Volume 6, Issue 11, November 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

On Scope of Ontology Modeling and Database Modeling

Dr. Farheen Siddiqui
Astt. Professor, Department of Computer Science
Jamia Hamdard, New Delhi, India

Abstract: Two main data models are currently used for representing knowledge and information in computer systems. A data model is a plan for building a database and is comparable to an architect's building plans. There are two major methodologies used to create a data model: the Entity Relationship (ER) approach and the Object Model. This paper will be discussed only the object model approach. The goal of the data model is to certify that all data objects required by the database are completely and accurately represented. Ontologies are objects of interest (Universal of discourse). The objective of this paper is to simplify object models compare with ontologies models.. On the other hand, ontologies have appeared as an alternative to databases in applications that require a more 'enriched' meaning. However, there is controversy regarding the best information modeling technique, as both models present similar characteristics. In this paper, we present a review of how ontology modeling and databases modeling are related, of what their main differences are and of the mechanisms used to communicate with each other.

Keywords Relational databases · Ontologies · Ontology-based databases

I. INTRODUCTION

Several models for representing machine readable information have emerged since file-based models disappeared and relational databases increased in popularity due to their efficiency, flexibility and performance for representing and managing data. Many efforts have failed to change the hegemony of relational databases by means of alternative database models that complement their lacks such as Object Oriented, Multimedia, Deductive, Spatial models, etc. Ontology is the concept which is separately identified by domain users, and used in a self contained way to communicate information. Combination of concept is the knowledge base or knowledge network. Ontologies themselves are rising as an important tool for coping with very great, compound and various sources of information. It has also been known that ontologies are advantageous for data modeling in software engineering [3]. Ontology is well known as description of declaration and abstract way the domain information of the application, it involved with vocabulary and how to constrain the use of the data [1] and they are used widely in the semantic web approach, which requires a significant degree of structure. In the area of ontology the concept have been supplemented above which allow to express the similarity of concept in ontology with object (or atom) in object oriented [2]. Ontology is actually well known in philosophy research area since 1960s, in the artificial intelligence arena, has been focused on knowledge modeling. The term ontology is used to refer to an explicit specification of a conceptualization [of a domain] is mentioned by Tom Gruber which we are already familiar with for quite sometimes. In other words, ontology refers to a formalization of the knowledge in the domain. Ontologies are used for various purposes: first is the documents in the document base are annotated and classified according to the ontology [4].

II. ONTOLOGY MODELING

Ontology describes basic concepts in a domain and defines relations among them. Basic building blocks of ontology design include: classes or concepts, properties of each concept describing various features and attributes of the concept, restrictions on slots (facets). This section will clarify each part of ontology modeling as following. 2.1. Concept/Class Concepts that are objects are likely to be best represented by classes. Classes (Concepts) are abstract groups, sets, or collections of objects. Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain [9].

2.1 Class

A class is defined for a conceptual grouping of similar terms. For example, a Person could be represented as a class which would have many subclasses such as Professor. Each class is described by a definition which specifies the slots and values that describe the class itself. They may contain individuals, other classes, or a combination of both. Ontologies vary on whether classes can contain other classes, whether a class can belong to itself, whether there is a universal class (that is, a class containing everything), etc. The concepts arranged in an inheritance hierarchy.

2.2. Slots

Facets Own slots are slots used to describe properties (or "is part of") of the term itself. For most terms, the only slots they have are own slots. The example of slots on the class Mother, subclass-of is an own slot because it is used to indicate a property of the class Mother. Whereas, has-children would be an instance slot because it describes a property of instances of the class Mother. A slot is used to describe a relationship between two terms. The first term must be an instance of the class that is the domain of the slot and the second must be an instance of the class that is the Range of the slot. For example, brother could be represented as a slot such that its Domain was Animal and its Range was Male-Animal. A slot may also be referred to as a binary relation . They are attached to classes or slots and contain meta-information, such as comments, constraints and default values. Slots represent the attributes of the classes. Possible slot types are primitive types (Integer, Boolean, String etc.), references to other objects (modeling relationships) and sets of values of these types [7]. For each Professor, we want to know her address and salary, and what course she works for. As we continue to generate terms, we are completely defining the scope of ontology, by what we finally decide to include and what to exclude. For example, in order to verify the status of Professor, a Professor has to work for at least one Course. Each Professor which inherits the slot from concept Person has an address slot of type String. Properties of the classes, such as age or address can be represented by slots, and restrictions on properties or relationships between classes and or slots, are represented by slot facets.

