A Practical Implementation of Contextual Reasoning for the Semantic Web

Sahar Aljalbout, Gilles Falquet and Didier Buchs
Centre Universitaire d'Informatique, University of Geneva, 7 route de Drize, Geneva, Switzerland

Keywords: Contextual Reasoning, Contexts, Contextual OWL.

Abstract: Dealing with context-sensitive information is a crucial aspect in the management of semantic web data. Despite the importance of this topic, there is so far no accepted consensus regarding the precise way of encoding and even more reasoning on contextual knowledge. In this paper, we introduce an approach to reason over contextual knowledge in RDF, while committing to the semantics of a contextual description logic. The lines of this paper are many folds. First, we present an extension of OWL 2 DL for contexts, that we call OWL 2 DL^C. It is a two-dimensional web ontology language with one dimension for contextualized object knowledge and one dimension for contexts. Second, we define an OWL^C profile for contextual reasoning, similar to OWL 2 RL. And finally, we demonstrate that the model can be practically implemented using existing semantic web technologies, especially using SPIN rules.

1 INTRODUCTION

The contextuality of knowledge is the problem of the general inability of determining the meaning of a piece of information or verifying its validity, without assuming the context in which this information has been stated, and thereby, in which it should be interpreted (Klarman and Gutiérrez-Basulto, 2011). The primary role of contexts in the semantic web is to provide additional knowledge about individual triples, such as the source, the occurring time or place, certainty etc.

Although many data providers and semantic web practitioners have attempted local approaches for treating contexts representation; there is, so far, no consensus regarding the precise ways of encoding and much less reasoning, on contextual knowledge. Nevertheless, the representation of contexts in the semantic web has been considered separately as, first, a data problem, giving rise to several proposals to encode contexts into RDF (Nguyen et al., 2014)(Welty et al., 2006) (Noy et al., 2006), and second, as a theoretical problem where several attempts to include the context dimension to description logics have emerged (Kutz et al., 2004) (Benslimane et al., 2006) (Klarman and Gutiérrez-Basulto, 2011) .

In this paper, we propose an approach to reason over contextual knowledge in RDF using SPIN[1], while committing to the semantics of a contextual description logic. The key idea behind this approach is the definition of a formally solid contextual model, but also practically applicable to data while using existing semantic web languages and tools. Throughout this work, we adopted McCarthy's theory of contexts (McCarthy, 1987), primarily because this theory offers an instrumental view of contexts, where contexts are considered as formal objects, describable in first-order logic languages. In order to achieve our goal, we do the following:

- First, we propose a two-dimensional web ontology language OWL 2 DL^C, similar to OWL 2 DL[2] but based on a two dimensional description logic (2DL) (Klarman and Gutiérrez-Basulto, 2011). The idea is to have two interacting languages: the core and the context language that we present in section 3.
- Second, in section 4, we propose a profile aimed at applications that require scalable reasoning without sacrificing too much expressive power that we call OWL^C.
- Finally, the practical implementation of the formal model arises two important questions: how to encode the contexts in the RDF data model? And how to practically implement the new con-

---

[1] http://spinrdf.org

[2] http://www.obitko.com/tutorials/ontologies-semantic-web/owl-dl-semantics.html

Aljalbout, S., Falquet, G. and Buchs, D.
A Practical Implementation of Contextual Reasoning on the Semantic Web.
DOI: 10.5220/0006936802550262
In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2018) - Volume 2: KEOD*, pages 255-262
ISBN: 978-989-758-330-8

textual rules correlated with the generation of new objects? As a first attempt, we extended the fluent model (Aljalbout and Falquet, 2017) to represent many dimensions of contexts. We also demonstrate, that using SPIN, the contextual entailments rules can be feasibly implemented without the need for creating new contextual reasoners (section 5).

2 REQUIREMENTS FOR CONTEXTS REPRESENTATION AND REASONING

In the search for a suitable knowledge representation and reasoning model, a relevant question arises about the requirements that would serve best achieving a community consensus on this topic. According to our opinion, the following requirements should be considered. The following list should therefore not be considered as final, but rather as a starting point in this discussion. The requirements are divided into two groups respectively: requirements for contextual knowledge representation and reasoning requirements.

