Implementing Simple Modular ERDF ontologies

Carlos Viegas Damásio[1] and **Anastasia Analyti**[2] and **Grigoris Antoniou**[3]

Abstract. The Extended Resource Description Framework has been proposed to equip RDF graphs with weak and strong negation, as well as derivation rules, increasing the expressiveness of ordinary RDF graphs. In parallel, the Modular Web framework enables collaborative and controlled reasoning in the Semantic Web. In this paper we show how to use the Modular Web Framework to capture the modular semantics for ERDF graphs, supporting local semantics and different points of view, local closed-world and open-world assumptions, and scoped negation-as-failure.

1 Introduction

The Extended Resource Description framework [1] (ERDF) provides a model theoretical semantics for RDF graphs allowing negative triples, and ontologies defined by first-order rules including the two forms of negation, weak and strong. Subsequently, the ERDF framework has been extended to allow the specification of import and export declarations of classes and properties, resulting in the Modular ERDF framework [3]. The Modular Web Framework (MWeb) is a proposal to address the issues of programming-in-the-wide faced by the new Semantic Web rule-engines [4]. MWeb defines general constructs to allow sharing of knowledge in the Semantic Web provided by logic based knowledge bases, including scoped open and closed world assumptions as well as contextualized and global interpretation of predicates. It provides separate interface and implementation of rulebases with modular and independent compilation and loading. A compiler of MWeb into XSB Prolog is available making use of the tabling features to guarantee termination of recursive rules with negation. This paper defines the embedding of an important fragment of ERDF ontologies into the MWeb system, allowing for the integration of both systems, and resulting in a working implementation available at `http://centria.di.fct.unl.pt/~cd/mweb`.

2 Simple Modular ERDF ontologies

In this section, we recap the notion of simple modular ERDF ontology [3] which allows the combination of knowledge in different ontologies. We assume throughout a Web vocabulary V is given, formed by a set of URI references and/or literals (plain or typed). Variable symbols are prefixed by "?" belonging to a set Var disjoint from V.

The notion of RDF triple is extended by allowing the use of strong negation (negative triples), and by permitting literals in the subject.

Definition 1 (ERDF triple and ERDF graph [1]) *A* ERDF triple *over vocabulary* V *is an expression of the form* $s[p\ \texttt{->>}\ o]$ *or* $\texttt{neg}\ s[p\ \texttt{->>}\ o]$*, where* $s, o \in V \cup \mathit{Var}$ *are called* subject *and* object*, respectively, and* $p \in V \cap$ URI *is called* property*. An* ERDF graph G *over* V *is a set of ERDF triples over* V*. The* skolemization *of* G*, denoted by* $sk(G)$*, is the ground ERDF graph derived from* G *after replacing each variable by a new fresh constant.*

Definition 2 (r-ERDF literal and simple r-formula) *Let* $O_{\texttt{nam}} \subseteq$ URI *and* T *be an arbitrary* ERDF triple*. An* r*-ERDF literal is a positive or negative* ERDF triple T*, its weak negation* $\texttt{naf}\ T$*, or a qualified literal* $T@\mathit{oname}$ *or* $\texttt{naf}\ T@\mathit{oname}$ *where* $\mathit{oname} \in O_{\texttt{nam}}$*. A simple* r*-ERDF formula is a conjunction of* r*-ERDF literals.*

The set $O_{\texttt{nam}}$ contains the URIs for identifying r-ERDF ontologies. Weak negation $\texttt{naf}\ T$ is non-monotonic and can be used to check if T is believed false. Qualified literals of the form $L@\mathit{oname}$ are used to evaluate L at the r-ERDF ontology identified by oname.

Definition 3 (Simple r-ERDF rule, r-ERDF program) *A simple* r-ERDF rule r *is an expression of the form:* $G\texttt{:-}F$*, where* F *is a simple* r*-ERDF formula or* `true` *and* G *is either an ERDF triple over* V *or* `false`*. A simple* r-ERDF program P *is a finite set of simple* r*-ERDF rules over* V *and* $O_{\texttt{nam}}$*.*

A simple r-ERDF ontology specifies a unit of knowledge, which is described by an interface, and formed by an ERDF graph and a simple r-ERDF program.

