and universal data model method CIOM is utilized to represent ontology.

In this representation on ontology, only the main components of ontology development are elucidated and described, including ontology classes and their hierarchy, class attributes, and inter-classes relations. It is impossible to cover all the issues that developing an ontology need to handle with. Only the major aspects of the whole development progress are illustrated and discussed into details.

4.1 Classes and Their Hierarchy

In the ontology, classes are defined to classify all kinds of existences. A class usually refers to a collection or a category of objects sharing some common character and well accepted under commonsense. All the objects under this category are normally named as "instances" of this class. Objects of the same class are also differentiated themselves by their own traits. This diversity implies that the objects are organized in hierarchy. The objects sharing the same trait of a class are grouped as instances of a subclass of this class. When classes and their hierarchy are represented in CIOM, an ontology class is an oval with a class name written inside, while a subclass is also a class but drawn as a double-line arrow pointing from its parent class. An ontology class "Vehicle" represented with CIOM is shown in Figure 2. A "Vehicle" has two types of subclasses: "Car" and "Truck", while A "Car" has two types of subclasses: "Sedan" and "Sports Utility Vehicle (SUV)".

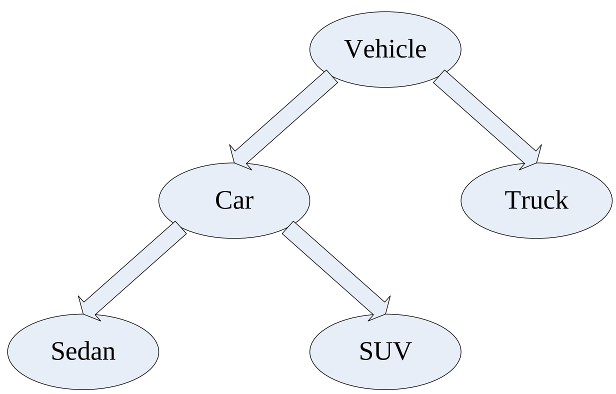

Figure 2: An ontology class "Vehicle" represented with CIOM

4.2 Class Attributes

For an ontology class, its attributes are described as all the related ontology classes, typically some built-in ontology classes. These attributes are those sharing traits that identify the class itself from other classes. The representing built-in classes include String, Number, Date, and other atomic classes. String is the collection of all alphanumeric characters. Number is the collection of all numeric values of digits. Date is the collection of all time entities in a calendar system.

Cardinality is a measure of the number of the corresponding attributes an ontology class has. The cardinality of an ontology attribute is quite different to a data model attribute. Normally speaking, the cardinality of a data model attribute is categorized into three types: One-to-One, One-to-Many, and Many-to-Many. Their meanings of these types are self-explanatory. However, an ontology attribute does not has Many-to-Many cardinality as ontology stresses a formulized conceptualization the while data model technologies emphasize on the representation of all entities. As a result, the inverse attributes of an ontology class is always has a monotony cardinality. For example, a person has attributes, such as “Social Security Number (SSN)”, “Name”, “Date of Birth (DOB)”, and “Phone Number”. A person can have several phone numbers at the same time, while a phone number can belong to several persons. For an ontology class “Person”, it is only focused on the main target “Person” while arguing its several objects of this class is meaningless as ontology is an abstraction at conceptual level. The attributes of an ontology class “Person” represented with CIOM is shown in Figure 3. When drawing a CIOM ontology class, just simply neglect the inverse attributes or assign their cardinality values to one.

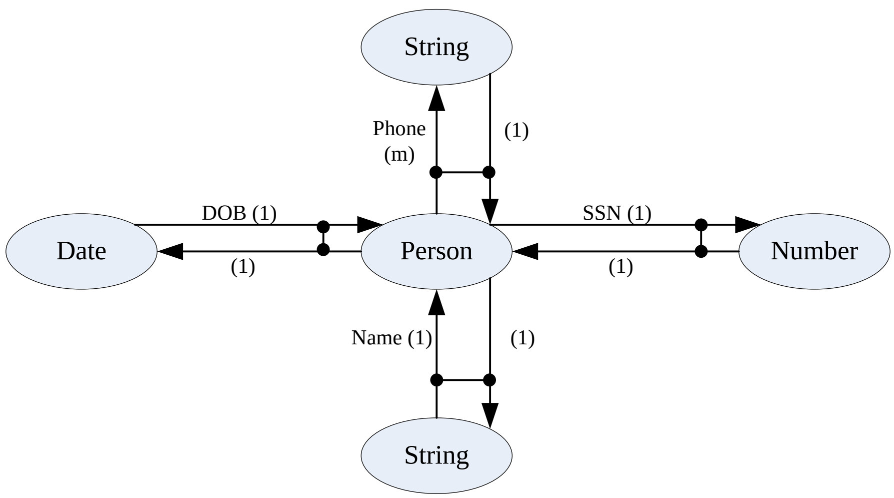

Figure 3: The attributes of an ontology class “Person” represented with CIOM

4.3 Inter-classes Relations

A relation between two ontology classes interprets how the two classes, more precisely the objects of these classes, are related. Typically a relation is a particular connection between two classes specifies how an object is connected to the other in ontology. The CIOM gains its expression power by providing an effective way for relation description. CIOM ascribes a relation between two classes as a special kind of attribute, denoted as Class Attribute. The only visible difference might be that attributes are built-in classes while the classes within relation category are abstract defined classes, normally customized by users. However, it is this difference that renders the efficiency of semantic expression of ontology.

