Automated Semantic Enrichment of Ontologies in the Construction Domain

Cheikh Kacfah Emani, cheikh.kacfah@cstb.fr
Centre Scientifique et Technique du Bâtiment (CSTB), Sophia Antipolis, France & Université Lyon 1, Lyon, France

Catarina Ferreira Da Silva, catarina.ferreira@univ-lyon1.fr
Université Lyon 1, Lyon, France

Bruno Fiès, bruno.fies@cstb.fr
Centre Scientifique et Technique du Bâtiment (CSTB), Sophia Antipolis, France

Parisa Ghodous, parisa.ghodous@univ-lyon1.fr
Université Lyon 1, Lyon, France

Marc Bourdeau, marc.bourdeau@cstb.fr
Centre Scientifique et Technique du Bâtiment (CSTB), Sophia Antipolis, France

Abstract
Technologies built by the Semantic Web (SW) community are more and more adopted by experts in the Construction domain. Indeed, SW languages like Resource Description Framework (RDF) and the Web Language Ontology (OWL) have been used for various purposes like information modeling, automatic rules' checking. Their massive adoption is due mainly to their ability to capture deep semantics, their standardization and their large adoption. At the base of semantic applications, we find ontologies. Their construction is time consuming and involves domain experts and SW engineers. It is thus interesting for experts to have tools which could help them enrich automatically ontologies from textual resources. In this paper, we bridge this gap through an approach which aims to provide automatically an OWL expression of a given concept from its natural language definition. This formalization process uses foremost Industry Foundation Classes entities. A first evaluation of this approach gives promising results.

Keywords: Automatic formalization, ontology enrichment, Industry Foundation Classes (IFC)

1 Introduction

1.1 Motivations

More and more, researchers in building engineering Construction, take advantage of Semantic Web (SW) languages. Expressiveness, standardization, a large and very active community are the main strengths of these languages. In Construction, SW languages are used for expressing new concepts (Farias, et al., 2014), for rule checking purposes like in (Pauwels, et al., 2011), (Yurchyshyna & Zarli, 2009) and (Zhong, et al., 2012). They have proven their efficiency even for technical document authoring (Bouzidi, et al., 2011). All these approaches have in common the use of an existing ontology. In some cases (for instance (Pauwels, et al., 2011), (Yurchyshyna & Zarli, 2009)) this ontology needs to be enriched with new concepts. Sometimes it may be necessary to build it from scratch (Bouzidi, et al., 2011). But build or enrich manually an ontology is tedious. This is the gap we want to bridge in this work. We propose an approach to rewrite automatically a definition as an OWL DL (Web Ontology Language – Description Logics) expression. For instance, we transform a

definition in natural language like *"An aluminium door is any door made of aluminium"*[1] into the formal expression:

new:AluminiumDoor ≡ IfcDoor **and** IsTypedBy **some** (RelatingType **some** (ConstructionType **value** ALUMINIUM)) (1)

In this formula, `IfcDoor` stands for the entity IfcDoor in Industry Foundation Classes (IFC) vocabulary. Likewise `isTypedBy`, `RelatingType` and `ConstructionType` are relationships in IFC. `ALUMINIUM` is an instance of `IfcDoorStyleConstructionEnum` which both represent entities in IFC with exactly the same name. The prefix '`new:`' before the concept `AluminiumDoor` indicates an entity newly created because non-existing in IFC. Finally, words in bold (and, some and value) are reserved OWL DL keywords.

1.2 Contributions

The approach we propose has the main advantage of being fully automatic. Moreover the formalization process is done w.r.t to an existing ontology or vocabulary and has a high level expressiveness (the one of OWL DL). In the current work, we consider the IfcOWL ontology recently proposed by (Terkaj & Pauwels, 2014). The formal expression we provide uses as much as possible the conceptual schema and the entities found in IfcOWL and thus in the official IFC specification. In addition, when there is not a suitable entity in IfcOWL to represent a phrase of the input definition, we propose a new concept to formalize this phrase. The result of this formalization process can thus be added directly to IfcOWL. To the best of our knowledge it is the first fully automatic approach for the formalization of natural language definitions applied in the field of Construction. A significant impact of being able to automatically generate the formal expression of new concepts is to better cover the terminology used in current construction rules. Indeed, existing rules in this domain have not being written to match IFC or any formal vocabulary. It is thus very common to find terms in rules which do not appear in any formal resource.

