

RDF FRAMEWORK INSTITUTIONS

Dorel LUCANU*, Yuan Fang LI**, Jin Song DONG**

* Faculty of Computer Science, "Alexandru Ioan Cuza" University, Romania

** School of Computing, National University of Singapore, Singapore

Corresponding author: Dorel Lucanu: dlucanu@info.uaic.ro

The Semantic Web is a vision of enabling software agents to autonomously understand, process and integrate Web resources. This ability is based on semantically marking up Web resources using ontologies. Ontology languages are based on Resource Description Framework (RDF) and RDF Schema (RDFS), which form the foundation of the Semantic Web. In this paper, we define the RDF framework in terms of institution theory and investigate the exactness property.

Key words: Semantic Web, RDF, RDF Schema, institution.

1. INTRODUCTION

The Semantic Web [2] has attracted much attention as a new generation of the current World Wide Web. It attempts to realize the full potential of the Web by semantically marking up Web resources so that they can be readily processed by software agents on the Web. Ontologies provide such markups for the Semantic Web.

As depicted in Fig. 1, the development of Semantic Web languages takes a layered approach, in which languages higher in the stack are developed based on those below them. The Resource Description Framework (RDF) [9] and RDF Schema (RDFS) [3] are the cornerstone languages of the Semantic Web. They define the set of basic vocabularies and syntax for later ontology languages.

Directly sitting on RDF and RDFS, ontology languages such as DAML+OIL [15], OWL [11] and the Horn-style rules extension SWRL [10] have been developed to improve the expressivity of the Semantic Web. These languages extend the expressivity of RDF and RDFS by defining more language constructs to further classify and relate Web resources.

The Web Ontology language (OWL) is the mainstream ontology language. It has three increasingly expressive sub languages: OWL Lite, DL and Full. By imposing certain restrictions on the use of RDF, RDFS and OWL Full constructs, OWL Lite and DL are decidable.

Recently, researchers have observed some issues related to the layering of OWL languages on top of RDFS [5]. It has been argued by some researchers that the layering of OWL Lite and DL on top of RDFS is not very proper as they redefine the semantics of some of the RDFS vocabularies [5, 14]. Such improper layering may cause interoperability problems between ontologies written in these languages. Hence, it is of fundamental importance to thoroughly study the properties of RDF and RDFS to provide more insight into investigating the layering issue.

The notion of institutions [7] was proposed to formalize the concepts of logical systems. Institutions provide a means of reasoning about software specifications regardless of the logical system. Hence, it serves as a natural candidate to study the relationship among the various SW languages, as they are based on different logical systems (semantics). Institution (co)morphisms [8] capture the migration from one logical system to another.

The rest of the paper is organized as follows. In Section 2, we briefly present the background information on the Semantic Web, the RDF and RDFS languages, institution and institution morphisms. In

Section 3, we present three institutions. The first one is for the bare RDF, which is a bare bone RDF we constructed that only includes the triple format and resource references. Building on the institution of bare RDF, we then develop the institution for the actual RDF, including its vocabularies and defining their semantics. Thirdly, we construct the institution for RDFS by defining semantics for its language constructs. The exactness properties of these institutions are discussed. Finally, Section 4 concludes the paper and discusses future works.

2. PRELIMINARIES

2.1 RDF FRAMEWORK

The Semantic Web is a vision as the new generation of the current Web in which information is semantically marked-up so that intelligent software agents can autonomously understand, process and aggregate data. This ability is realized through the development of a “stack” of languages, as depicted by Berners-Lee in Fig. 1.

