Mapping ER Schemas to OWL Ontologies

Igor Myroshnichenko, M.S.                  Marguerite C. Murphy, Ph.D.  
Synarc, Inc.                            San Francisco State University  
igor.myroshnichenko@synarc.com                  mmurphy@sfsu.edu

Abstract

As the Semantic Web initiative gains momentum, a fundamental problem of integrating existing data-intensive WWW applications into the Semantic Web emerges. In order for today’s relational database supported Web applications to transparently participate in the Semantic Web, their associated database schemas need to be converted into semantically equivalent ontologies. In this paper we present a solution to an important special case of the automatic mapping problem with wide applicability: mapping well-formed Entity-Relationship (ER) schemas to semantically equivalent OWL Lite ontologies. We present a set of mapping rules that fully capture the ER schema semantics, along with an overview of an implementation of the complete mapping algorithm integrated into the current SFSU ER Design Tools software.

Keywords: ER, OWL, RDF, Semantic Web, ontology, data model, schema mapping, data-intensive applications

1. Introduction

1.1. Background

The Semantic Web is a network of data on the Web that is currently gaining in popularity and utility [1]. Among many benefits, the Semantic Web will be able to give more precisely targeted answers to search queries. For example, finding a nearby restaurant that serves a particular dish and is open for at least another hour will be trivial with the help of the Semantic Web [2].

The primary technology supporting the Semantic Web is the Resource Description Framework or RDF [3]. RDF uses XML for data (and metadata) representation. RDF adds semantics to data via sets of triples, which can be thought of as the subject, verb and object in a sentence. For example, a triple can assert that person A (subject) is a brother of (verb) person B (object). Subjects, verbs and objects are identified by URIs (Universal Resource Identifiers). Using this framework, anyone on the Web can define a new concept by defining its URI. As the number of these definitions grows and becomes more highly interlinked, ontologies become necessary to provide a higher level of semantic organization. An ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. Ontologies are defined using OWL, a web ontology language that is built on top of RDF. An OWL ontology documents a common understanding of a domain that can be shared between software agents. Semantic Web applications require that ontologies exist for all data over which they operate.

1.2. Problem statement and contributions

In this paper we present a solution to the problem of automatically mapping well-formed ER schemas into semantically equivalent OWL Lite ontologies.

Table 1. Well-formed ER Model subset

<table>
<thead>
<tr>
<th>Component</th>
<th>Subcomponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>Simple Attributes</td>
</tr>
<tr>
<td></td>
<td>Composite Attributes</td>
</tr>
<tr>
<td></td>
<td>Multi-valued Attributes</td>
</tr>
<tr>
<td></td>
<td>Primary Keys</td>
</tr>
<tr>
<td>Weak Entity</td>
<td>Partial Keys</td>
</tr>
<tr>
<td></td>
<td>Attributes</td>
</tr>
<tr>
<td></td>
<td>Owner Entity</td>
</tr>
<tr>
<td>Relationship</td>
<td>Participating Entities (2, 3+)</td>
</tr>
<tr>
<td></td>
<td>Attributes</td>
</tr>
<tr>
<td></td>
<td>Role Names (recursive only)</td>
</tr>
<tr>
<td></td>
<td>Structural Min/Max Constraints</td>
</tr>
</tbody>
</table>

Well-formed ER schemas are based on Chen’s original definition of the ER Model [4] and have simple relational (first-order logic) semantics, as described in most introductory level Database Systems.
textbooks (e.g., [5]). Well-formed ER schemas consist of Entities, Weak Entities, Relationships and Attributes and are complete: all entities have primary key attributes, all weak entities have partial keys, and all relationships have full structural constraints (i.e., an explicit min constraint and an explicit max constraint for each participating entity). Furthermore, there are no additional functional dependencies among the non-key attributes. The well-formed subset of the ER Model is summarized in Table 1.

OWL Lite is described by the W3C as a subset of OWL DL and OWL Full with lower formal complexity [6]. This subset of OWL is computationally guaranteed and was designed to describe classification hierarchies, with ease of developing supporting tools a primary concern. In section 2.2 we describe the subset of OWL Lite used in our work.