2.3. Axioms

An axiom is a sentence in first order logic (F-logic) that is assumed to be true without proof. In practice, axioms can be used to refer to the sentences that cannot be represented using only slots and values on a frame2 . Axioms must be entered in prefix notation. Use => to indicate logical implication, <=> to indicate logical equivalence, and to indicate conjunction, or to indicate disjunction, not to indicate negation, and exists to indicate existential quantification2 . For example in OWL, it is possible to state that two or more classes are equivalent3 , axiom ::= 'EquivalentClasses(' classID classID { classID } ')'

2.4. Instances

Instances represent specific entities from the domain knowledge base (KB). An example KB based on the Person ontology might contain the specific Professor Jane and Course SP233 2 http://www-ksl-svc. stanford.edu:5915/doc/frame-editor/glossary-of-terms.html 3 http://www.w3.org/TR/owl-semantics/syntax.html#2.3.1

Database. Individuals (instances) are the basic components of ontology. The individuals in ontology may include concrete objects such as people, animals, tables, automobiles, molecules, and planets, as well as abstract individuals such as numbers and words.

2.5. Relationships

A relation is used to describe a relationship among two or more terms. If a relation represents a relationship between only two terms, it is called a slot or a binary relation. If the relation describes a relationship among n terms such that there is a unique nth term corresponding to any set of the first n-1 terms, then the relation is called a function (section 2.6). For examples of binary relations include: subclass-of and connected-to. If we introduce relationships to ontology, we find that this simple and well-designed hierarchy structure becomes complex and significantly more difficult to interpret manually. It is not difficult to understand why an entity that is described as 'part of' another entity might also be 'part of' a third entity. Consequently, entities may have more than one parent. The structure that emerges is known as a directed acyclic graph (DAG4). As well as the standard is-a and part-of relationships, ontologies often include additional types of relation that further refine the semantics they model. These relations are often domain-specific and are used to answer particular types of question. The is-a relationship can be used to inherit attributes and semantic relationships down (against the direction of the arrows) from higher nodes to lower nodes in the DAG. This is very similar to inheritance in Object Oriented Databases such as C++5 . Example: If Person has the attribute Name then Professor would inherit Name. We don't have to specify that Professor has Name.

Conceptual Database Modeling

Another topic we need to address is the difference between conceptual model and conceptual schema. As Hirschheim et al. [5] mention, there is some confusion about the terms model and schema in the field. A model is "a set of conceptual and notational conventions which help to perceive, organize and specify some data" [6]. Entity-Relationship (ER) , Object Modeling Technique (OMT) [7], and Unified Modeling Language are examples of conceptual models. Conceptual schemas, on the other hand, refer to the result of the modeling, namely a set of diagrams that use a given conceptual model as a language to express the specific data structures for an application that is going to be developed. Wand & Weber [8] take further the distinction between model and schema. They propose a framework set in what they call a conceptual-modeling context. In this context we use a conceptual-modeling method that applies a conceptual-modeling grammar to model real-world phenomena and create a conceptual-modeling script.

III. DIFFERENT SCOPES FOR ONTOLOGIES AND DATABASE SCHEMAS

There is a continuum from the developer/user conceptualizations, later expressed in informal and formal languages, to the creation of conceptual schemas and the subsequent representation of facts in a database. However, in order to achieve our objectives with IS, we need to separate things in clearly defined levels. We propose that computational ontologies and conceptual schemas definitely should belong to two different epistemic levels. The search for data independence and the creation of more generic and abstract models which started with two level architecture continued with the three-level architecture in the ANSI/X3/SPARC. Techniques and tools for information systems design and analysis are always in constant evolution. Automatic generation of conceptual schemas from ontologies [9,10] and the creation of conceptual schemas that are more independent from the implementations [11,12] are examples of cases in which the borders of these two epistemic levels sometimes are blurred. At the same time that the research in conceptual modeling argues for the creation of more generic models, the research on ontologies sometimes (wrongly in our point of view) go to specifics such as having instances of classes within ontologies. As McGuinness [13] points out, "some classification schemes only include class names while others include ground individual content". some medical ontologies names represent classes or universals while in other ontologies names convey how reality is perceived, measured, and understood. We argue that the recording of instances or ground content should be done by the IS itself under the guidance of the conceptual schema. In general, ontologies should not include instances of its concepts. The recording of facts belongs to the domain of measurement (i.e., the level of conceptual schemas). Regarding the scope of ontologies and conceptual schemas, some researchers suggest that ontological research is the study of instrumentally useful formal models, not of the formal properties of reality. Theory-focused ontologies are the ones created using philosophical theory. Pragmatically-oriented ontologies are very common in the practice of information systems and are targeted to specific domains such as banking or taxation. Ontologies often combine a philosophical approach with pragmatic purposes. This odd combination may result in incompatible philosophical underpinnings being used in the same pragmatically-oriented ontology. The pragmatic approach allows the investigator to deftly avoid users' objection that a given ontology does not coincide with his or her view of reality. The developer of ontologies can reply that what is at issue is not whether the ontology is correct, but whether the models it defines are useful, or adequate, for some (limited) purposes. In this case, an ontology may have relevance to a narrow range of problems without being correct. We strongly disagree with this strategy [Fonseca and Martin, 2005]. Adjustment of an IS to local conditions is the role to be fulfilled by conceptual schemas. An excessively narrow view of ontologies will lead to the same problems in modeling that led to the very emergence of ontologies in ISAD research in the first place. These were the problems uncovered by Guarino [1998] in his introduction of Ontology-driven Information Systems. From the discussion above we can begin to understand why it is implicit in Sowa's argument that a conceptual schema (the result of the modeling process) is essentially the same as a computational ontology (the result of the ontology engineering process). Other researchers have asked themselves whether ontologies were actually the well-known conceptual data modeling techniques in disguise.