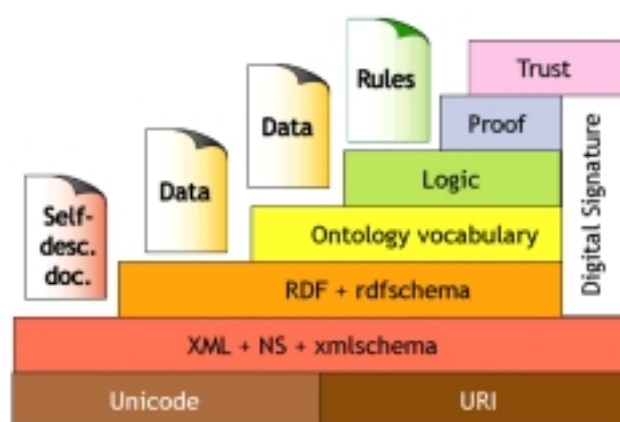


Figure 1

Based on mature technologies such as XML, Unicode and URI (Uniform Resource Identifier), The Resource Description Framework (RDF) [9] is the foundation of later languages in the SW. RDF is a model of metadata defining a mechanism for describing resources that makes no assumptions about a particular application domain. It allows structured and semi-structured data to be mixed and shared across applications. It provides a simple way to make *statements* about Web resources. An RDF document is a collection of *triples*, statements of the form $\langle \text{subject predicate object} \rangle$, where *subject* is the resource we are interested in, *predicate* specifies the property or characteristic of the subject and *object* states the value of the property. RDF also defines vocabularies for constructing containers such bag, sequence, list, etc.

RDF Schema [3] provides additional vocabularies for describing RDF documents. It defines semantical entities such as resource, class, property, literal and various properties about these entities, such as *subClassOf*, *domain*, *range*, etc. In RDF Schema, *resource* is the universe of description. It can be further categorized as class, property, datatype or literal. With these semantical constructs, RDF Schema can be regarded as the basic ontology language.

2.2 INSTITUTIONS

Institutions supply a uniform way for structuring the theories in various logical systems. Many logical systems have been proved to be institutions. Recent research showed that institutions are useful in designing tools supporting verification over multiple logics. The basic reference for institutions is [7]. A well structured approach of the various institution morphisms and many other recent constructions can be found in [8].

A recent application of institutions in formalizing the information integration is given in [6]. The institutions use intensively category theory; we recommend [1] for a detailed presentation of categories and their applications in computer science. An *institution* is a quadruple $\mathfrak{I} = (\text{Sig}, \text{Mod}, \text{Sen}, |=)$, where *Sig* is

a category whose objects are called *signatures*, $Mod : Sig^{op} \rightarrow Cat$ is a functor which associates with each signature Σ a category whose objects are called Σ -*models*, Sen is a functor $Sen : Sig \rightarrow Set$ which associates with each signature Σ a set whose elements are called Σ -*sentences*, and $|\equiv$ is a function which associates with each signature Σ a binary relation $|\equiv_{\Sigma} \subseteq |Mod(\Sigma)| \times |Sen(\Sigma)|$, called *satisfaction relation*, such that for each signature morphism $\phi : \Sigma \rightarrow \Sigma'$ the *satisfaction condition*

$$Mod(\phi^{op})(M') |\equiv_{\Sigma} \varphi \Leftrightarrow M' |\equiv_{\Sigma'} Sen(\phi)(\varphi)$$

holds for each model $M' \in |Mod(\Sigma')|$ and each sentence $\varphi \in |Sen(\Sigma)|$. The functor Mod is defined over the opposite category Sig^{op} because a "translation between vocabularies" $\phi : \Sigma \rightarrow \Sigma'$ defines a forgetful functor $Mod(\phi^{op}) : Mod(\Sigma') \rightarrow Mod(\Sigma)$ such that for each Σ' -model M' , $Mod(\phi^{op})(M')$ is M' viewed as a Σ -model and it is called the *reduct* of M' along ϕ . The functor Sen abstracts the way the sentences are constructed from signatures (vocabularies). The satisfaction condition may be read as " M' satisfies the ϕ -translation of φ iff M' , viewed as a Σ -model, satisfies φ ", i.e., the meaning of φ is not changed by the translation ϕ .

A specification (presentation) is a way to represent the properties of a system independent of model (= implementation). Formally, a *specification* is a pair (Σ, F) , where Σ is a signature and F is a set of Σ -sentences. A (Σ, F) -*model* is a Σ -model M such that $M |\equiv_{\Sigma} \phi$ for each $\phi \in F$. A *theory* is a specification (Σ, F) with F *closed*, i.e., satisfying the property

$$(\forall \phi \text{ a } \Sigma\text{-sentence}) [(\forall M \text{ a } (\Sigma, F)\text{-model}) M |\equiv_{\Sigma} \phi] \Rightarrow \phi \in F.$$

A signature morphism $\phi : \Sigma \rightarrow \Sigma'$ is a *theory morphism* $\phi : (\Sigma, F) \rightarrow (\Sigma', F')$ iff for each Σ -sentence φ , we have $\varphi \in F \Rightarrow \phi(\varphi) \in F'$. We denote by Th the category of theories.