Definition 4 (Simple r-ERDF ontology) *A simple* r-ERDF ontology O *is a triple* $O = \langle \mathit{Nam}_O, L_O, \mathit{Int}_O \rangle$*, where: (i)* $\mathit{Nam}_O \in O_{\texttt{nam}}$ *is the* name *of* O*, (ii)* $L_O = \langle G_O, P_O \rangle$ *is the* logic *of* O*, where* G_O *is an ERDF graph over* V *and* P_O *is a simple* r*-ERDF program over* V *and* $O_{\texttt{nam}}$*, and (iii)* $\mathit{Int}_O = \langle \mathit{Exp}^{\texttt{pr}}_O, \mathit{Exp}^{\texttt{cl}}_O, \mathit{Imp}^{\texttt{pr}}_O, \mathit{Imp}^{\texttt{cl}}_O \rangle$ *is the* interface *of* O*. For* $\texttt{t} \in \{\texttt{pr}, \texttt{cl}\}$*, let* $\mathit{Exp}^{\texttt{t}}_O$ *be a set of pairs* $\langle x, \mathit{Exp} \rangle$ *where* $x \in V$ *and* $\mathit{Exp} \subseteq O_{\texttt{nam}} - \{\mathit{Nam}_O\}$ *or* $\mathit{Exp} = \{*\}$*, and* $\mathit{Imp}^{\texttt{t}}_O$ *is a set of pairs* $\langle x, \mathit{Imp} \rangle$ *where* $x \in V$*, and* $\mathit{Imp} \subseteq O_{\texttt{nam}} - \{\mathit{Nam}_O\}$ *or* $\mathit{Imp} = \{*\}$*. In each of the sets in the interface* Int_O *there are no duplicate pairs for the same* x*.*

Each pair $\langle x, \mathit{Exp} \rangle \in \mathit{Exp}^{\texttt{pr}}_O$ (resp. $\langle x, \mathit{Exp} \rangle \in \mathit{Exp}^{\texttt{cl}}_O$) corresponds to an `export` declaration of O, where x is a property (resp. class) exported by O and Exp is the list of simple r-ERDF ontologies to which O is willing to export x. If O is willing to export x to any requesting r-ERDF ontology then $\mathit{Exp} = \{*\}$. The pairs in $\mathit{Imp}^{\texttt{t}}_O$ are `import` declarations, and have the corresponding interpretation.

Definition 5 (Simple modular ERDF ontology) A *simple modular ERDF ontology* (SMEO) $\mathcal{R}$ is a set of simple r-ERDF ontologies.

---

[1] CENTRIA, Departamento de Informática da Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal. email: cd@di.fct.unl.pt

[2] Institute of Computer Science, FORTH-ICS, Crete, Greece. email: analyti@ics.forth.gr

[3] Institute of Computer Science, FORTH-ICS, and Department of Computer Science, University of Crete, Crete, Greece. email: antoniou@ics.forth.gr

From the interfaces of the simple `r`-ERDF ontologies in a particular SMEO $\mathcal{R}$, it is defined $Export^{\mathtt{t}}_{O,\mathcal{R}}(x)$ with $\mathtt{t} \in \{\mathtt{pr}, \mathtt{cl}\}$, denoting the `r`-ERDF ontologies in $\mathcal{R}$ to which O is willing to export property or class x, and $Import^{\mathtt{t}}_{O,\mathcal{R}}(x)$ denoting the `r`-ERDF ontologies in $\mathcal{R}$ from which O imports property or class x. Additionally, the dependencies of an `r`-ERDF ontology O with respect to $\mathcal{R}$, and denoted by $D^{\mathcal{R}}_{O}$, are the `r`-ERDF ontologies in $\mathcal{R}$ on which depends. Briefly, they are obtained by the transitive closure of the imported ontologies relation plus all the ontologies used in qualified literals, starting from O. For the formal definition of these notions see [3].