Contextual Knowledge Representation Requirements:

- Distinction between the object knowledge and the contexts knowledge: the union of the vocabularies used for each level must be disjoint. One should unequivocally recognize if a statement belongs to the contextualized object knowledge, or to the contexts' knowledge.
- Expressiveness of the model: the model must be able to deal with polymorphism when adding new dimensions of contexts. Additionally, relations between contexts must be clearly and explicitly specified within the model.
- Representation compactness: Technically speaking, a model that introduces many properties and objects can lead to undesirable graph size increases, which oftentimes cause detrimental memory performance. The worst-case scenario could lead to an explosion of the number of triples.

Contextual Reasoning Requirements:

- Reasoning should take into the account the contextual meta-knowledge and relations between contexts.
- The contextual layer should not increase the complexity of reasoning. Given the fact that some of the semantic web languages already exhibit quite high complexity (e.g., OWL 2, based on the SROIQ DL, is 2NExpTime-complete (Kazakov, 2008)), we believe that the contextual layer must be added without any increase in complexity.

3 OWL 2 DL^C: A CONTEXTUAL TWO- DIMENSIONAL WEB ONTOLOGY LANGUAGE

OWL 2 DL was designed to support the existing description logic business segment and has desirable computational properties for reasoning systems. OWL 2 DL is so named due to its correspondence with description logics. In this section, we introduce an extension of OWL 2 DL for contexts, that we call OWL 2 DL^C. It contains new contextual constructors and is based on a two-dimensional description logic (Klarman and Gutiérrez-Basulto, 2011) with two interacting languages: the core language intended to define a context-dependent description of the domain concepts, roles, axioms and the context language used to express knowledge about contexts.

A contextualized vocabulary is therefore a pair of DL signatures $(\langle N_C, N_R, N_I \rangle, \langle N_{KC}, N_{KR}, N_{KI} \rangle)$ where N_C (resp. N_{KC}) is a set of domain (resp. context) concept names, N_R (N_{KR}) is a set of domain (context) role names, and N_I (N_{KI}) is a set of domain (context) individuals names.

3.1 General Syntax

A concept (resp. role) expression is either

- a DL concept expression (resp. role) on the core signature $\langle N_C, N_R, N_I \rangle$
- or an expression of the form $[K]C$ or $\langle K \rangle C$ (resp. $[K]R$ or $\langle K \rangle R$) where K is a concept expression on the context signature $\langle N_{KC}, N_{KR}, N_{KI} \rangle$, and C is a concept expression (resp. role expression) on the core signature $\langle N_C, N_R, N_I \rangle$.

An axiom expression is either

- a DL axiom expression on the core signature $\langle N_C, N_R, N_I \rangle$
- an expression of the form $K : \phi$ where K is a context name (an element of N_{KI}) or a concept expression on the context signature $\langle N_{KC}, N_{KR}, N_{KI} \rangle$, and ϕ is a concept axiom ($C \sqsubseteq D$, $C \equiv D$, $C\ disjoint\ D$) or a role axiom ($R \sqsubseteq S$, , $functional(R)$, $transitive(R)$,...) or a class or role assertion ($C(a)$, $R(a,b)$) defined on the core signature with contextual concept and role expressions. Such an expression states that the axiom

ϕ holds in the contexts that belong to the context class K. For instance

$$before1970 : CanVote \sqsubseteq Aged21orMore$$

states that the axiom $CanVote \sqsubseteq Aged21orMore$ holds in the temporal context *before1970*.

3.2 Semantics

A contextual interpretation is a pair of interpretations $\mathcal{M} = (\mathcal{I}, \mathcal{J})$ where $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}[\cdot]})$ is the core interpretation, $\mathcal{J} = (\Omega, \cdot^{\mathcal{J}})$ is the context interpretation, and $\Delta \cap \Omega = \emptyset$. $\cdot^{\mathcal{I}[\cdot]}$ is a family of interpretation functions, one for each context $k \in \Omega$. $.^{\mathcal{J}}$ is the (non-contextual) interpretation function of every context in the context language.

