Chapter XVI
Lightweight Data Modeling in RDF

Axel Rauschmayer
University of Munich, Germany

Malte Kiesel
DFKI, Germany

INTRODUCTION

When looking at what "information" means in the context of the Semantic Web, there is an interesting dichotomy (Spyns, Meersman, & Jarrar, 2002; Motik, Horrocks, & Sattler, 2007): On one hand, there is *ontology engineering* where knowledge is being universally represented and reasoned about. On the other hand, there is *data modeling* where Semantic Web technology is used to store and process information that is mainly application-dependent. The former receives most of the attention of the research community, but one must not forget that, even in semantic work environments, one frequently encounters the latter: citations, contacts, product information, events, and so forth. In this chapter, we present the *editing meta-model* (EMM), which provides standards and techniques for implementing RDF editing: It defines an RDF vocabulary for editing and clearly specifies the semantics of this vocabulary. It also sketches user interface mechanisms to illustrate how the vocabulary would be used in practice. RDF has many existing standards that are more or less related to editing and presentation:

1. **Schema:** The *RDF Schema Language* RDFS (Brickley & Guha, 2004) and the *Web Ontology Language* OWL (Bechhofer et al., 2004) are schema[1] languages for describing the structure of data.
2. **Presentation:** The *Fresnel Display Vocabulary* (Bizer, Pietriga, Karger, & Lee, 2006) helps with presenting the data. It declaratively specifies how RDF data should be formatted and laid out. This means that you can package both the data and instructions on how to display it in the same RDF graph.

3. **Reversible embedding for publication:** *RDFa* (Adida & Birbeck, 2006) extends XHTML so that RDF data can be embedded inside it. This means that the process of merging semi-structured data (XHTML) and structured data (RDF) for publication becomes reversible; tools become feasible that, when given a Web address, can extract the data embedded in it, in a clearly defined, unambiguous way.
4. **Querying:** The *SPARQL query language* (Prud'hommeaux & Seaborne, 2005) aids in flexible data retrieval and is used in advanced Fresnel applications.
5. **Computed data:** Not all information has to be explicitly stated, some of it can be derived from existing data. OWL inferencing is one mechanism for such derivation. More powerful, rule-based approaches are currently being standardized ("*The Rule Markup Initiative*," 2007) for RDF.

EMM focuses on (1), (2), and (3). It is a reflection on how they can be used and extended to better support data modeling. Both OWL and Fresnel are lacking for this task: OWL is too complex for basic data editing, as stated by Hendler who advocates a simplified version of OWL called "OWL Mini" (Hendler, 2006), which is currently work in progress. By its very nature, OWL does not support editing-specific views on RDF data: For a class of resources, such a view would specify what properties to edit (while ignoring the rest), how to display them and in what order. Fresnel has been created to complement OWL in this regard and calls editing-specific views *lenses*. But Fresnel is only concerned with displaying data and lacks several crucial features for editing. The EMM is a combination of subsets and extensions of existing standards and is split into three parts:

- **Schema:** We specify a simple type system that is loosely based on OWL and provides everything that is needed for most data editing applications. We also introduce some constructs that OWL is missing.
- **Presentation:** We restrict and extend Fresnel to suit our purposes. Some parts of Fresnel are too advanced for the current iteration of the EMM, some things are naturally missing, as Fresnel was never intended to support editing.
- **Editing:** Data structures and algorithms that are necessary during the actual editing of data. We never edit the data directly, but change it in a three step process: First a lens is used to create a *projection* of the data to be edited. This projection is a tree-structured view on the data that reflects the definitions of the lens. Second, the user changes the data, rearranging the nodes of the projection as the user pleases. Third, the projection is *applied* to the data: The changes encoded inside the projection are committed to the data.

In order to better illustrate how the EMM could be implemented, we also show what a graphical user interface for the EMM looks like. For this chapter, we assume familiarity with RDF. Knowledge about OWL and Fresnel is helpful, but we try to explain most of the basic ideas.

OVERVIEW OF THE EDITING META-MODEL (EMM)

In this section, we explain some of the conventions we adhere to and give a brief overview of the EMM.

Conventions Used in this Chapter

The following markers are use to indicate how we extend or restrict existing standards:

(Ignored) A feature of a standard is ignored, because it is not (nor will it ever be) useful for editing.