Given an institution, the *theoroidal institution* \mathfrak{S}^{th} of \mathfrak{S} is the institution $I^{th} = (Th, Mod^{th}, Sen^{th}, |\equiv^{th})$, where Mod^{th} is the extension of Mod to theories, Sen^{th} is $sign; Sen$ with $sign : Th \rightarrow Sig$ the functor which forgets the sentences of a theory, and $|\equiv^{th} = |sign|; |\equiv$.

An important property of the logical systems is that the finite limits are preserved by the model functor. This property is called in literature *exactness*. We recall that the functor Mod is defined over Sig^{op} and the limits in this category are in fact colimits in Sig . If the model functor preserves only the pullbacks, then the institution is *semiexact*.

Let $\mathfrak{S} = (Sig, Mod, Sen, |\equiv)$ and $\mathfrak{S}' = (Sig', Mod', Sen', |\equiv')$ be two institutions. An *institution morphism* $(\Phi, \beta, \alpha) : \mathfrak{S} \rightarrow \mathfrak{S}'$ consists of:

1. a functor $\Phi : Sig \rightarrow Sig'$,
2. a natural transformation $\beta : Mod \Rightarrow \Phi^{op}; Mod'$, i.e., a natural family of functors $\beta_{\Sigma} : Mod(\Sigma) \rightarrow Mod(\Phi(\Sigma))$, and
3. a natural transformation $\alpha : \Phi; Sen' \Rightarrow Sen$, i.e., a natural family of functions $\alpha_{\Sigma} : Sen'(\Phi(\Sigma)) \rightarrow Sen(\Sigma)$,

such that the following satisfaction condition holds:

$$M |\equiv_{\Sigma} \alpha_{\Sigma}(\varphi') \Leftrightarrow \beta_{\Sigma}(M) |\equiv_{\Phi(\Sigma)} \varphi'$$

for any Σ -model M in \mathfrak{S} and $\Phi(\Sigma)$ -sentence φ' in \mathfrak{S}' . Usually, the institution morphisms are used to express the embedding relationship. An example of institution morphism is $(\Phi, \beta, \alpha): \mathfrak{S}^{th} \rightarrow \mathfrak{S}$ which express the embedding of \mathfrak{S} in \mathfrak{S}^{th} .

We show how a theory (Σ_0, F_0) and a semantical constraint can define an institution $\mathfrak{S}(\Sigma_0, F_0)$. A *semantical constraint* is a map $\llbracket _ \rrbracket_c$ which associates a subcategory $\llbracket (\Sigma, F) \rrbracket_c \subseteq Mod^{th}(\Sigma, F)$ with each theory (Σ, F) , such that $Mod(\phi)(M') \in \llbracket (\Sigma, F) \rrbracket_c$ for all $\phi: (\Sigma, F) \rightarrow (\Sigma', F')$ and $M' \in \llbracket (\Sigma', F') \rrbracket_c$. The constraints defined in [7] are a particular case of semantical constraints defined here when the subcategory can be syntactically represented. The institution $\mathfrak{S}(\Sigma_0, F_0)$ is defined as follows:

1. the category of signatures is the comma category $(\Sigma_0, F_0) \downarrow Th$, where the objects are theory morphisms $f: (\Sigma_0, F_0) \rightarrow (\Sigma, F)$, and the arrows $\phi: f \rightarrow f'$ are consisting of theory morphisms $\phi: (\Sigma, F) \rightarrow (\Sigma', F')$ such that $f; \phi = f'$,
2. the model functor $Mod(\mathfrak{S}(\Sigma_0, F_0))$ maps each signature $f: (\Sigma_0, F_0) \rightarrow (\Sigma, F)$ into the subcategory $\llbracket (\Sigma, F) \rrbracket_c$,
3. the sentence functor $Sen(\mathfrak{S}(\Sigma_0, F_0))$ maps a signature $f: (\Sigma_0, F_0) \rightarrow (\Sigma, F)$ into the set of Σ -sentences,
4. the satisfaction relation is defined by $M \models_f \varphi$ iff $M \models_\Sigma \varphi$.

