Domain Ontologies

Matteo Cristani
University of Verona, Italy

Roberta Cuel
University of Verona, Italy

INTRODUCTION

In conceptual modeling we need to consider a general level of abstraction where the domain of interest is formalized in an independent way with respect to the specific application for which the conceptual modeling process is performed. This leads to an integrated approach that takes into account knowledge about a domain and metaknowledge about a methodology. Indeed, knowledge about a domain is represented by a system of concepts and instances that reify the knowledge that is managed within a domain, and the metaknowledge about a methodology is the description of the knowledge deriving from the method used. For instance, when a technology is used to unveil ontologies within a specific domain, the knowledge about the domain is the resulting ontology, and the metaknowledge about a methodology is the description of the method used to construct the ontology. In this article, a novel method for the creation of both upper level and specific domain ontologies, called the bidirectional method for developing ontologies, is described. In particular, it will guide the developer to obtain ontologies resulting from the combination of both top-down and bottom-up approaches. The first one focuses on conceptual modeling through "armchair" research (philosophical, psychological, sociological aspects) and figures out a formal draft schema. The second approach employs an automatic (or semiautomatic) extraction of categories, taxonomies, partonomies, and dependency graphs in particular from linguistic corpora of documents related to the topics of the domain.

BACKGROUND

Formal ontologies are a popular research topic in several communities, such as knowledge management, knowledge engineering, natural language processing, artificial intelligence (AI), and others (Fensel, 2000). Formal ontology can be defined as the systematic, formal, axiomatic development of the logic of all forms and modes of being (Cocchiarella, 1991). More generally, we employ the term formal ontology to designate an explicit specification of a shared conceptualization that holds in a particular context. In other words, an ontology provides an explicit conceptualization that describes semantics of data, providing a shared and common understanding of a domain (from an AI perspective, see the definitions of Gruber, 1998, and Jasper & Ushold, 1999). Ontologies are used to manage knowledge within and among communities, to manage and organize corporate knowledge bases, and to negotiate meanings among individuals. Moreover, ontologies are used to share knowledge among people, and heterogeneous and widely spread application systems, such as semantic-Web applications (Schwartz, 2003). They are implied in projects, as conceptual models, to enable content-based access on corporate knowledge memories, knowledge bases, or data warehouses. They are employed to allow agents to understand each other when they need to interact, communicate, and negotiate meanings. Finally, they refer to common information and share a common understanding of their structure.

In computer science, knowledge management, knowledge representation, and other fields, several languages and tools exist for helping final users and system developers in creating good and effective ontologies. In particular, various tools help people in manually or semiautomatically creating categories, partonomies, taxonomies, and other organization levels of ontologies. The generally accepted term to designate these tools is ontology editors. Some of them are open source such as Protégé-2000, KAON, and SWOOP, and others are commercial suites for knowledge management based on ontology development, such as tools provided by the onto-Knowledge Project (for an in-depth description, see http://protege.stanford.edu, http://kaon.semanticweb.org/, http://www.mindswap.org/2004/SWOOP/, http://www.ontoknowledge.org/index.shtml).

Some Important Methodologies

Behind these tools and techniques, different (domain-independent) approaches and methods are used to develop numerous heterogeneous ontologies. In particu-

lar, Ushold's (2000; who proposed codification in a formal language) methodology and methontology, which constructs an ontology in a sequence of intermediate representations finally translated into the actual object (Fernández, Gòmez-Pérez, & Juristo, 1997), are the most representative. Here are short descriptions of some important methodologies:

- One of the first modules of the foundational ontologies library is the descriptive ontology for linguistic cognitive engineering (DOLCE). DOLCE is an ontology of particulars and refers to cognitive artefacts that depend on human perception, cultural imprints, and social conventions. This ontology derives from armchair research in particular, referring to enduring and durable entities from philosophical literature. The main authors' idea is to develop not a monolithic module, but a library of ontologies (WonderWeb Foundation Ontologies Library) that allows agents to understand one another despite enforcing them to interoperate by the adoption of a single ontology (Masolo, Borgo, Gangemi, Guarino, & Oltramari, 2002). Finally, basic functions and relations (according to the methodology introduced by Gangemi, Pisanelli, & Steve, 1998) should be general enough to be applied to multiple domains, be sufficiently intuitive and well studied in the philosophical literature, and hold as soon as their relations are given without mediating additional entities.
- In Gatius and Rodríguez (1996), the authors developed a three-step process (natural-language interface generator [GISE]) to build a domain ontology: the building and maintenance of general linguistic knowledge, a definition of the application in terms of the conceptual ontology, and a definition of the control structure. It includes the metarules for mapping objects in the domain ontology with those in the task ontology, the metarules for mapping the conceptual ontology onto the linguistic ontology, and those for allowing the generation of the specific interface knowledge sources, mainly the grammar and the lexicon.
- One of the most famous ontology-design environments is methontology. It tries to define the necessary activities that people carry out when building an ontology (Fernández et al., 1997). In other words, it is a flow of ontology development for three different processes: management, technology, and support. The ontology-development process is composed of the following steps: project-management activities that include planning, control, and quality assurance; development-oriented activities that include specification, conceptualization, formalization, and implementation; and activities that include knowledge acquisition, evaluation, integration, and documentation.
- The authors Lauser, Wildemann, Poulos, Fisseha, Keizer, and Katz (2002) use the multilingual methontology methodology defined by Fernández et al. (1997), and enrich this one by stressing on specific actions for supporting the creation process for ontology-driven conceptual analysis. The domain ontology is built by using two different knowledge-acquisition approaches: the creation of the core ontology and the derivation of the domain ontology from a thesaurus. The first one is basically comprised of the first three steps of methontology-development activities defining a list of frequent terms and a list of domain-specific documents to analyze. The second one consists of descriptive keywords linked by a basic set of relationships. The goal of this step is to refine an RDFS ontology model to develop a pruned ontology and a list of frequent terms.
- Toronto Virtual Enterprise (TOVE) is a methodology for ontological engineering that allows the developer to build ontology following these steps: scenarios motivation, ontology requirements definitions, terminology specification, formal description requirements, axiom specification, and completeness theorems (Fox & Gruninger, 1994, 1998).
- Ontology Development 101 has been developed by authors involved in these ontology-editing environments: Protégé-2000, Ontolingua, and Chimaera (Noy & McGuinnes, 2001). They propose a very simple guide, based on iterative design, that helps developers to create an ontology using these tools. The sequence of the steps to develop an ontology are to determine the domain and scope of the ontology, consider reusing existing ontologies (e.g., Ontolingua ontology library, DAML ontology library, UNSPSC, RosettaNet, and DMOZ), enumerate important terms in the ontology, define the classes and the class hierarchy, define the properties of class slots, define the facets of the slots, and create instances.
- Ushold's (2000) methodology uses formal language for building ontologies via a purely manual process, identifying purpose and scope, capturing (the identification of key concepts and relationships, and the provision of definitions), and finally coding ontology (committing to the basic terms