IV. CONCLUSION

The origin of the confusion lies not only in the similarity of the two concepts but also to the fact that research on the use of ontologies in information systems is a relatively recent effort. The distinction between the two terms is needed because a misuse or misunderstanding of the terms will hinder progress both in the research and in the practice of software development. Ontologies actually reflect the result of the evolution of research in modeling. This paper make a similar comparison on the scope of ontologies (reality driven) and conceptual schemas to highlight the differences between them.

REFERENCES

[1] Z. Hu, "Using Ontology to Bind Web Services to the Data Model of Automation Systems", In Journal Lecture Notes in Computer Science 0302-9743 (Print) 1611-3349 (Online) Springer Berlin / Heidelberg Volume 2593/2003, pp 154-168.

[2] W. V. Siricharoen, "Ontologies and Software Engineering " In Proceeding International Conference on , Computational Science (ICCS 2007) Lecture Notes of Computer Science, May 27-30, 2006, Beijing, China, Volume 44881105, pp. 1151-1162. [5] S. Cranefield, M. Purvis, "UML as an

[3] W. V. Siricharoen, "Ontologies and Object models in Object Oriented Software Engineering" In International Association of Engineers (IAENG) Journal of Computer Science (JICS) Volume 33(1), pp. 1151-1162, 2006, [Online] available: http://www.iaeng.org/IJCS/issues_v33/issue_1/IJCS_33_1_4.pdf

[4] Davies, J., Studer, R., and Warren, P., Semantic Web Technologies Trends and research in ontology-based systems, John Wiley & Sons Inc., West Sussex, England, 2006.

[5] Hirschheim, R. A., H.-K. Klein, and K. Lyytinen (1995) Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations. Cambridge; New York: Cambridge University Press.

[6] Jarrar, M., J. Demey, and R. Meersman (2003) "On Using Conceptual Data Modeling for Ontology Engineering," Journal on Data Semantics

[7] Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy et al. (2003) Object-Oriented Modeling and Design. Englewood Cliffs, NJ: Prentice-Hall.

[8] Wand, Y. and R. Weber (2004) "Reflection: Ontology in Information Systems," Journal of Database Management (15) 2, pp. iii-v

[9] Fonseca, F. and J. Martin (2005) "Toward an Alternative Notion of Information Systems Ontologies: Information Engineering as a Hermeneutic Enterprise," Journal of the American Society for Information Science and Technology (56) 1, pp. 46-57.

[10] Shanks, G., E. Tansley, and R. Weber (2003) "Using ontology to validate conceptual models," Communications of the ACM (46) 10, pp. 85-89.

[11] Caruso, F. and A. Umar (2004) "Architectures to Survive Technological and Business Turbulences," Information Systems Frontiers (6) 1, pp. 9-21

[12] Zuniga, G. (2001) Ontology: its transformation from philosophy to information systems. The 2nd international conference on Formal Ontology in Information Systems (FOIS'01), Ogunquit,ME, 2001, pp. 187-197.

[13] McGuinness, D. L. (2003) Ontologies Come of Age, in D. Fensel, W. Wahlster, H. Lieberman, and J. Hendler (Eds.) Spinning the semantic Web: bringing the World Wide Web to its full potential, Cambridge, Mass.: MIT Press, pp. xxiii, 479.