Note that the semantical constraint is required only for the theories (Σ, F) for that there exists a theory morphism $f: (\Sigma_0, F_0) \rightarrow (\Sigma, F)$. If $\phi: (\Sigma_0, F_0) \rightarrow (\Sigma'_0, F'_0)$ is a theory morphism, then there is an institution morphism $(\Phi, \beta, \alpha): \mathfrak{S}(\Sigma'_0, F'_0) \rightarrow \mathfrak{S}(\Sigma_0, F_0)$.

3. INSTITUTION OF RDF FRAMEWORK

In this section, we define the institutions for the languages RDF and RDF Schema. The construction of these institutions is divided into three parts. Firstly, we construct a bare-bone institution for RDF logic, capturing only the very essential concepts in RDF, namely the resource references and the triples format. This logic then serves as the basis on which the institutions of the actual RDF and RDF Schema are constructed.

3.1 BARE RDF INSTITUTION

The Bare RDF institution **BRDF** is a bare-bone institution with resource references sets as the only signatures and triples as sentences. We use it as a basis over which we develop the other institutions involved in Semantic Web.

A *signature* RR in **BRDF** is a set of resource references. A signature morphism $\phi: RR \rightarrow RR'$ is an arrow in *Set*. The *RR-sentences* are triples of the form (sn, pn, on) , where $sn, pn, on \in RR$. Usually, sn is for subject name, pn is for property (predicate) name, and on is for object name. *RR-models* I are tuples $I = (R_I, P_I, S_I, ext_I)$, where R_I is a set of *resources*, P_I is a subset of R_I - the set of *properties*, $S_I: RR \rightarrow R_I$ is a function that maps each resource reference to some resource, and $ext_I: P_I \rightarrow \wp(R_I \times R_I)$ is an extension function mapping each property to a set of pairs of resources that it relates. An *RR-homomorphism* $h: I \rightarrow I'$ between two *RR-models* is a function $h: R_I \rightarrow R_{I'}$ such that $h(P_I) \subseteq h(P_{I'})$, $S_I; h = S_{I'}$, and $ext_I; (h \times h) = ext_{I'}$. If $\phi: RR \rightarrow RR'$ is a signature morphism and I'

an RR' -model, then the reduct of I' along ϕ is the RR -model I defined as follows: $R_I = R_{I'}$, $P_I = P_{I'}$, $S_I = S_{I'}|_{RR}$, $ext_I = ext_{I'}$. The *satisfaction* is defined as follows:

$$I \models_{RR} (sn, pn, on) \text{ iff } (S_I(sn), S_I(on)) \in ext_I(S_I(pn)),$$

that (sn, pn, on) is satisfied if and only if the pair consisting of the resources associated with the subject name sn and the object name on is in the extension of pn .

Since the signatures in BRDF are sets, it follows that both the category of signatures and category of theories are cocomplete.

Theorem 1. BRDF is not exact.

Proof. Let $\{RR_1, RR_2\}$ be a discrete diagram and $(\psi_1 : RR \rightarrow RR_1, \psi_2 : RR \rightarrow RR_2)$ its limit in Set^{op} . If I_j is a RR_j -model, $j=1,2$, such that $R_{I_1} \neq R_{I_2}$, then there is no an RR -model I with $Mod(\psi_j)(I) = I_j, j=1,2$.

Theorem 2. BRDF is semiexact.

Proof. Let $(\psi_1 : RR \rightarrow RR_1, \psi_2 : RR \rightarrow RR_2)$ be the pullback of the following diagram in Set^{op} $\phi_1 : RR_1 \rightarrow RR_0 \leftarrow RR_2 : \phi_2$. If I_j is a RR_j -model, $j=1,2$, such that $Mod(\phi_1)(I_1) = Mod(\phi_2)(I_2)$ then $R_{I_1} = R_{I_2}$. We can define an RR -model I with $Mod(\psi_j)(I) = I_j, j=1,2$. We set $R_I = R_{I_1}$, $P_I = P_{I_1} \cup P_{I_2}$, $S_I(rr) = S_{I_j}(rr)$ if $rr \in RR_j$, and $ext_I(p) = ext_{I_j}(p)$ if $p \in P_j, j=1,2$.