The semantics of modular ERDF ontologies has been specified model-theoretically in [3], and addresses the issues of compatibility with the RDF semantics, extending $\#n$-stable-model semantics on ERDF ontologies [2] and RDFS semantics on RDF graphs. An important issue is that the vocabulary of RDF must be restricted by limiting the maximum number of container membership properties, otherwise reasoning would become undecidable.

3 The MWeb embedding of ERDF ontologies

Instead of the model theoretical approach, we follow an alternative path by providing a syntactical embedding into the Modular Web Framework (MWeb) and use MWeb semantics [4] to obtain a mapping into extended logic programming. The MWeb framework requires for each ontology the definition of an interface document and of the corresponding rulebase (logic) document.

Definition 6 *Let $O \in \mathcal{R}$ be a simple* `r`-ERDF ontology. *The corresponding MWeb interface document $MWeb^{Int}(O)$ is constructed as follows:*

`:- rulebase '`Nam_O`'.`
`:- import('erdf.mw',interface).`
% If there are RDF container membership properties in L_O, let n be
% the maximum i of `rdf:'_i'` *properties in L_O, then add*
`:- vocabulary rdf:'_1',`$\ldots$`, rdf:'_n'.`
`:- defines global normal class(mw:Vocabulary).`
% If $\exists\ \langle c, Exp\rangle \in Exp^{\mathtt{cl}}_{O}$, let $\{r_1,\ldots,r_n\} = Export^{\mathtt{cl}}_{O,\mathcal{R}}(c)$ and add
`:- defines global normal class(`c`)`
`   visible to '`r_1`',`$\ldots$`,'`r_n`'.`
% If $\exists\ \langle p, Exp\rangle \in Exp^{\mathtt{pr}}_{O}$, let $\{r_1,\ldots,r_n\} = Export^{\mathtt{pr}}_{O,\mathcal{R}}(p)$ and add
`:- defines global normal property(`p`)`
`   visible to '`r_1`',`$\ldots$`,'`r_n`'.`
% If $\exists\ \langle c, Imp\rangle \in Imp^{\mathtt{cl}}_{O}$, let $\{r_1,\ldots,r_n\} = Import^{\mathtt{cl}}_{O,\mathcal{R}}(c)$ and add
`:- uses normal class(`c`)from '`r_1`',`$\ldots$`,'`r_n`'.`
% If $\exists\ \langle p, Exp\rangle \in Exp^{\mathtt{pr}}_{O}$, let $\{r_1,\ldots,r_n\} = Import^{\mathtt{pr}}_{O,\mathcal{R}}(p)$ and add
`:- uses normal property(`p`)from '`r_1`',`$\ldots$`,'`r_n`'.`
% Let $\{u_1,\ldots,u_m\} = \{Nam_{O'} \mid O' \in D^{\mathcal{R}}_{O}\}$
`:-uses normal class(mw:Vocabulary)from'`u_1`',`$\ldots$`,'`u_m`'.`

The first line identifies the rulebase, while the `import` directive includes in the interface the necessary declarations for supporting ERDF reasoning, namely the vocabularies of RDF, RDFS and ERDF. The `erdf` ontology implements in MWeb itself the underlying semantics of modular ERDF ontologies just using simple `r`-*ERDF* rules, including RIF, RDFS and RDF combination, and is not included for lack of space. The vocabulary of the rulebase is collected in the pre-defined class `mw:Vocabulary` of the MWeb framework and is formed by the local vocabulary and the imported vocabularies of the rulebases in its dependency list (last uses statement). The rulebase vocabulary is used for providing the domain for (scoped) negation as failure, open and closed world assumptions. The remaining declarations specify the exported and imported classes and properties, with appropriate visibility and import lists. The important point is that all classes and properties are defined normal and global. This means that all classes and properties exported can be used and redefined (`global`), and that rules can use weak negation (`normal`) and thus are non-monotonic. The translation of the logic document of an `r`-*ERDF ontology* is immediate.

