DIDACTICA MATHEMATICA, Vol. 31(2013), N° 1, pp. 43–46

ONTOLOGIES IN COMPUTER SCIENCE

Diana Man

Abstract. Research on ontology is becoming increasingly widespread in the computer science community, and its importance is being recognized in a multiplicity of research fields and application areas, including knowledge engineering, database design and integration, information retrieval and extraction.

MSC 2000. 34C40.

Key words. Ontology, natural language understanding, knowledge-based problem solving

1. INTRODUCTION

"Ontology is an explicit specification of conceptualization. The term is borrowed from philosophy, where Ontology is a systematic account of Existence" (Tom Gruber and the Knowledge Systems Laboratory at Stanford University).

In computer science, ontology is a formal representation of the knowledge by a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain and may be used to describe the domain. In theory, ontology is a "formal, explicit specification of a shared conceptualization". Ontology provides a shared vocabulary, which can be used to model a domain that is, the type of objects, and/or concepts that exist, and their properties and relations.

Ontologies are used in Artificial Intelligence, the Semantic Web, Systems Engineering, Software Engineering, Biomedical Informatics, Library Science, Enterprise Bookmarking, and Information Architecture as a form of knowledge representation about the world or some part of it. The creation of domain ontologies is also fundamental to the definition and use of an enterprise architecture framework. There are four categories of ontology: static, dynamic, intentional and social. Static ontology describes things that exist, their attributes and relationship. Dynamic ontology describes the world in terms of states, state transitions and processes. Intentional ontology encompass the world of agents, things believe in, want, prove or disprove and argue about. Social ontology covers social settings, permanent organizational structures or shifting networks of alliances and independencies [1].

2. DESIGN CRITERIA FOR ONTOLOGIES

A preliminary set of design criteria for ontologies whose purpose is knowledge sharing and interoperation among programs based on a shared conceptualization are:

a) Clarity. An ontology should communicate the propose meaning of defined terms. Definitions should be objective and should be independent of social and computational context. A complete definition (a predicate defined by necessary and sufficient conditions) is preferred over a partial definition (defined by only necessary or sufficient conditions). All definitions should be documented with natural language.

b) Coherence. Ontology should permit inferences that are consistent with the definitions. At the least, the defining axioms should be logically consistent. Coherence should also apply to the concepts that are defined informally, such as those described in natural language documentation and examples. If a sentence that can be inferred from the axioms contradicts a definition or example given informally, then the ontology is incoherent.

c) Extensibility. Ontology should be designed to anticipate the uses of the shared vocabulary. It should offer a conceptual foundation for a range of anticipated tasks and the representation should be crafted so that one can extend and specialize the ontology monotonically. In other words, one should be able to define new terms for special uses based on the existing vocabulary, in a way that does not require the revision of the existing definitions.

d) Minimal encoding bias. The conceptualization should be specified at the knowledge level without depending on a particular symbol-level encoding. An encoding bias results when representation choices are made purely for the convenience of notation or implementation. Encoding bias should be minimized because knowledge-sharing agents may be implemented in different representation systems and styles of representation.

e) Minimal ontological commitment. Ontology should require the minimal ontological commitment sufficient to support the intended knowledge sharing activities. Ontology should make as few claims as possible about the world being modeled, allowing the parties committed to the ontology freedom specialize and instantiate the ontology as needed [2].

3. THE IMPACT OF ONTOLOGY ON INFORMATION SYSTEM

Every information system has its own ontology. When discussing the impact an ontology can have on an Information System, we can differentiate two orthogonal dimensions: a temporal dimension, concerning whether an ontology is used at development time or at run and a more structural dimension, regarding the particular way an ontology can affect the main Information System components [3].

3.1. The temporal dimension: using ontologies at development time vs. run time. When the ontology is used by an Information System at run time, we speak of an "ontology-driven Information System" proper; when it is used at development time, we speak of "ontology-driven Information System development". In the run time context, we can distinguish two different scenarios. In the first scenario, we have a set of reusable ontologies at our disposal, organized in an ontology library containing domain and task ontologies. In the second scenario, the degree of reusability is very limited, as we only have a very generic ontology, consisting of coarse domain-level distinctions among the basic entities of the world and meta-level distinctions about kinds of class and kinds of relation. In the development time context we must distinguish an ontology-aware Information System from an ontology-driven Information System: in the first case, an Information System component is just awake of the existence of an ontology and can use it for whatever specific application reason is needed. In the second case, the ontology is just another component, cooperating at run time towards the "higher" overall Information System goal. An important motivation for using an ontology at run time is enabling the communication between software agents. Software agents are communicating with each other via messages that contain expressions formulated in terms of an ontology (ontology-driven communication). In order for a software agent to understand the meaning of these expressions, the agent needs access to the ontology they give to [4].

3.2. The structural dimension: impact of ontologies on IS components. The most evident use of an ontology is in association with the database component. At the development time, an ontology can play an important role in the requirement analysis and conceptual modelling phase, especially if integrated with lexical. At run time, there are many ways in which ontologies and databases can cooperate. The availability of explicit ontologies for information resources is at the core of the mediation-based approach to information integration. Ontologies can support "intensional queries" concerning the content of a particular database or dynamic management of queries concerning multiple databases [5]. Maybe not so evident, but however very important, is the use of an ontology in connection with the user interface component. At run time, the first role an ontology can play within the user interface is to allow itself to be queried and browsed by the user. In this case, the user is awake of the ontology, and uses it as part of his normal use of the Information System. In this way, the user can browse the ontology in order to better understand the vocabulary used by the Information System, being able therefore to formulate queries at the desired level of specificity. Application programs are still an important part of many Information System. They usually contain a lot of domain knowledge, which, for various reasons, is not explicitly stored in the database. At the development time, an Information System developer can generate the static part of a program with help of an ontology. At run time, we may decide to represent explicitly all the domain knowledge implicitly encoded in the application program, turning the program in a knowledge-based system. As well known, this has large benefits from the point of view of ease-of-maintenance, extensibility and flexibility. In this case, the knowledge base could be constituted by a core knowledge base plus an ontology. Ontologies can help therefore to increase the transparency of application software [3].

4. CONCLUSION

The ontology has been applied in many different ways. The core meaning within computer science is a model for describing the world that consists of a set of types, properties, and relationship types. There is also generally an expectation that there be a close resemblance between the real world and the features of the model into ontology.

REFERENCES

[1] Gruber, T., *A translation approach to portable ontologies*, Knowledge Acquisition, **5** (1993), 199–220.

[2] Gruber, T., *Toward principles for the design of ontologies used for knowledge sharing*, International Journal of Human-Computer Studies, **43** (1965), 907–928.

[3] Guarino, N., *Formal Ontology in Information Systems*, Proceedings of FOIS, 1998, 3–15.

[4] Van Heijst, G., Schreiber, A. T.and Wielinga, B. J., *Using Explicit Ontologies in KBS Development*, International Journal of Human and Computer Studies, **46** 1997 183–292.

[5] Miller, G. A, *WORDNET: A Lexical Database for English*, Communications of ACM, **11** 1995 39–41.

BABES-BOLYAI University of Cluj-Napoca
Faculty of Mathematics and Computer Science
e-mail: mandiana77@yahoo.com

Received: August 1, 2010