Corollary. BRDFth is semiexact.

3.2 RDF INSTITUTION

The RDF institution $\mathfrak{S}(\text{RDF})$ is defined using the construction we defined in Section 2 starting from a theory RDF and a semantical constraint $\llbracket _ \rrbracket_{\text{RDF}}$. RDF theory is $\text{RDF} = (V_{\text{RDF}}, T_{\text{RDF}})$, where the RDF vocabulary V_{RDF} includes the following items:

`rdf:type`, `rdf:Property`, `rdf:value`,
`rdf:Statement`, `rdf:subject`, `rdf:predicate`, `rdf:object`,
`rdf:List`, `rdf:first`, `rdf:rest`, `rdf:nil`,
`rdf:Seq`, `rdf:Bag`, `rdf:Alt`, `rdf:_1` `rdf:_2` ...

and T_{RDF} includes the following triples:

`(rdf:type, rdf:type, rdf:Property)`, `(rdf:first, rdf:type, rdf:Property)`,
`(rdf:subject, rdf:type, rdf:Property)`, `(rdf:rest, rdf:type, rdf:Property)`,
`(rdf:predicate, rdf:type, rdf:Property)` `(rdf:nil, rdf:type, rdf:List)`,
`'`, `(rdf:_1, rdf:type, rdf:Property)`,
`(rdf:object, rdf:type, rdf:Property)`, `(rdf:_2, rdf:type, rdf:Property)`,
`(rdf:value, rdf:type, rdf:Property)`, ...

We suppose that there is given a set R_{RDF} of RDF resources and a function $S_{\text{RDF}} : V_{\text{RDF}} \rightarrow R_{\text{RDF}}$ which associates a resource with each RDF symbol. It is easy to see that R_{RDF} and S_{RDF} can be extended to a RDF-model I_{RDF} .

For each theory such that there is theory morphism, we consider the semantical constraint $\llbracket (RR, T) \rrbracket_{\text{RDF}}$ as consisting of those (RR, T) -models I such that:

1. R_I includes R_{RDF} , and the restriction of S_I coincides with S_{RDF} ,

2. if $p \in P_i$ then $(p, \text{rdf}:\text{Property}) \in \text{ext}_i(\text{rdf}:\text{type})$

Since f is theory morphism, the restriction of I to V_{RDF} is an RDF-model.

Example. The *reification* is given by the RDF signature $\text{reification} : \text{RDF} \rightarrow (\{r, s, p, o\}, T)$, where T further to T_{RDF} includes the triples:

$(r, \text{rdf}:\text{type}, \text{rdf}:\text{Statement}),$
 $(r, \text{rdf}:\text{subject}, s),$
 $(r, \text{rdf}:\text{predicate}, p),$
 $(r, \text{rdf}:\text{object}, o).$

We denote by $\mathfrak{S}(\text{RDF})$ the institution defined by the theory RDF together with the semantical constraint $\llbracket _ \rrbracket_{\text{RDF}}$ using the method presented in Section 2.

Proposition 1. There is an institution morphism $\mathfrak{S}(\text{RDF}) \rightarrow \text{BRDF}$.

Theorem 3. $\mathfrak{S}(\text{RDF})$ is exact.

Proof. Any two RDF signatures $f_1 : \text{RDF} \rightarrow (RR_1, T_1)$ and $f_2 : \text{RDF} \rightarrow (RR_2, T_2)$ have a pushout in $\text{Th}(\text{BRDF})$ and we apply Corollary of Theorem 2.

3.2 RDF SCHEMA INSTITUTION

RDF Schema defines additional language constructs for the RDF language. It expands the expressivity of RDF by introducing the concept of universe of resources $\text{rdfs}:\text{Resource}$, the classification mechanism $\text{rdfs}:\text{Class}$, and a set of properties that relate them $\text{rdfs}:\text{subClassOf}$, $\text{rdfs}:\text{domain}$, $\text{rdfs}:\text{range}$. Hence, it is natural for the RDF Schema institution $\mathfrak{S}(\text{RDFS})$ to be developed on top of $\mathfrak{S}(\text{RDF})$, with some more semantical constraints added to capture the semantics of RDF Schema language constructs.