Definition 7 *Let $O = \langle Nam_O, L_O, Int_O\rangle$ be a simple* `r`-ERDF ontology. *The MWeb logic document $MWeb^{Log}(O)$ is obtained from $L_O = \langle G_O, P_O\rangle$ as follows:*

% Include the rules of $ERDF$ semantics
`:- import('erdf.rb',rulebase).`
% For each triple $T \in sk(G_O)$ add to $MWeb^{Log}(O)$
T.
% For each simple `r`*-ERDF rule G*`:-`*$L_1,\ldots,L_m \in P_O$ add the rule*
G`:- `$L_1,\ldots,L_m$.
% For each well-typed XML literal lll in G_O add the triple
`lll.[rdf:type->>rdf:XMLLiteral].`
% For each ill-typed XML literal lll in G_O add the triple
`neg lll.[rdf:type->>rdf:Literal].`
% For each plain literal lll in G_O (with or without language tag) add triple
`lll.[rdf:type->>rdfs:Literal].`

The new transformation into extended logic programming of MWeb rulebases obtained from SMEOs, resorts to a 4-ary predicate `'->'(`r`,`p`,`s`,`o`)` denoting that s`[`p `->>` o`]` is true in rulebase r:

Definition 8 (Translation of simple ERDF ontologies into ELP) *Construct ELP Π_O from a simple* `r`-ERDF ontology *O as follows: i) a uses* `property(`P`)` *declaration adds to Π_O the rule* `'->'('Nam_O',`P`,?S,?O) :- '->'('`r_i`',`P`,?S,?O)`, *for each 'r_i' in the from list; ii) a uses* `class(`C`)` *adds to Π_O the rule* `'->'('Nam_O',rdf:type,?S,?C) :- '->'('`r_i`',rdf:type,?S,?C)`, *for each 'r_i' in the from list; iii) replace in $MWeb^{Log}(O)$ recursively the import declarations by the contents of documents; iv) add to Π_O each rule of $MWeb^{Log}(O)$ after replacing a triple* s`[`p `->>` o`]` *by* `'->'('Nam_O',s,p,o)` *and occurrences of* s`[`p `->>` o`]@'`$oname$`'` *by* `'->'('`$oname$`',s,p,o)`.

The semantics of simple modular ERDF ontologies is based on a generalization of Answer Set semantics [3], an equivalent rule-based declarative semantics is specified next and is our major result:

Theorem 1 (Semantics of simple modular ERDF ontologies) *Let $\mathcal{R}$ be a set of simple* `r`-ERDF ontologies *and let $O \in \mathcal{R}$. Consider extended logic program $\Pi_{O,\mathcal{R}} = \bigcup_{O' \in D^{\mathcal{R}}_{O}} ground(\Pi_{O'})$*[4]*. Then,* O entails *a ground simple* `r`-ERDF formula *F w.r.t. $\mathcal{R}$ under the* modular ERDF stable model semantics *[3] iff for every answer set M of $\Pi_{O,\mathcal{R}}$ it is the case that $M \models F$.*

Practical evaluation of the resulting implementation in XSB has shown promising results when compared to Jena, CWM, and Eye.

REFERENCES

[1] A. Analyti, G. Antoniou, C. V. Damásio, and G. Wagner, 'Extended RDF as a Semantic Foundation of Rule Markup Languages', *Journal of Artificial Intelligence Research (JAIR)*, **32**, 37–94, (2008).

[2] A. Analyti, G. Antoniou, C. V. Damásio, and G. Wagner, 'On the Computability and Complexity Issues of Extended RDF', in *Procs. of PRICAI 2008*, volume 5351 of *LNCS*, pp. 5–16. Springer, (2008).

[3] A. Analyti, G. Antoniou, and Carlos V. Damásio, 'A Formal Theory for Modular ERDF Ontologies', in *Web Reasoning and Rule Systems (RR 2009)*, volume 5837 of *LNCS*, pp. 212–226. Springer, (2009).

[4] Anastasia Analyti, Grigoris Antoniou, and Carlos Viegas Damásio, 'A Principled Framework for Modular Web Rule Bases and Its Semantics', in *Proc. of KR-2008*, pp. 390–400. AAAI press, (2008).

---

[4] Grounding is performed with respect to the constants in $\Pi_{O'}$.