The interpretation of the class constructors of the core language are the following:

- $\top^{\mathcal{I}[k]} = \Delta$
- $\bot^{\mathcal{I}[k]} = \emptyset$,
- $(C \sqcup D)^{\mathcal{I}[k]} = C^{\mathcal{I}[k]} \cup D^{\mathcal{I}[k]}$
- $(C \sqcap D)^{\mathcal{I}[k]} = C^{\mathcal{I}[k]} \cap D^{\mathcal{I}[k]}$
- $(\neg C)^{\mathcal{I}[k]} = \Delta^{\mathcal{I}[k]} \setminus C^{\mathcal{I}[k]}$
- $(\forall R.C)^{\mathcal{I}[k]} = \{x \in \Delta | \forall y : (x,y) \in R^{\mathcal{I}[k]} \rightarrow y \in C^{\mathcal{I}[k]}\}$
- $(\exists R.C)^{\mathcal{I}[k]} = \{x \in \Delta | \exists y \in \Delta : (x,y) \in R^{\mathcal{I}[k]} \wedge y \in C^{\mathcal{I}[k]}\}$
- $\{i_1, i_2, \ldots, i_k\}^{\mathcal{I}[k]} = \{i_1^{\mathcal{I}[k]}, i_2^{\mathcal{I}[k]}, \ldots, i_k^{\mathcal{I}[k]}\}$

where $C \in N_C$, $D \in N_C$, $R \in N_R$ and $i_k \in N_I$.

In the following we will consider only contextual interpretations that satisfy the *rigid designator hypothesis* (LaPorte, 2006), i.e. $i^{\mathcal{I}[k]} = i^{\mathcal{I}[k']}$ for any $i \in N_I$, $k \in \Omega$, and $k' \in \Omega$.

The interpretation of the context-based concept and role forming operators is as follows:

- $(\langle C \rangle D)^{\mathcal{I}[k]} = \{x \in \Delta | \exists y \in C^{\mathcal{J}} \ : x \in D^{\mathcal{I}[y]}\}$
- $([C]D)^{\mathcal{I}[k]} = \{x \in \Delta | \forall y \in C^{\mathcal{J}} \rightarrow x \in D^{\mathcal{I}[y]}\}$
- $(\langle C \rangle R)^{\mathcal{I}[k]} = \{(x,z) \in \Delta \times \Delta | \exists y \in C^{\mathcal{J}} \ : (x,z) \in R^{\mathcal{I}[y]}\}$
- $([C]R)^{\mathcal{I}[k]} = \{(x,z) \in \Delta \times \Delta | \forall y \in C^{\mathcal{J}} \rightarrow (x,z) \in R^{\mathcal{I}[y]}\}$

One can observe that the interpretations of these constructors are independent of the context k. In fact, these constructors yield concepts that are non-contextual (or context independent).

On the other hand, the great appeal of the context theory (McCarthy, 1987) stems from the postulate that declares a context as a formal object, as a consequence it can have its own properties and relations with other contexts. The axioms of the contexts language are formulas:

$$\mathrm{A} \sqsubseteq \mathrm{B} \mid \mathrm{C(a)}$$

where A $\in N_{KC}$, B $\in N_{KC}$, C $\in N_{KC}$, a $\in N_{KI}$. As we can see, the interpretation of the context language is standard (non-contextual).

A contextual axiom $K : \phi$ is satisfied by an interpretation $\mathcal{M}$ if in every context k that belongs to the interpretation of K, the interpretation in k of the concepts, roles and individuals that appear in ϕ satisfy the axiom condition

- $\mathcal{M} \models K : C \sqsubseteq D$ iff $\forall k \in K^{\mathcal{J}} : C^{\mathcal{I}[k]} \subseteq D^{\mathcal{I}[k]}$, where $C \in N_C$ and $D \in N_C$
- $\mathcal{M} \models K : R \sqsubseteq S$ iff $\forall k \in K^{\mathcal{J}} : R^{\mathcal{I}[k]} \subseteq S^{\mathcal{I}[k]}$, where $R \in N_R$ and $S \in N_R$
- $\mathcal{M} \models K : prop(R)$ iff $\forall k \in K^{\mathcal{J}} : R^{\mathcal{I}[k]}$ has the property *prop*, where *prop* can be *functional*, *transitive*, *reflexive*, etc.
- $\mathcal{M} \models K : C(a)$ iff $\forall k \in K^{\mathcal{J}} : C(a)^{\mathcal{I}[k]}$
- $\mathcal{M} \models K : R(a,b)$ iff $\forall k \in K^{\mathcal{J}} : R(a,b)^{\mathcal{I}[k]}$