The cardinality of a relation is categorized into three types: One-to-One, One-to-Many, and Many-to-Many. Unlike class attributes, all of them are fully supported within the same relation category. For example, a vehicle is equipped with only one engine while an engine can only equip one vehicle; a vehicle has only one owner while a person can have several vehicles at the same time; a vehicle can only be manufactured by one manufacturer while a manufacturer can produce server types of vehicles. The inter-class relations of a vehicle represented with CIOM are shown in figure 4.

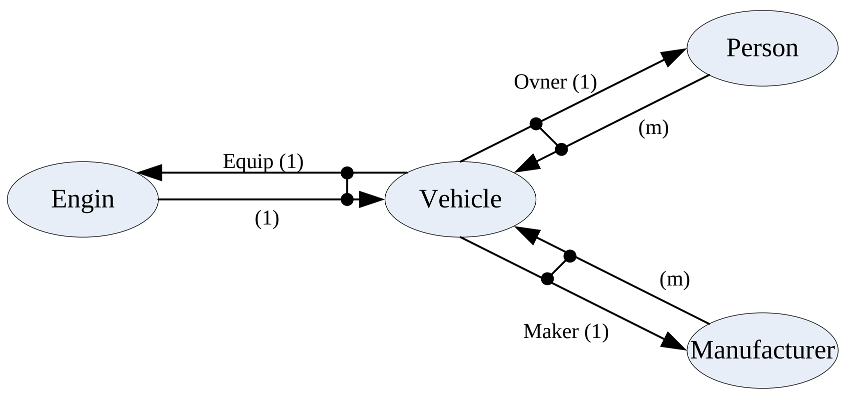

Figure 4: The inter-classes relations of a vehicle represented with CIOM

5. Conclusion

In this paper, a general purpose method, CIOM, instead of specific ontology language and develop tools, is used to represent ontology. The definition and major components of ontology are briefly discussed in the beginning. Then the most outstanding features of CIOM are also brought into instruction. Finally, this paper illustrates the progress to use CIOM to represent the major component of ontology.

The potential applications of this method include representing ontology with a general purpose modeling technology, such as EER, UML, and CIOM; storing ontology on general database, such as MySQL, DB2, and Oracle; and facilitating the sharing of ontology via general purpose tools, such as Eclipse, PowerBuilder, and Visual Studio.

Despite of the express power of CIOM, only part of CIOM features are applied in this paper. CIOM also provides operation level modeling. CIOM supports: using predicates, operations, and set-operators to define subclasses; class attribute mapping; class grouping. To apply these fruitful features of CIOM to represent ontology requires, as ever, further study.

References

1. A Maedche, S Staab. Ontology learning for the semantic web. Intelligent Systems IEEE 2001.

2. Gruber T. SOFTWARE INTELLIGENCE - MONOGRAPH - ontology progress . what is an ontology. Software World 2003;34(1):6 (2 pages).

3. M HAMMER. Database description with SDM: A semantic database model. ACM Transactions on Database Systems 1981.

4. Miguel I Aguirre-Urreta, George M Marakas. Comparing conceptual modeling techniques: A critical review of the EER vs. OO empirical literature. Database for Advances in Information Systems (New York) 2008;39(2):9 (24 pages).

5. NF Noy, DL McGuinness. Ontology development 101: A guide to creating your first ontology. Knowledge Systems Laboratory Stanford University 1993.

6. Youn, S. and McLeod, D.“Ontology Development Tools for Ontology-Based Knowledge Management,” Encyclopedia of E-Commerce, E-Government and Mobile Commerce, Idea Group Inc, 2006

7. Tharam Dillon, Elizabeth Chang, Maja Hadzic, Pornpit Wongthongtham. Differentiating conceptual modelling from data modelling, knowledge modelling and ontology modelling and a notation for ontology modelling. APCCM '08: Proceedings of the Fifth on Asia-Pacific Conference on Conceptual Modelling - Volume 79, Volume 79 ; Australian Computer Society, Inc 2008.

8. Waralak Vongdoiwang, Dencho N. Batanov. An ontology-based procedure for generating object model from text description. Knowledge and Information Systems (London) 2006;10(1):93.

Bio-statement

TAO YANG is a graduate student and student worker at University of Southern California (USC). He was employed by HSBC Software Development (Guangdong) Ltd. (HSDC) from 2005 to 2007 after his graduation from Guangdong University of Foreign Studies as an Excellent Graduate. He was Information Technology Officer of the Banking Desktop & Domino Intranet team at HSDC in 2007. He started to pursue his Master's degree in Computer Science at USC from 2008. At the same time, he worked as System Analyst & Web Developer in the IT Team of Keck School of Medicine at USC. His research interests include semantic information representation and distributed scalable systems.

Acknowledgement

I wish to thank Professor Eric Roth for his guidance and advices throughout the whole writing of this paper, thank Dr. Dennis McLeod of his intrusions on the research on this topic, and thank my wife Fengqiong Chen for her support.