Knowledge Conceptualization and Software Agent based Approach for OWL Modeling Issues

Shuxin Zhao, Pornpit Wongthongtham, Elizabeth Chang, Tharam Dillon

Digital Ecosystems & Business Intelligence Institute, Curtin University of Technology
GPO Box U1987
Perth WA 6845, Australia
{s.zhao, p.wongthongtham, e.chang, tharam.dillon@curtin.edu.au}

Abstract. In this paper, we present the issues and solutions of using OWL ontology to model the knowledge captured in relational databases. Two specific types of knowledge, which are common to various domains, are identified that cannot be represented directly using constructs specified in OWL DL. Firstly the data value range constraint and secondly the calculation knowledge representation. The solution to the first problem is to conceptualize the data range as a new class and the solution to the second problem is proposed, based on utilizing software agent technology. Examples with OWL code and implementation code are given to demonstrate the problems and solutions.

Keywords: OWL modeling issue, Knowledge conceptualization, Software-agent, Ontology

1 Introduction

Since its advent, database technologies have been broadly applied to the development of information systems where persistent data repository and efficient data retrieval are required. This has consequently produced a vast number of databases. These databases embed extensive domain knowledge and up-to-date business information that are crucial to various domains and business processes. With the increasing trend of collaborations amongst organizations and business needs for sharing and publishing their products information, these databases are demanded to be shared and integrated without organizational and application boundaries. However, databases are enterprise and application dependant in that their design and development are subjected to a particular business problem domain of an organization. This has prevented the databases from being shared and integrated in an open environment.

Ontology-based technologies provide a feasible approach to this problem. Ontology-based technologies promote knowledge sharing and integration by formally and explicitly defining the meanings and associations of information and data. Knowledge represented by ontologies will enable machines or software agents to combine and share them from heterogeneous sources meaningfully. An ontology is defined as "a formal, explicit specification of shared conceptualization" [2-4]. Ontologies allow specially designed software agents to automatically process and integrate information from distributed sources. Many approaches have been proposed to transform the knowledge embedded in databases, particularly in relational databases, into ontologies [5-10]. The transformation process involves database reverse engineering to acquire the implicit knowledge from databases and involves mapping the acquired knowledge onto an ontology language. Ontology Web Language (OWL) [11], as the WWW consortium recommendation for the Semantic Web, has gained the popularity as the target ontology language.

However, there is a critical issue of using OWL to fully and accurately represent the knowledge captured in relational databases. Although there are many similarities between a conceptual data model of a database, such as UML or EER model, and an ontology (some researchers classify UML as lightweight ontologies [12]), there are many practical issues when mapping the knowledge captured in a conceptual model onto an OWL ontology. For example, there are three common types of relationships between concepts we model in an UML model, namely, generalization/specialization, aggregation and composition and association. While generalization/specialization can be modeled straightforward using OWL hierarchical mechanism i.e. *Class* and *Subclass, Property* and *Subproperty,* the *aggregation/composition* relationship cannot be represented directly using OWL elements. There are also other types of knowledge captured by a relational database that we found hard to model in OWL such as the value range restrictions on an attribute, and the functional dependency among several attributes of one or more tables which capture some sort of relationships between attributes rather than concepts.

In this paper, we present two alternative solutions to tackle this OWL modeling issue, namely, conceptualization approach and software agent based approach. Two specific examples are used to demonstrate each of the approaches respectively: firstly the problems of modeling the data value range constraint; secondly, the problem of modeling mathematic calculation knowledge, whose operands are derived from attributes of one or more concepts, which represents relationships between these attributes. Our motivation is to reveal some ideas of extending the expressiveness of OWL in the mean time to retain computational completeness of the ontology model, thus to make OWL more useful and more adaptive. The rest of this paper is organized as follows: Section 2 reviews related work on these issues; Section 3 introduces some preliminary concepts necessary for understanding the foundations of the proposed solutions; then Section 4 describes the problems in details with examples; followed by Section 5 demonstrating the solutions to the problems with code example; last in Section 6, we conclude the paper and indicate future work.

2 Related Work

W3c rules is an addition for modeling knowledge in addition to OWL. Expressiveness vs. decidability. HP report OWL weakness.

There is not much work that has been reported on addressing the issues of the knowledge representation with OWL. Stojanovic et al. [6] mentioned that the basic data type system in a database cannot be preserved in F-logic [13] or RDF [14]. Introducing a new class in RDF for each of the types still cannot retain the operators on the basic data types. Furthermore, some database related dynamic knowledge embedded in SQL stored procedures, triggers and built-in functions cannot be mapped to RDF.

Other research considered relevant to mapping databases to ontologies is those that introduce mapping languages such as R2O [15] and D2R MAP [16]. R2O specifies how to populate ontology instances of an existing ontology automatically from the data stored in a relational database. One assumption, on which the proposed approach is based, is that the mapping between an ontology's elements and their correspondent database elements is somehow known already. Under this assumption, R2O aims to be expressive and fully declarative to specify how the ontology instances of the existing ontology can be created from its correspondent database elements such as columns of a table. It, however, does not specify how the data model of a database can be represented by an ontology in RDF or OWL. The other mapping language D2R MAP [16] specifies how to transform the data stored in a relational database into RDF syntax. It requires domain experts and database experts to identify relationships amongst tables via SQL queries in *"D2R sql"* element.

Both of the approaches do not intend to analysis the semantic mappings between a relational data model and the targeting ontology, nor to identify implicit knowledge from databases. Rather, they aim to provide an agile means of wrapping the data held in existing relational databases using RDF or OWL ontology language.

3 Preliminary Concept

3.1 OWL Specification (reduce to an simple description)

OWL[17] is the WWW consortium recommendation for the Semantic Web language. It is designed based on the formal foundation of Description Logics [18]. OWL not only allows formally describing of the meaning of terminology used in web documents but also permits machine inference and reasoning upon literally presented facts. OWL is designed based on RDF [14] and extends RDF. In order to pursue the trade-off between expressiveness and efficient reasoning, OWL has three increasingly expressive sublanguages designed to serve specific levels of implementation and users' needs. They are, namely, *OWL Lite*, *OWL DL*, *OWL Full.* Each of the sublanguages is an extension of its simpler predecessor as stated in OWL specifications [17]. *OWL Lite* provides constructs only for specifying primary needs including classification hierarchy and simple constraints. *OWL DL* supports maximum expressiveness while retaining computational completeness and decidability which means all conclusions are guaranteed to be computable and all computations can be finished in finite time. *OWL Full* [17] supports maximum expressiveness but not computational guarantees. In this paper, we refer the knowledge representation issues with OWL to OWL DL as it is the more practical one to be used in Semantic Web applications. The term *"construct" and "element"* of OWL DL are used interchangeably to describe the building blocks specified in OWL specifications.