A Software Modeling Approach to Ontology Design via Extensions to ODM and OWL

Rishi Kanth Saripalle, Department of Computer Science & Engineering, University of Connecticut, Storrs, CT, USA
Steven A. Demurjian, Department of Computer Science & Engineering, University of Connecticut, Storrs, CT, USA
Alberto De la Rosa Algarín, Department of Computer Science & Engineering, University of Connecticut, Storrs, CT, USA
Michael Blechner, Department of Pathology, University of Connecticut Health Center, Farmington, CT, USA

ABSTRACT

Ontologies are built to establish standard terminologies representing a semantic agreement between humans and knowledge systems via representational frameworks (e.g., KIF, DAML+OIL, OWL, etc.) that have been proposed in the research community, with limited adoption in industry. One possible reason is a lack of a formal model and associated process to more precisely and accurately design and develop ontologies. The authors’ prior work explored UML, entity-relationship diagrams, and XML as compared to RDF and OWL, identifying modeling capabilities lacking in ontologies. In all three approaches, design precedes instantiation which contrasts with ontology developers who build ontologies at the application level targeted to a specific domain. The paper proposes design-level modeling enhancements to ontologies by extending the OMG Ontology Definition Model (ODM) and OWL grammar with capabilities from the three aforementioned approaches, promoting a software engineering-based process. As a result, this work provides a more software engineering-oriented process to ontology design and development.

Keywords: Ontology Design and Development, Ontology Domain Profile, Ontology Life Cycle Model, Ontology Modeling, Standard Terminologies

1. INTRODUCTION

Software domain modeling is a process where individuals seek to conceptualize an application in order to arrive at a solution that meets all of the application’s domain needs and requirements. Conceptualization can be defined as a structure \( \{D, R\} \), where \( D \) is the domain (scope of the application) and \( R \) is the set of relevant relations (functionality and interactions of the application) in the \( D \) (Gruber, 2005). Currently, one popular domain conceptualization

DOI: 10.4018/jswis.2013040103
approach is ontologies where the representation of knowledge occurs via the definition of concepts that can be related to one another for sharing and reuse of that knowledge. In practice, W3C supports two major ontology development frameworks, the Resource Description Framework (RDF) (Powers, 2003) and the Web Ontology Language (OWL) (Allemang & Hendler, 2011), where the latter is built on top of the former. Additionally, a wide range of knowledge representational frameworks (e.g., KIF (Genesereth, 1991), KL-ONE (Brachman & Schmolze, 1985), DAML+OIL (Horrocks, 2002), etc.) are available. When reviewing the usage of these frameworks in real applications, the overriding theme is realizing a specific instance-level solution for a particular system, rather than designing a solution that can be reused by multiple similar applications within the same general domain. For example, in health information technology (HIT), multiple systems such as AllScripts (AllScripts, 2012), Centricity (GE Centricity, 2012), MS Health Vault (Microsoft Health Vault, 2013), etc., are developed with their own specific (and different) medical ontologies despite the fact that each one is storing and managing patient data. Any attempt to integrate data from across multiple EHRs, referred to as health information exchange (HIE) requires integrating their different medical ontologies which at best is a semi-automated and arduous task. While many of these HIT systems leverage existing standard medical terminologies (e.g., LOINC (LOINC, 2001), UMLS (Bodenreider, 2004), ICD (ICD, 2009), etc.), there is no uniform attempt to design an ontology domain model for medical knowledge that is general purpose and reusable in multiple contexts. For example, one HIT system may organize medical knowledge in an ontology by disease, symptoms (for each disease), diagnoses (for each disease), and treatments (for each disease), while another HIT system might invert and target this information from a symptom to diagnosis to disease to treatment basis. Attempting to reconcile this medical knowledge in two or more different ontologies is an arborous, time-consuming, and at best semi-automated task, making HIE difficult to achieve.

In order to successfully employ ontologies in existing/new applications, structural and semantic interoperability issues among the ontologies that are used for the overall domain must be addressed. There are a number of key issues. First, the individual ontologies of each constituent system used by a new application may each organize knowledge in different ways to suit their specific application and organizational processes, meaning that the ontologies across the constituent systems are often incompatible and difficult to integrate. Second, the ontology development and deployment process is predominantly instance and construction based, often dictated by the talent and expertise of the ontology designer rather than using any concrete software development process; such an approach limits the reuse since ontologies end up being very domain centric. For a new application, the existence of consistent ontologies of the constituent systems will greatly simplify the semantic interoperability. Finally, many existing ontology representational frameworks lack an ability to design solutions that are broader in scope; the end result is often narrowed to not just a single domain, but to a subset of the domain that is very application specific. Thus, the overriding issue is that ontologies solely focus on the domain knowledge and its usage by constituent systems rather than abstracting back from the problem to consider the entire domain and its appropriate set of ontologies in a more comprehensive and general manner. Clearly, there is a lack of design and process in the current ontology definition process – focusing solely on the domain knowledge and its usage by a particular application rather than abstracting back from the problem to consider domain knowledge in a more comprehensive and general manner (Kuhn, 2010). Such an approach is contrary to the long history of design in software, databases, and web settings, where the emphasis is on individual modeling techniques that can applied to conceptualize problem solutions in a fashion that promotes abstraction and fosters reuse. In computing, there is a wide
34 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:
www.igi-global.com/article/a-software-modeling-approach-to-ontology-design-via-extensions-to-odm-and-owl/94599?camid=4v1

www.igi-global.com/e-resources/library-recommendation/?id=2

Related Content

Improving Wiki Article Quality Through Crowd Coordination: A Resource Allocation Approach
www.igi-global.com/article/improving-wiki-article-quality-through-crowd-coordination/97655?camid=4v1a

Semantic Verification of Business Process Models: Prospects and Limitations
www.igi-global.com/chapter/semantic-verification-business-process-models/60060?camid=4v1a
Family History Information Exchange Services Using HL7 Clinical Genomics Standard Specifications
www.igi-global.com/article/family-history-information-exchange-services/2814?camid=4v1a

Advancing Learning Through Virtual Worlds
www.igi-global.com/chapter/advancing-learning-through-virtual-worlds/39191?camid=4v1a