(if K is not a concept expression but a context name k, $K^{\mathcal{J}}$ designates the singleton $\{k^{\mathcal{J}}\}$ in the above expressions).

3.3 OWL 2 DL^C Abstract Syntax

The contextual extension of the web ontology language OWL 2 DL^C is done systematically by adding four new abstract syntax constructors whose semantics is given in table 1. The OWL frame-like abstract syntax is given in the first column, and the contextual description logic (CDL) syntax is given in the second column.

Table 1: New constructs of the OWL language.

OWL 2 DL^C Abstract syntax	CDL syntax
SomeConceptValuesFromContext(D[C])	$\langle C \rangle D$
AllConceptValuesFromContext(D[C]))	$[C]D$
SomePropertyValuesFromContext(p[C])	$\langle C \rangle p$
AllpropertyValuesFromContext(p[C])	$[C]p$

4 OWL^C: A PROFILE FOR SCALABLE CONTEXTUAL REASONING

The evolving OWL 2 standard comes with a profile called OWL 2 RL. According to the OWL 2 RL W3C page[3], the OWL 2 RL profile is aimed at applications that require scalable reasoning without sacrificing too much expressive power. In this section, we define a profile for the contextual web ontology language that we defined in section 3, by adapting the idea of OWL 2 RL to OWL 2 DL^C. This is achieved by restricting the use of constructs to certain syntactic positions, exactly as in OWL 2 RL[4]. We limit our description to the ALCO fragment which is proven to be decidable, with a complexity of reasoning NEXPTIME-complete (Klarman and Gutiérrez-Basulto, 2011).

Reasoning with OWL^C is divided into two parts: reasoning for the object knowledge and reasoning for the contexts knowledge. However, reasoning for the contexts language is similar to classical reasoning in OWL 2 RL, so we will introduce the rules of the core language. In the original version of OWL 2 RL, the rules are given as universally quantified first-order implications over a ternary predicate T. This predicate represents a generalization of RDF triples thus, *T(s,p,o)* represents a generalized RDF triple with the subject s, predicate p, and the object o. Variables in the implications are preceded with a question mark. To be able to represent contexts, we introduce a quaternary predicate $Q(s, p, o, co)$ where s is the subject, p is the predicate, o is the object and co is the context for which the predicate holds. If the ontology has multiple contextual dimensions (e.g. time and provenance) co must be understood as $co_1, \ldots, co_m$ and hence Q as a $m+3$-ary predicate.

We divided the rules into two categories. In table 2, we redefine the semantics of the classical OWL 2 RL rules of the ALCO fragment by including the contextual semantics described in section 3 and in table 3, we introduce new contextual rules for the new concept forming operators that we presented in the previous section. Syntactic restrictions are applied to the new constructors: an existential contextual restriction ($\langle C \rangle D$, $\langle C \rangle R$) may only appear in the left-hand side of a subclass axiom, whereas a universal contextual restriction ($[C]D$, $[C]R$) may only appear in the right-hand side.

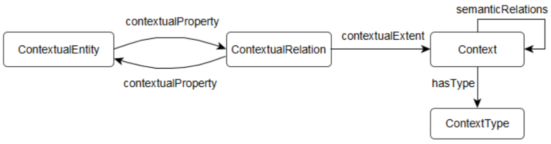


Figure 1: The design pattern for contextual property assertions.

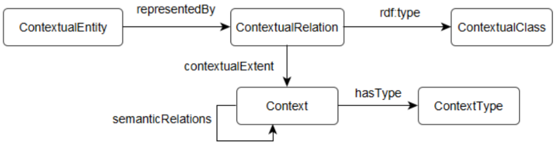


Figure 2: The design pattern for contextual class assertions.