The RDF Schema theory $\text{RDFS} = (V_{\text{RDFS}}, T_{\text{RDFS}})$ is composed of the RDF Schema vocabulary V_{RDFS} including V_{RDF} together with

$\text{rdfs}:\text{domain}, \text{rdfs}:\text{range}, \text{rdfs}:\text{Resource}, \text{rdfs}:\text{Literal},$
 $\text{rdfs}:\text{Datatype}, \text{rdfs}:\text{Class}, \text{rdfs}:\text{subClassOf},$
 $\text{rdfs}:\text{subPropertyOf}, \text{rdfs}:\text{member}, \text{rdfs}:\text{Container},$
 $\text{rdfs}:\text{ContainerMembershipProperty}$

and the triples T_{RDFS} including T_{RDF} together with:

$(\text{rdf}:\text{type}, \text{rdfs}:\text{domain}, \text{rdfs}:\text{Resource}),$
 $(\text{rdfs}:\text{domain}, \text{rdfs}:\text{domain}, \text{rdf}:\text{Property}),$
 $(\text{rdfs}:\text{range}, \text{rdfs}:\text{domain}, \text{rdf}:\text{Property}),$
 $(\text{rdfs}:\text{subPropertyOf}, \text{rdfs}:\text{domain}, \text{rdf}:\text{Property}),$
 $(\text{rdfs}:\text{subClassOf}, \text{rdfs}:\text{domain}, \text{rdfs}:\text{Class}),$
 $(\text{rdf}:\text{subject}, \text{rdfs}:\text{domain}, \text{rdf}:\text{Statement}),$
 $(\text{rdf}:\text{predicate}, \text{rdfs}:\text{domain}, \text{rdf}:\text{Statement}),$

...

Here we present only a part of triples added to T_{RDFS} . The complete set of triples defining T_{RDFS} can be found in [9]. As for RDF, we suppose that there is given a set R_{RDFS} of RDF Schema resources and a

function $S_{RDFS} : V_{RDFS} \rightarrow R_{RDFS}$ which associates a resource with each RDF symbol and that satisfies $S_{RDFS}|_{V_{RDF}} = S_{RDF} \cdot R_{RDFS}$ and S_{RDFS} can be extended to a RDFS-model I_{RDFS} .

For each theory such that there is a theory morphism $f : RDFS \rightarrow (RR, T)$, we consider the semantical constraint $\llbracket (RR, T) \rrbracket_{RDFS}$ as consisting of those (RR, T) -models I such that:

1. R_I includes R_{RDFS} , and the restriction of S_I coincides with S_{RDFS} ,
2. $ext_I(\text{rdfs:Resource}) = R_I$
3. $(\forall x, y, u, v \in R_I)(x, y) \in ext_I(\text{rdfs:domain}) \wedge (u, v) \in ext_I(x) \Rightarrow u \in ext_I(y)$
4. $(\forall x, y, u, v \in R_I)(x, y) \in ext_I(\text{rdfs:range}) \wedge (u, v) \in ext_I(x) \Rightarrow v \in ext_I(y)$
5. $(\forall x, y \in R_I)(x, y) \in ext_I(\text{rdfs:subClassOf}) \Rightarrow ext_I(x) \subseteq ext_I(y)$
6. $(\forall x \in ext_I(\text{rdfs:Class})(x, \text{rdfs:Resource}) \in ext_I(\text{rdfs:subClassOf}))$
7. $(\forall x, y \in R_I)(x, y) \in ext_I(\text{rdfs:subPropertyOf}) \Rightarrow ext_I(x) \subseteq ext_I(y)$
8. $(\forall x \in ext_I(\text{rdfs:ContainerMembershipProperty}))$
 $(x, \text{rdfs:member}) \in ext_I(\text{rdfs:subPropertyOf})$

Proposition 2. There is an institution morphism $\mathfrak{S}(RDFS) \rightarrow \mathfrak{S}(RDF)$.