A well-formed ER schema is semantically equivalent to an OWL Lite ontology when every set of ER data that is consistent with the well-formed ER schema is also consistent with the semantically equivalent OWL Lite ontology.

The following are major contributions of this paper.

1. A novel approach to mapping ER to OWL. As opposed to existing mapping strategies ([7] and [8]), our approach is based on the strictest, most rigorously defined, and widely supported core subsets of the ER and OWL models: well-formed ER and OWL Lite.

2. Five unambiguous and well defined sequential mapping rules that translate all well-formed ER schemas into semantically equivalent OWL Lite ontologies.

3. A prototype implementation of the complete mapping algorithm based on our mapping rules. Our prototype is built as an extension to the SFSU ER Design Tools and is part of the current distribution of this software made publically available for educational purposes [9].

This paper is based on research done by the first author [10] under supervision of the second author while a postgraduate student at San Francisco State University.

1.3. Paper outline

The remainder of this paper is organized as follows. In Section 2 we describe a real world ER to OWL mapping use case, outline the semantic correspondence between components of well-formed ER, and present our mapping rules. Section 3 contains a description of the prototype implementation. Section 4 provides an overview of related work. In Section 5 we conclude with a brief discussion of our approach and possible areas for future work.

2. Mapping ER to OWL

2.1. Real world use case

In order to demonstrate the practical aspects of mapping well-formed ER schemas to OWL Lite ontologies and to informally validate our mapping rules, we translated a real world example of an ER schema (modeling an X-Ray diffraction process) into an OWL Lite ontology. Our X-Ray diffraction schema is based on a subset of the OPM schema used to store and analyze information about protein macromolecules for RCSB Protein Data Bank [11].

Figure 1 depicts the ER schema we used as input for our mapping use case. Primary components of the ER schema are oCrystal, oDiffractStudy, oAsu, oChain, oResidueMolec, oAtom entities with corresponding simple and composite attributes and PerformedOnCrystal, PerformedOnAsu, ContainsChains, ContainsMolecs, ContainsAtoms relationships.

Executing our mapping algorithm with the X-Ray diffraction ER schema as input produced a valid OWL Lite ontology presented in Figure 2 (notation described in [12]). The ontology consists of classes corresponding to the entities of the ER schema and object properties corresponding to ER relationships.
Figure 2. X-Ray diffraction OWL Lite schema

named in accordance with the conventions introduced by our mapping rules. We observe that all of the structural constraints in the input ER schema have been correctly mapped to equivalent OWL constraints.

In a realistic scenario, an ER schema similar to our X-Ray diffraction schema could represent a data source participating in a (heterogeneous) data integration workflow. Ontologies are widely used in industry as a universal data representation format over which data integration and merging operations are implemented [13]. Our mapping approach could prove useful as a component of heterogeneous data source integration by providing a simple, automated way to map schemas of existing ER data sources to equivalent OWL ontologies, thus simplifying and speeding up the overall data integration process.

2.2. Correspondence between components of well-formed ER schemas and OWL Lite ontologies

Table 2 presents a list of corresponding components of the well-formed ER Model and the OWL Lite Web Ontology Language. Full details are presented in [10]. Entities with single-valued and multi-valued attributes correspond to classes with datatype properties. Binary relationships without attributes correspond to a pair of inverse object properties associated with classes equivalent to participating entities. Composite attributes, weak entities, binary relationships with attributes and ternary relationships do not have direct equivalents in the realm of OWL and correspond to a collection of OWL components. A composite attribute and a weak entity correspond to a class with datatype properties associated with its base class via a pair of inverse object properties. A binary relationship with attributes corresponds to a class with datatype properties and two pairs of inverse object properties pointing to classes equivalent to participating entities. A ternary relationship corresponds to a class with three pairs of inverse object properties pointing to classes equivalent to participating entities.