5 IMPLEMENTATION

In this section, we apply the logical model proposed in section 3 and 4 on the RDF data model. This involves the following: i) the choice of a context representation method that fits the requirements of section 2 ii) mapping the 2DL to the chosen method iii) choosing a way to implement the entailment rules.

5.1 Contexts Representation

We previously presented in (Aljalbout and Falquet, 2017) a pattern for the representation of temporal properties (i.e. fluent) in RDF. This pattern was implemented in a historical knowledge base (Aljalbout and Falquet, 2018). In this section, we extend this pattern to represent contextual classes and properties assertions with many dimensions of contexts. Figure 1 shows the design pattern of a contextual property assertion and figure 3 shows the design pattern of a contextual class assertion.

We used the standard mapping of OWL to RDF[5] to map the DL formalization presented in section 3 with the RDF pattern presented in section 5.1. Table 4 and 5 illustrates respectively the mapping of the two languages to RDF :

- $CE[co]$ denotes a contextual class expression;
- $OPE[co]$ denotes a contextual object property expression;
- $T : O \to T(O)$ where T maps a structural element specification E from the ontology O to a set of triples $T(E)$ in the RDF graph.

Due to space limitations in the table, we use the prefix owl^c instead of owl-contextual.

---

[3] https://www.w3.org/TR/owl2-profiles/

[4] https://www.w3.org/TR/owl2-profiles/#Feature_Overview_3

[5] https://www.w3.org/TR/owl2-mapping-to-rdf/

Table 2: Part of the entailment rules for the core language.

	IF	**THEN**
cls-com $\neg C$	T(?c_1, owl:complementOf, ?c_2) Q(?x, rdf:type, ?c_1, ?co) Q(?x, rdf:type, ?c_2, ?co)	false
cls-int1 $C \sqcap D$	T(?c, owl:intersectionOf, ?x) LIST[?x, ?c1, ..., ?cn] Q(?y, rdf:type, ?c1, ?co) Q(?y, rdf:type, ?c2, ?co) ... Q(?y, rdf:type, ?cn, ?co)	Q(?y, rdf:type, ?c, ?co)
cls-int2 $C \sqcap D$	T(?c, owl:intersectionOf, ?x) LIST[?x, ?c_1, ..., ?c_n] Q(?y, rdf:type, ?c, ?co)	Q(?y, rdf:type, ?c_1, ?co) Q(?y, rdf:type, ?c_2, ?co) ... Q(?y, rdf:type, ?c_n, ?co)
cls-uni $C \sqcup D$	T(?c, owl:unionOf, ?x) LIST[?x, ?c_1, ..., ?c_n] Q(?y, rdf:type, ?c_i, ?co)	Q(?y, rdf:type, ?c, ?co)
cls-svf1-1 $\exists R.C$	T(?x, owl:someValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, ?p, ?v, ?co) Q(?v, rdf:type, ?y, ?co)	Q(?u, rdf:type, ?x, ?co)
cls-svf1-2 $\exists R.C$	T(?x, owl:someValuesFrom, ?y) T(?x, owl:onProperty, ?p) T(?u, ?p, ?v) Q(?v, rdf:type, ?y, ?co)	Q(?u, rdf:type, ?x, ?co)
cls-svf1-3 $\exists R.C$	T(?x, owl:someValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, ?p, ?v, ?co) T(?v, rdf:type, ?y)	Q(?u, rdf:type, ?x, ?co)
cls-avf-1 $\forall R.C$	T(?x, owl:allValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, rdf:type, ?x, ?co) Q(?u, ?p, ?v, ?co)	Q(?v, rdf:type, ?y, ?co)
cls-avf-2 $\forall R.C$	T(?x, owl:allValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, rdf:type, ?x, ?co) T(?u, ?p, ?v)	Q(?v, rdf:type, ?y, ?co)
cls-avf-3 $\forall R.C$	T(?x, owl:allValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, rdf:type, ?x, ?co) Q(?u, ?p, ?v, ?co)	T(?v, rdf:type, ?y)

5.2 Reasoning with OWL^C using Spin

SPIN[6] or in other terms SPARQL rules can run directly on RDF data. It can be used to encapsulate reusable SPARQL queries as templates. One advantage of the templates is that they are flexible enough that you can simply pass parameters to them to customize their behavior. Then, they can be instantiated in any RDF or OWL ontology to add inference rules!

