Ontology and Expertise Map Building in Virtual Organizations

Gila Molcho
Technion—Israel Institute of Technology, Israel

Anca Draghici
Politehnica University of Timisoara, Romania

Nada Matta
University of Technology of Troyes, France

INTRODUCTION

When organizations collaborate in virtual space, a common frame of reference, or at least a common terminology, is necessary for human-to-human, human-to-machine, and machine-to-machine communication. Similarly, within a core organization characterized by distributed collaboration between remote sites and research or production units, a common understanding of reference terms is indispensable. Yet this common understanding of terms is often implicit at best and frequently not present at all. Misunderstandings between distributed team members and faulty translations of software applications contribute to the rising costs of interoperability in virtual, distributed organizations. Indeed, the growing implementation of distributed software agents necessitates developing and adopting a shared terminology and syntax for efficient and effective interoperability.

Ontology offers a solution for solving the interoperability problems brought about by semantic obstacles, that is, obstacles related to definitions of business and scientific terms and software classes. An ontology is a taxonomy of concepts and their definitions supported by a logical theory. It is often captured in the form of a semantic network—a graph whose nodes are concepts or individual objects and whose arcs represent relationships or associations among the concepts (Huhns & Singh, 1997).

Ontologies have been defined as explicit specifications of a particular conceptualization (Gruber, 1993). They aim at explicating the knowledge for a particular domain contained within software applications and/or within an organization and its business procedures. An ontology expresses, for a particular domain, the set of terms, entities, objects, and classes and the relationships among them, and provides formal definitions and axioms that constrain the interpretation of these terms (Gomez-Perez, 1998).

Ontologies facilitate a rich variety of structural and nonstructural relationships, such as generalization, inheritance, aggregation, and instantiation. They can supply a precise domain model for software applications and include frameworks for modeling domain knowledge and agreements about representations (Huhns & Singh, 1997). For instance, an ontology can provide the object schema of object-oriented systems and class definitions for conventional software (Fikes & Farquhar, 1999).

Ontological definitions, written in a variety of logical languages, are human-readable. They can also automatically infer translation engines for software applications. By making explicit the implicit definitions and relations among classes, objects, and entities, ontologies contribute to knowledge sharing and reuse (Gomez-Perez, 1998).

Ontologies may differ not only in their content but also in their structure and implementation. Various methodologies exist to guide the theoretical approach taken, and numerous ontology-building tools are available. The problem is that these procedures have not coalesced into popular development styles or protocols, and the tools have not yet matured as in other software practices. However, an ontology is typically built in more or less the following manner (Denny, 2002):

- **Acquire domain knowledge:** Assemble appropriate information resources and expertise that will define, with consensus and consistency, the terms formally used to describe things in the domain of interest. These definitions must be collected so that they can be expressed in a common language selected for the ontology.

- **Organize the ontology:** Design the overall conceptual structure of the domain. This will likely involve identifying the domain's principal concrete concepts and their properties, identifying the relationships among the concepts, creating abstract concepts as organizing features, referencing or including supporting ontologies, distinguishing which concepts have instances, and applying other guidelines of the chosen methodology.
- **Flesh out the ontology :** Add concepts, relations, and individuals to the level of detail necessary to satisfy the purposes of the ontology.
- **Check the work:** Reconcile syntactic, logical, and semantic inconsistencies among the ontology elements. Consistency checking may also involve automatic classification that defines new concepts based on individual properties and class relationships.
- **Commit to the ontology:** Incumbent on any ontology development effort is a final verification of the ontology by domain experts and subsequent commitment to the ontology by publishing it within its intended deployment environment.

Based on these considerations and perspectives, the present chapter will outline the detailed approach used to build the ontology and complementary knowledge map for a particular virtual organization, the Virtual Research Laboratory for a Knowledge Community in Production (VRL-KCiP), a network of excellence established in the context of the 6th Framework Programme (www.vrl-kcip.org).

ONTOLOGY IN THE VRL-KCIP BACKGROUND

The central aim of the VRL-KCiP is to create synergy by integrating the research expertise and capabilities of the different member teams to support product life cycle engineering in the modern manufacturing environment. Hence, knowledge sharing and collaborative research constitute the core competency and potential for the network's success and the essence of its existence.

Among the central activities required to fulfill the VRL-KCiP vision are: (a) building an ontology with the purpose of generating a common reference language among the member teams that can overcome differences in culture, location, language, and fields of expertise; (b) implementing a central knowledge management system (KMS) that will allow expertise-identification and knowledge-sharing capabilities; and (c) implementing IT-enabled one-to-one or many-to-many communications capabilities to complement the face-to-face meetings of the distributed network. Because of the potential major impact of ontology development on the success of the network, significant effort was invested in completing this task efficiently and effectively.

Ontology-building focuses on what the ontology is required *for* (Gruber, 1993). The VRL-KCiP ontology was developed to enable knowledge sharing and reuse. Initially, the ontology had two objectives: (1) to ensure a common understanding of specific terms describing members' fields of expertise and research relevant to state-of-the-art life cycle engineering; (2) to provide the structure of the VRL-KCiP knowledge map. The goal of the knowledge map was to enable explicit charting of member expertise to clearly define and locate experts within the network and to develop a concise core competency depiction (Molcho, 2005).

The need for the VRL-KCiP ontology was magnified by the nature of the network—a virtual multilingual, multidisciplinary, multicultural dispersed research team, researching the state-of-the-art in the vast field of life cycle engineering that, contrary to most virtual enterprises, did not evolve gradually from a central core but rather emerged as a *fait accompli*.

During the process of developing the ontology it rapidly became evident that in addition to the points outlined above, the ontology would provide the structured context required to cultivate a high quality knowledge base for capturing, accessing, archiving, and validating knowledge objects in the VRL-KCiP knowledge management system (KMS).

The following discussion focuses on the main stages in achieving the above goals.

Stage I: Goal and Methodology Definition

Ontology definition is an art. Even with the aid of the many tools developed to help build ontologies, the process is often based on years of research. While the consensus was that the results of ontology-building were central to the success of the network, lengthy research was not viable since the ontology was required for the VRL-KCiP network to function. Therefore, compromises had to be made. As a result, although the goals of building the ontology were clearly defined in the initial