3

The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS

Brian McBride

Hewlett Packard Laboratories, Bristol, UK.

Summary. An informal introduction to the W3C's updated Resource Description Framework (RDF) and its vocabulary description language is given. RDF's role in the semantic web and its relationship to other semantic web languages is described. The basic concepts of RDF and RDF Schema are explained and an example RDF schema is given. Limitations of RDF are described.

3.1 Introduction

The Resource Description Framework (RDF) [1,2] is a W3C recommendation that defines a language for describing resources. It was designed for describing Web resources such as Web pages. However, RDF does not require that resources be retrievable on the Web. RDF resources may be physical objects, abstract concepts, in fact anything that has identity. Thus, RDF defines a language for describing just about anything.

RDF describes resources in terms of named properties and their values. The RDF Vocabulary Description language, RDF Schema (RDFS) [3] describes vocabularies used in RDF descriptions. RDF vocabularies describe properties, classes of resources and relationships between them.

RDF is the foundation of the Semantic Web [4]. Just as the Web is a global infrastructure representing information in documents, the Semantic Web is a global infrastructure representing information in a form that can be processed by computer. Like the Web, the semantic Web is decentralized, which imposes a severe constraint on the mechanism it uses to represent information.

Traditional frame based and object oriented systems are resource-centric. They define classes and the properties instances of those classes must or may have. This is a centralized approach. Whoever defines a class defines what properties its instances have. To add more properties requires either the cooperation of the owner of the class or the definition of a new subclass.

S. Staab et al. (eds.), *Handbook on Ontologies*

Both RDF and RDFS[1] are Web languages. They are designed to allow information and vocabularies to be developed in a decentralized fashion. Just as the Web permits anyone with access to a server to create a Web page and link it to any other Web page, RDF(S) was designed following the principle that anyone should be able to say anything about anything[2]. To be able to say anything about anything, anyone must be able to define new properties for a class. Further, they should be able to use the same property to describe any class they choose. To support this need, RDF(S) is property centric. It enables properties to be defined and then used to describe resources.

RDF(S) has a formal semantics [5]. A formal semantics is needed for two reasons. Firstly, it brings precision to the specification of RDF. Without a formal semantics, there is too much scope for implementations to differ. Secondly, languages such as Owl [6] that extend RDF(S) have a formal semantics. Such languages require RDF(S) to have a formal semantics that they can extend, otherwise, they must define their own semantics for RDF, creating the possibility that different extension languages define different semantics.

RDF(S) are members of a family of Semantic Web languages. They builds on URI's [7] the Web's language for naming things, and on XML [8], the standard syntax for representing information in the Web. The DAML+OIL ontology language and its standardized form, the Owl [6] family of ontology languages, are extensions of RDF(S). Research on languages for querying the semantic Web [9, 10] and for expressing rules is likely to inform future efforts to standardize such languages.

The remainder of this chapter is organized as follows. Section 2 describes RDF, its abstract syntax, the use of blank nodes, support for datatypes and how it may written as XML. Section 3 describes RDFS, the notions of class and subclass, property and subproperty, domain and range constraints and the built in vocabularies for containers, collections and reification. Section 4 summarizes the key features of RDF(S), describes some of its limitations and refers to later chapters that describe more powerful languages that overcome these limitations.

The reader should note that at the time of writing (March 2003), the W3C's RDFCore Working Group are revising the original specification of RDF [11] and completing work on the RDF Schema specification [3]. Whilst this work is thought to be nearly complete, late changes made by the working group may not be represented in here.

---

[1] For the remainder of this chapter, the term RDF(S) will be used instead of "both RDF and RDFS".

[2] There are things that RDF is too weak to express; for example, it lacks negation and universal quantification. It is not possible to say everything about everything using RDF.