Extension of RDF(S) with Contextual and Definitional Knowledge

Alexandre Delteil, Catherine Faron-Zucker, Rose Dieng
INRIA, ACACIA Project, 2004 route des Lucioles, BP 93,06902 Sophia Antipolis, France
{Alexandre.Delteil,Catherine.Faron,Rose.Dieng}@sophia.inria.fr

1 Introduction

The need of a Semantic Web is now well recognized and always more emphasized. The huge amount of information available on the Web has become overwhelming, and knowledge based reasoning now is the key to lead the Web to its full potential. In the last few years, a new generation of knowledge based search engines has arisen which rely on extensions of HTML to annotate Web documents with semantic metadata, thus enabling semantic content guided search. For interoperability on the Web, the importance of widely accepted standards is emphasized. RDF is the emerging standard proposed by the W3C for the representation and exchange of metadata on the Semantic Web. RDF Schema is the standard dedicated to the representation of ontological knowledge used in RDF statements.

ACACIA is involved in the CoMMA European IST project, dedicated to corporate memory management through agents. The memory is materialized by the electronic documents of the organization which are described by RDF annotations. These are the key for knowledge based information retrieval on the Intranet by using the inference engine *CORESE* implemented in our team (Corby *et al.*, 2000). However the expressivity of RDF(S) appears too much limited to represent the ontological knowledge of the corporate memory. Inference rules representing domain axioms, class and property definitions are crucial for intelligent information retrieval on the Web. The need for axiomatic knowledge is well-known since the first information retrieval systems on the Semantic Web. It is the key to discover implicit knowledge in Web page annotations so that information retrieval be independent of the point of view adopted when annotating.

When compared to object-oriented knowledge representation languages, description logics, or conceptual graphs, RDF(S) does not enable to declare class, property and axiom definitions. We have specified an extension of RDF(S) with class, property and axiom definitions based on the similarity between the RDF and the Conceptual Graphs models. We call it DRDF(S) for *Defined Resource Description Framework* (Delteil *et al.*, 2001a). Other extensions of RDFS have been proposed, such as DAML and OIL. DAML provides useful primitives for declaring intersection, disjunction, complementary of classes, … . OIL stems from a description logic; as DRDFS, it provides a way of expressing class and property definitions. However DRDFS and OIL have two incomparable expressivities, in the sense that none can be considered as a fragment of the other.

What DRDFS provides also is the ability to express contextual knowledge on the Web (Delteil *et al.*, 2001b). The RDF philosophy consists in letting anybody free to declare anything about any resource. Therefore the knowledge of who and in which context a certain annotation has been stated is crucial. DRDF(S) enables to assign a context to any set of annotations. We hope that DRDF(S) will contribute to the ongoing work of the W3C committee for improving RDFS.

2 RDF(S) and its Limitations

RDF is a data model provided with an XML syntax. RDF knowledge is positive, conjunctive and existential. A set of statements is viewed as a directed labeled graph: a vertex is either a resource or a literal; an arc between two vertices is labeled by a binary property. RDFS is dedicated to the specification of schemas representing the ontological knowledge used in RDF statements. A schema consists in declarations of classes and properties. It is defined by refining the core RDFS: domain specific classes and properties are declared as instances of the *Class* and *Property* resources; the *subClassOf* and *subPropertyOf* relations enable the representation of class and property hierarchies.

2.1 A Triple Model.

The RDF data model is a triple model: an RDF statement is a triple (resource, property, value). When asserted, RDF triples are clustered inside annotations. An annotation can thus be viewed as a graph, subgraph of the great RDF graph representing the whole set of annotations on the Web. However, there is no distinction between the statements made in a single sentence and the statements made in separate sentences. Let us consider two different annotations relative to two different research projects which the employee 46 of T-Nova participates to:

- {(employee-46, worksIn, T-Nova), (employee-46, project, CoMMA), (employee-46, activity, endUser)} and
- {(employee-46, worksIn, T-Nova), (employee-46, project, projectXX), (employee-46, activity, developer)}.