Using TopBraid Composer[7], we converted the OWL^C rules, presented in section 4, into SPIN templates available at the following link http://cui.unige.ch/isi/owl-rlc. The structure of the templates is based on the previously defined patterns in section 5.1. In order to perform contextual reaso-

---

[6] http://spinrdf.org

[7] https://www.topquadrant.com/tools/ide-topbraid-composer-maestro-edition/

Table 3: Entailment rules for the new contexts-based forming operators.

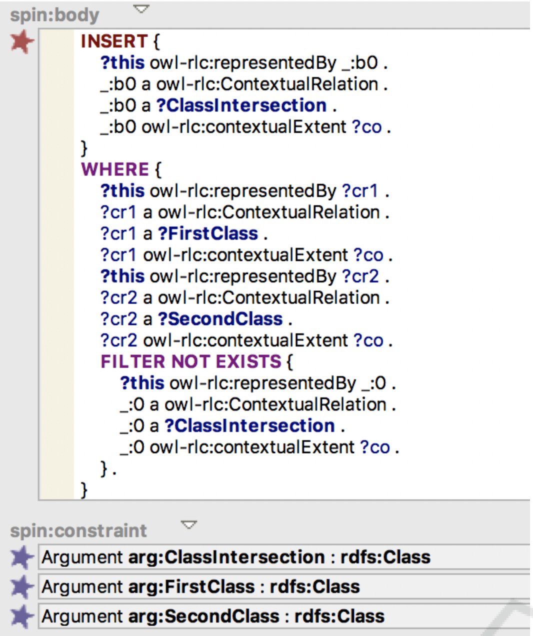
	IF	**THEN**
cxt-svf $(\langle C\rangle D)$	T(?e, owl^c : onClass, ?d) T(?e, owl^c : inSomeContextOf, ?co) Q(?x, rdf:type, ?d, ?y) T(?y, rdf:type, ?co)	T(?x, rdf:type, ?e)
cxt-avf $([C]D)$	T(?e, owl^c : onClass, ?d) T(?e, owl^c : inAllContextOf, ?co) T(?x, rdf:type, e) Q(?x, rdf:type, ?d, ?y)	T(?y, rdf:type, ?co)
cxt-svf $(\langle C\rangle P)$	T(?e, owl:onProperty, ?p) T(?e, owl^c:inSomeContextOf, ?co) Q(?x1, ?p, ?x2, ?y) T(?y, rdf:type, ?co)	T(?p, rdf:type, ?e)
cxt-avf $([C]P)$	T(?e, owl:onproperty, ?p) T(?e, owl^c: inAllContextOf, ?co) T(?p, rdf:type, ?e) Q(?x1, ?p, ?x2, ?y)	T(?y, rdf:type, ?co)

Table 4: Samples of the mapping of the core language to RDF.

Element E of the Structural Specification	**Triples Generated in an Invocation of T(E)**
ClassAssertion(CE[co] a)	T(a) owl^c:representedBy _:x. _:x rdf:type T(CE) . _:x rdf:type owl^c:contextualRelation. _:x owl^c:contextualExtent T(co).
ObjectPropertyAssertion(OP[co] a1 a2)	T(a1) T(OP) _:x. _:x rdf:type owl^c:contextualRelation . _:x T(OP) T(a2). _:x owl^c:contextualExtent T(co).

Table 5: Mapping the context language to RDF.