Proof. The institution morphism is given by the inclusion $RDF \rightarrow RDFS$.

Theorem 4. $\mathfrak{S}(RDFS)$ is exact.

It is known that unsorted first-order logic is only semiexact. Hence the embedding of RDF Schema in first-order logic can be inappropriate for many software applications. In turn, the sorted first-order is exact and therefore is more appropriate to embed RDF Schema.

4. CONCLUSION

RDF and RDF Schema define the basic vocabularies for describing Semantic Web resources. They are the foundation on which more expressive ontology languages are constructed. Institutions and institution morphisms are used to formally represent and reason about the migration of different logical systems. In this paper, we use institutions to represent languages RDF and RDF Schema and use institution morphisms to relate them. The exactness property of various RDF institutions is also discussed. One interesting conclusion we reached is that the institution for RDFS is exact. As unsorted first-order logic is known to be semiexact, the exactness of RDF Schema implies that there might be issues of applying some first-order logic reasoning tools to RDF Schema ontologies.

As stated previously, the layering of the ontology language OWL on top of RDF Schema has been criticized to be improper, possible follow-up work would be to further investigate the layering issue.

REFERENCES

1. BARR, M., WELLS, Ch., *Category Theory for Computing Science*, Les Publications CRM, Montreal, third edition, 1999.
2. BERNERS-LEE, T., HENDLER, J., LASSILA, O., *The Semantic Web*, Scientific American, May 2001.
3. BRICKLEY, D., GUHA, R.V., *Description Framework (RDF) Schema Specification 1.0*, <http://www.w3.org/TR/rdf-schema/>, February 2004.
4. DIACONESCU, R., *Grothendieck institutions*, Applied Categorical Structures, 10, pp. 383--402, 2002.
5. de BRUIJN, J., LARA, R., POLLERERS, A., FENSEL, D., *OWL DL vs. OWL flight: conceptual modeling and reasoning the semantic Web*, Proc. of the 14th International Conference on World Wide Web (WWW2005), Chiba, Japan, May 10-14, 2005, pp. 623--632.

6. GOGUEN, J., *Three Perspectives on Information Integration*, Proc of Semantic Interoperability and Integration, Schloss Dagstuhl, Germany, 2005.
7. GOGUEN, J., BURSTALL, R., *Institutions: Abstract Model Theory for Specification and Programming*, Journal of the Association for Computing Machinery, 39(1), pp.95--146, 1992.
8. GOGUEN, J., ROȘU, G., *Institution Morphism*, Formal Aspects of Computing, 13, pages 274--307, 2002.
9. HAYES P., *RDF Semantics*, <http://www.w3.org/TR/rdf-mt/>, February 2004.
10. HORROCKS, I., PATEL-SCHNEIDER, P. F., BOLEY, H., TABET, S., GROSOFF, B., DEAN, M., *SWRL: A semantic web rule language combining OWL and RuleML*, <http://www.daml.org/2004/11/fof/rules-all.html>, November 2004.
11. HORROCKS, I., PATEL-SCHNEIDER, P. F., van HARMELEN, F., *From SHIQ and RDF to OWL: The making of a web ontology language*, Journal of Web Semantics, 1(1):7--26, 2003.
12. KLYNE, G., CARROLL, J. (editors), *Resource Description Framework (RDF): Concepts and Abstract Syntax*, W3C Recommendation, <http://www.w3.org/TR/rdf-concepts/>, 2004.
13. LUCANU, D., LI, Y.F., DONG, J.S., *Web Ontology Verification and Analysis in the Z Framework*, Technical Report TR 05-01, University "Alexandru Ioan Cuza" of Iași, Romania, January 2005, <http://thor.info.uaic.ro/~tr/tr05-01.ps>.
14. PATEL-SCHNEIDER, P. F., HAYES, P., HORROCKS, I. (editors), *OWL Web Ontology Semantics and Abstract Syntax*, <http://www.w3.org/TR/2004/REC-owl-semantics-20040210/>, 2004.
15. van HARMELEN, F., PATEL-SCHNEIDER, P. F., HORROCKS, I. (editors), *Reference description of the DAML+OIL ontology markup language*, <http://www.daml.org/2001/03/reference.html>, March, 2001.

Received October 17, 2005