The whole RDF graph does not distinguish between these two clusters of statements. Employee 46 is both endUser and developer: the knowledge of which activity he is implicated in inside of a project is lost.

DRDF(S) enable to represent independent clusters of RDF statements through the *context* feature.

2.2 RDF Reification.

The RDF model is provided with a reification mechanism dedicated to higher order statements about other statements. A statement (r, p, v) is reified into a resource s described by the four following properties: the *subject* property identifies the resource r, the *predicate* property identifies the original property p, the *object* property identifies the property value v, the *type* property describes the type of s; s is an instance of *rdf:Statement*.
However, the reification of a set of statements requires the use of a container to refer to the collection of the resources reifying these statements. This leads to quite complicate graphs. Moreover a statement containing an anonymous resource can not always be reified: the values of the properties *subject* and *object* must have an identifier. The notion of context we introduce in DRDF(S) enable to reify a set of statements much more easily.

2.3 Existential quantification through anonymous resources.

The RDF model focuses on the description of identified resources but allows a limited form of existential quantification through the anonymous resource feature. It is handled by RDF parsers by automatically generating an ID for the anonymous resource. However, it is a limited solution and a graph containing a cycle with more than one anonymous resource can not be represented in the XML syntax of RDF. DRDFS enable to represent every existential, positive and conjunctive statement, without any restriction.

The roots of DRDF(S) stand in the correspondence between RDF(S) and the conceptual graph (CG) model (Sowa, 2001). The CG model provides a direct way of expressing independent pieces of knowledge through graphs. It thus enables the representation of contexts for various applications (quotations, viewpoint, …). CGs are particularly useful as definitional contexts enabling the definition of concepts or axioms (Delteil *et al.*, 2001a). An in-depth comparison of both models is studied in (Corby *et al.*, 2000).

3 Extending RDFS with Contexts, Existential Quantification and Coreference

In DRDF(S), a resource of type *Context* expresses the clustering of statements - much more easily than an RDF container. A context identifies a sub-graph of the whole RDF graph. A context is defined from a resource G of type *Context* as the largest subgraph of the whole RDF graph whose all internal nodes excepted G are anonymous resources $c\varnothing_i$. A context is an abstraction that enables to talk about representations of resources (through anonymous resources) rather than directly about resources. Anonymous resources are “externally identified” by the *referent* property.
DRDF(S) is provided with a general mechanism for existential quantification handling. It is represented by an anonymous resource described by a *referent* property whose value is an instance of *Variable*, a new RDF class we introduce. The scope of a variable is the context it belongs to, just like in FOL, where the scope of a variable is the formula it belongs to. We introduce a *parameter* property to link a resource of type *Variable* to a resource of type *Context*.

4 Special Contexts: Axioms, Class and Property Definitions

This general feature of context can be used for representing axioms, and class and property definitions. DRDF(S) class and property definitions are descended from type definitions in the CG model; DRDF(S) axioms are descended from Conceptual Graph rules. A class definition is a monadic abstraction, i.e. a context whose one resource of type Variable is considered as formal parameter; a property definition is a diadic abstraction. An axiom is a couple of lambda abstractions, i.e. two contexts representing the hypothesis and the conclusion.

5 References

Corby, O., Dieng, R., Hebert, C. (2000). A Conceptual Graph Model for W3C Resource Description Framework. ICCS’00, LNCS 1867, 2000.
Delteil, A., Faron, C., Dieng, R. (2001a). Extensions of RDFS Based on the Conceptual Graph Model. ICCS.
Delteil, A., Faron, C., Dieng, R. (2001b). Extensions of RDFS with Contextual and Definitional Knowledge. WebNet.
Sowa, J.F. (2001). Conceptual Graphs: DpANS. http://www.bestweb.net/%7Esowa/cg/cgstand.htm, 2001.