Element E of the Structural Specification	**Triples Generated in an Invocation of T(E)**
SomeConceptValuesFromContext(CE[co])	_:x rdf:type owl^c:ContextRestriction . _:x owl^c:onClass T(CE) . _:x owl^c:inSomeContextOf T(co) .
AllConceptValuesFromContext(CE[co])	_:x rdf:type owl^c:ContextRestriction . _:x owl^c:onClass T(CE) . _:x owl^c:inAllContextOf T(co) .
SomePropertyValuesFromContext(OPE[co])	_:x rdf:type owl^c:ContextRestriction . _:x owl:onProperty T(OPE) . _:x owl^c:inSomeContextOf T(co) .
AllPropertyValuesFromContext(OPE[co])	_:x rdf:type owl^c:ContextRestriction . _:x owl:onProperty T(OPE) . _:x owl^c:inAllContextOf T(co) .



```
INSERT {
   ?this owl-rlc:representedBy _:b0 .
   _:b0 a owl-rlc:ContextualRelation .
   _:b0 a ?ClassIntersection .
   _:b0 owl-rlc:contextualExtent ?co .
}
WHERE {
   ?this owl-rlc:representedBy ?cr1 .
   ?cr1 a owl-rlc:ContextualRelation .
   ?cr1 a ?FirstClass .
   ?cr1 owl-rlc:contextualExtent ?co .
   ?this owl-rlc:representedBy ?cr2 .
   ?cr2 a owl-rlc:ContextualRelation .
   ?cr2 a ?SecondClass .
   ?cr2 owl-rlc:contextualExtent ?co .
   FILTER NOT EXISTS {
      ?this owl-rlc:representedBy _:0 .
      _:0 a owl-rlc:ContextualRelation .
      _:0 a ?ClassIntersection .
      _:0 owl-rlc:contextualExtent ?co .
   } .
}
```

Argument **arg:ClassIntersection : rdfs:Class**
Argument **arg:FirstClass : rdfs:Class**
Argument **arg:SecondClass : rdfs:Class**

Figure 3: Template of the cls-int rule.

ning with OWL^C, the user has to:
(1) create his triples following the syntax described in section 5.1.
(2) instantiate the rules under the *contextualEntity* form[8].

Figure 3 shows the example of the *cls-int* rule encapsulated as a SPIN template. This template, similarly to all others, is implemented using a SPARQL INSERT request. It declares that the assertion of the same individual in two classes, holding for the same context, generates an assertion for this individual in the intersection of those classes, but also for the same holding contexts. Notice two things: first, the classes are declared as *spin:constraint* and second, the query contains a filter. The existence of the filter is of a major importance because it guarantees that an existing triple is not generated again and again, whenever the rules are running.

6 RELATED WORKS

Related works can be divided in two groups: theoretical and practical. In the theoretical group, in 2001, (Ghidini and Giunchiglia, 2001) introduced the idea of locality and compatibility where reasoning is considered mainly local and compatibility is argued to be used among the reasoning performed in different contexts. In 2003, (Borgida and Serafini, 2003) introduced the concept of distributed description logics. An advantage of DDLs is its support for multiple ontologies. However, the coordination between a pair of ontologies can only happen with the use of bridge rules. In 2004, a new concept called E-connections (Kutz et al., 2004) emerged: ontologies are interconnected by defining new links between individuals belonging to distinct ontologies. One major disadvantage is that it does not allow concepts to be subsumed by concepts of another ontology, which limits the expressiveness of the language. Then, in 2006, (Benslimane et al., 2006) attempted to extend description logics with new constructs with relative success. In 2011, a proposition was argued to use a two dimensional- description logics. Results showed that this approach does not necessarily increase the computational complexity of reasoning. In 2012, (Bozzato et al., 2012) argues that treating contexts in the semantic web needs more advanced means, such that contexts should be explicitly presented and logically treated...

In the practical group, many attempts to find a solution to the syntactic restriction of RDF binary relations emerged. Two types of works were proposed:

(a) Extending the data model or the semantic of RDF (Dividino et al., 2009) (Hartig and Thompson, 2014) (Nguyen et al., 2014)

(b) Using design patterns: It could be categorized along three axis (similarly to (Hayes, 2004)): 3D, 3D+1, 4D.
   - 3D representation: the contextual index co is attached to the sentence R(a,b) and thus R(a,b) holds for co such as RDF reification (Berners-Lee et al., 2001).
   - 3D+1 representation: the contextual index co is attached to the relation R(a,b,co) (Gangemi and Mika, 2003) (Aljalbout and Falquet, 2017).
   - 4D representation: the contextual index co is attached to the object terms R(a@co, b@co) where co is the contextual-slice of the thing named (Welty, 2010) (Welty et al., 2006)