<table>
<thead>
<tr>
<th>ER Component</th>
<th>OWL Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>Class</td>
</tr>
<tr>
<td>Strong Entity</td>
<td>Class</td>
</tr>
<tr>
<td>Weak Entity</td>
<td>Base class and additional class encapsulating</td>
</tr>
<tr>
<td></td>
<td>equivalents of the partial</td>
</tr>
<tr>
<td></td>
<td>key attributes and the key</td>
</tr>
<tr>
<td></td>
<td>attribute of the owner</td>
</tr>
<tr>
<td></td>
<td>entity</td>
</tr>
<tr>
<td>Attribute</td>
<td>Datatype property</td>
</tr>
<tr>
<td>Single-valued</td>
<td>Functional datatype property</td>
</tr>
<tr>
<td>Attribute (null)</td>
<td></td>
</tr>
</tbody>
</table>
### 2.3. Mapping rules

The mapping rules we use for translating well-formed ER schemas to OWL Lite ontologies are based on a strategy originally developed to map ER schemas into Object Data Management Group (ODMG) class definitions [5]. We have used the ER to ODMG rules as an initial guideline and have adapted and extended them to comply with the semantics of the OWL Lite Web Ontology Language. Reference [10] contains a more detailed description of our mapping rules with examples.

#### 2.3.1. First mapping rule.
Map each entity into a class. Map each simple attribute into a functional datatype property. Map each multi-valued attribute into a datatype property (in OWL simple non-functional datatype and object properties are multi-valued by default). Map each composite attribute into a separate class with functional datatype properties corresponding to the composite attribute’s components; add a functional object property with range set to the newly created class. Map each composite key attribute into a functional datatype property with min cardinality set to one. Map each simple key attribute into a functional datatype property with range set to the newly created class with min cardinality set to one.

#### 2.3.2. Second mapping rule.
Map each weak entity into a class with a functional object property whose range is set to the owner class, min cardinality set to one and an object property in the owner class with range set to the weak entity class. The object properties should be inverses of each other. Map the weak entity’s simple, multi-valued and composite attributes according to the first rule. Map partial key attributes into a separate class with corresponding functional datatype properties whose min cardinality is set to one, a functional object property with the range set to the owner entity class and whose min cardinality is set to one, a functional inverse-functional object property with the range set to the weak entity class and whose min cardinality is set to one; add a functional inverse-functional object property whose range is set to the newly created class and whose min cardinality is set to one.

#### 2.3.3. Third mapping rule.
Map each binary relationship without attributes into a pair of object properties in the two classes corresponding to the two participating entities. Map participating entities’ cardinality into min and max cardinality restrictions or combine characteristics of a functional property with a min cardinality restriction.

#### 2.3.4. Fourth mapping rule.
Map each binary relationship with attributes into a class with datatype properties corresponding to the relationship attributes and two pairs of inverse object properties between participating entity classes and the relationship class. Map participating entities’ cardinality into min and max cardinality restrictions or combine characteristics of a functional property with a min cardinality restriction for object properties pointing from participating entity classes to the relationship class. For their inverse object properties, min and max cardinality should be set to one.
2.3.5. Fifth mapping rule. Map each ternary relationship into a class with three pairs of inverse object properties between participating entity classes and the relationship class. Map participating entities’ cardinality into min and max cardinality restrictions or combine characteristics of a functional property with a min cardinality restriction for object properties pointing from participating entity classes to the relationship class. For their inverse object properties, min and max cardinality should be set to one.

3. Proof of concept implementation

We implemented our mapping rules in the form of an extension to the SFSU ER Design Tools [9]. This code is fully operational and is available for instructional use.

The SFSU ER Design Tools is a two-tier Java program designed to provide the user with GUI-driven utilities for creating, editing and mapping of schemas of the following data models: ER, Relational, ODMG, SQL, ER/XML, XML DTD, Spreadsheet, NIH and UML. Support for OWL Lite was added as part of this work [10].

![Figure 3. SFSU ER Design Tools ER to OWL mapping scenario workflow](image)

The workflow of the SFSU ER Design Tools to map well-formed ER schema to equivalent OWL Lite ontologies is as follows (see Figure 3):

1. The user interacts with the GUI of the system to enter a well-formed ER schema, internally represented by an ERSchema object graph.
2. The ERDesignToolsClient dehydrates the ERSchema object graph into an XML document following a proprietary ER XML format and sends this document to the ERDesignToolsServer for further processing.
3. The ERDesignToolsServer receives the XML document, hydrates it back into an ERSchema object graph, applies the appropriate mapping algorithm to map an ERSchema instance into an OWLLiteOntology, dehydrates it into an OWL Lite/RDF document and sends it back to the ERDesignToolsClient.
4. The ERDesignToolsClient displays the resulting RDF document to the user.