7 CONCLUSION

In this paper, we intended to push forward the task of contextual reasoning in the semantic web. Therefore, we highlighted in section 2 the requirements that a contextual reasoning formalism must have, in order to best serve its purpose. To comply with those requirements, we proposed OWL 2 DL^C, an extension of OWL 2 DL. The particularity of this web on-

---

[8] You can find all the rules under spin:modules/spin:templates

tology language is that it consists of two dimensions: a context-dependent object dimension and a context dimension. Additionally, we defined a profile for applications that require scalable reasoning that we call OWL^C. It contains new context-dependent rules and novel rules for handling the new contextual constructs. The model does not increase the complexity of reasoning making it conform to the requirements. A practical implementation of the model was provided in section 5 using spin rules and an extension of the fluent pattern we introduced in a previous work (Aljalbout and Falquet, 2017). In the future works, we tend to update OWL^C by considering the semantic relations that could exist between the contexts.

REFERENCES

Aljalbout, S. and Falquet, G. (2017). Un modele pour la representation des connaissances temporelles dans les documents historiques. *arXiv preprint arXiv:1707.08000.*

Aljalbout, S. and Falquet, G. (2018). A semantic model for historical manuscripts. *arXiv preprint arXiv:1802.00295.*

Benslimane, D., Arara, A., Falquet, G., Maamar, Z., Thiran, P., and Gargouri, F. (2006). Contextual ontologies. *Advances in Information Systems*, pages 168–176.

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific american*, 284(5):28–37.

Borgida, A. and Serafini, L. (2003). Distributed description logics: Assimilating information from peer sources. *J. Data Semantics*, 1:153–184.

Bozzato, L., Homola, M., and Serafini, L. (2012). Context on the semantic web: Why and how. *ARCOE-12*, page 11.

Dividino, R., Sizov, S., Staab, S., and Schueler, B. (2009). Querying for provenance, trust, uncertainty and other meta knowledge in rdf. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):204–219.

Gangemi, A. and Mika, P. (2003). Understanding the semantic web through descriptions and situations. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 689–706. Springer.

Ghidini, C. and Giunchiglia, F. (2001). Local models semantics, or contextual reasoning= locality+ compatibility. *Artificial intelligence*, 127(2):221–259.

Hartig, O. and Thompson, B. (2014). Foundations of an alternative approach to reification in rdf. *arXiv preprint arXiv:1406.3399.*

Hayes, P. (2004). Formal unifying standards for the representation of spatiotemporal knowledge a report on arlada task 02ta4-sp1-rt1" formalisms for spatiotemporal reasoning" advanced decision architectures alliance.

Kazakov, Y. (2008). Riq and sroiq are harder than shoiq. In *KR 2008*. AAAI Press.

Klarman, S. and Gutiérrez-Basulto, V. (2011). Two-dimensional description logics for context-based semantic interoperability. In *AAAI*.

Kutz, O., Lutz, C., Wolter, F., and Zakharyaschev, M. (2004). E-connections of abstract description systems. *Artificial intelligence*, 156(1):1–73.

LaPorte, J. (2006). Rigid designators.

McCarthy, J. (1987). Generality in artificial intelligence. *Communications of the ACM*, 30(12):1030–1035.

Nguyen, V., Bodenreider, O., and Sheth, A. (2014). Don't like rdf reification?: making statements about statements using singleton property. In *Proceedings of the 23rd international conference on World wide web*, pages 759–770. ACM.

Noy, N., Rector, A., Hayes, P., and Welty, C. (2006). Defining n-ary relations on the semantic web. *W3C working group note*, 12(4).

Welty, C. (2010). Context slices: representing contexts in owl. In *Proceedings of the 2nd International Conference on Ontology Patterns-Volume 671*, pages 59–60. CEUR-WS. org.

Welty, C., Fikes, R., and Makarios, S. (2006). A reusable ontology for fluents in owl. In *FOIS*, volume 150, pages 226–236.