4. Related work

In this section we provide a general overview of existing solutions rather than a comprehensive survey. This is due to the large scope and amount of related work. It is also worth noting that although OWL is layered on top of RDF, mapping database schemas to RDF [14] is outside of the scope of this project. In comparison to the multi-purpose nature of RDF, OWL is a high level language specifically designed only for describing ontologies.

Currently there are three major approaches to solving the problem of extracting ontologies from database schemas for data-intensive systems: mapping relational schemas to ontologies, mapping relational schemas in combination with additional domain semantics (such as structure of the user interface) to ontologies, and mapping ER schemas to ontologies. In our work we have used the latter approach. Most well designed database applications use fully normalized relational data derived from well-formed ER schemas [5]; all well-formed ER schemas are equivalent to fully normalized relational schemas [4]. By restricting attention to well-formed ER schemas, we avoid the need to consider many difficult, but not particularly useful special cases of the full Relational Model.

The primary goal of the first approach, as described by Stojanovic [15], Hu [18] and Gwani [17], is to enable relational database-driven applications to participate in the Semantic Web by means of (semi-) automatically mapping their relational schemas to equivalent ontologies. Stojanovic [15] and Hu [18] propose a set of rules that map relational schemas to Frame Logic and OWL ontologies respectively. In addition, Stojanovic [15] and Hu [18] specify a data migration process. Gwani [17] describes a framework in which a series of referential integrity constraint-based mapping steps produce a mapping descriptor which specifies relationships between relational schema and ontology components and is used to translate SPARQL queries to SQL.

The second approach, as proposed by Astrova [16], ports relational database-driven web applications to the Semantic Web by analyzing combinations of the underlying relational schema and the structure of corresponding HTML files in the front-end of the application.

The third approach discussed in Upadhyaya [7] and Fahad [8] is the closest to the work presented in this paper. References [7] and [8] focus on reverse engineering relational databases into ontologies by means of defining a set of mapping rules for mapping underlying Extended ER schemas to OWL ontologies.
In contrast to the work presented in these two papers, our mapping approach does not rely on the Extended ER Model (with support for generalization), but is limited to the core, well-formed subsets of the ER Model and OWL. As opposed to our approach, which can automatically map any valid well-formed ER schema to an equivalent OWL DL ontology, both [7] and [8] require some user input to finalize the resulting ontology ([7] produces a near-complete ontology to be finalized by a domain expert, [8] prompts the user for the direction of relationships at the time of mapping). There are also differences between how we map primary keys and what [7], [8] suggest: [7] and [8] map key attributes to inverse-functional datatype properties, which is illegal in OWL Lite and OWL DL. In both OWL Lite and OWL DL, datatype and object properties are disjoint [6] as opposed to OWL Full. OWL full, on the other hand is not computationally guaranteed and cannot be successfully used for implementing automated tools. We map primary keys to functional datatype properties with min constraint set equal to one.

5. Conclusions and future work

With the rapid recent development of the Semantic Web, it is becoming increasingly important to address the problem of porting existing data-intensive web applications to the new OWL infrastructure. Most recently created relational databases are designed using ER schemas and ER to relational schema mapping, with the ER Model serving as the semantic model during design [5]. Directly translating the initial ER schema into an equivalent OWL schema is not only more straightforward than attempting to translate the resulting relational schema, but serves to more easily preserve all of the associated semantic constraints in the original ER schema (some of which are frequently lost due to incomplete ER to Relational mapping strategies, e.g. [5]).

The mapping rules developed during this project serve as a foundation for a complete mapping algorithm that we implemented as an extension to the SFSU ER Design Tools. Implementation and unit tests of the module demonstrate that it is fully functional, produces correct results and can be effectively applied in practical use cases.

The next steps in our work will be to explore practical data migration paths, query support based on our ER to OWL mapping rules, and ways to allow reuse of existing vocabularies and ontologies. With the help of a system that can not only map ER schemas to OWL ontologies but also migrate data and translate SQL queries to SPARQL, legacy relational database-centric applications will be closer to seamless integration into the Semantic Web.

6. References