2 Semantic Web Technologies

One of the main trends in the current Web is the Web of data also known as the Semantic Web (SW). The main idea behind SW is to allow machines to *understand* data they actually process. To achieve this goal, one needs to describe data and to provide formal and processable expressions of entities. It is mainly done by means of ontologies.

Languages developed for the SW are based on both the standards Resource Description Framework (RDF) and RDF Schema (RDFS) (W3C, 2004). RDF is a language suitable to predicate something on a given subject. For sake of expressiveness, we found multiple tiers on top of RDF/S which include Web Ontology Language (OWL) (W3C, 2004b). OWL has a variant which is based on Description Logics (DL) formalisms, meaning that a concept is defined through a set of necessary and sufficient conditions. Therefore, an ontology comprises a terminological model (TBox) and an assertional model (ABox). The TBox contains the formal definitions of the concepts relevant for the domain and the Abox is made of instances of these concepts. The combination of terminological and assertional boxes results in a so-called knowledge base. For instance the concept `AluminiumDoor` formally expressed above, and the concept `IfcDoor` are members of the TBox and every concrete aluminium doors and doors are found in the ABox.

The main benefits of this representation are the following:

- *Automatic classification of qualified entities.* A key characteristic in the SW vision is to be able to infer new information. For instance, if in a knowledge base there is an instance `X` of the concept `IfcDoor` which has an `IfcDoorStyle` with a `ConstructionType` "`ALUMINIUM`", a *reasoner* will *automatically* deduce that `X` is an `AluminiumDoor` by using formula (1). Such classification can also occur at the TBox level. For instance by means of formula (1) the new concept `AluminiumDoor` will be seen as a sub-class of `IfcDoor`.

[1] All the definitions provided in this article are inspired from the dictionary (RSMeans, 2010).

- *Reusability*. Being able to reuse existing elements of schemas is a main concern in SW. Hence, if the formula (1) is added to an ontology, every additional definition which mentions the phrase "*aluminium door*" can be formalized using the new concept `AlulminiumDoor`. An actual example of this case is the definition "*An aluminium aperture is an opening reserved for an aluminium door or window.*"
- *Query writing simplification.* The standard language to query an RDF knowledge base in SW is SPARQL Protocol and RDF Query Language (SPARQL) (W3C, 2008). For example, if we want to retrieve all the aluminium doors we will actually use the SPARQL query[2] towards IfcOWL

```
SELECT ?door WHERE {?door rdf:type ifc:IfcDoor.
    ?door ifc:IsDefinedBy ?ifcRelDefByType.
  ?RelDefByType ifc:RelatingType ?doorStyle.
?doorStyle ifc:ConstructionType ifc:ALUMINIUM}
```

But if the concept `AluminiumDoor` is already defined, this query can be re-written more simply as :

```
SELECT ?door WHERE {?door rdf:type new:AluminiumDoor}
```

3 Automatic Formalization of Natural Language Definitions

To automatically provide the formal expression of each definition we sequentially follow these three steps:

i. The cutting of the definition into a set of atomic assertions which all follow the pattern <subject, verb, object>
ii. The matching of each part of each atomic assertion to IfcOWL
iii. The merging of all the atomic assertions to get a single expression

To illustrate these steps we will use the running example of the introduction *"An aluminium door is any door made of aluminium"*.

3.1 Atomic Assertions Identification

A definition in natural language is usually given through a complex sentence. By complex, we mean that many pieces of information are provided simultaneously. In this task, we aim to highlight all these pieces which will stand as a partition of the given definition. This task is also known as Open Information Extraction (OIE). In this work, each chunk has to respect the following constraints:

- A chunk must have three distinct parts: a *subject*, a *verb* to predicate something on the subject by means of an *object*. We hence call a chunk in an interchangeable manner a *triple*.
- A triple must be atomic. In other terms, we must not be able to extract another triple from a given triple.
- The set of chunks must cover all the pieces of information delivered by the original definition.

When we take the definition provided in our running example we have two triples: $triple_1$ = (*An aluminium door, is, any door*) and $triple_2$ = (*any door, is made of, aluminium*).
In practice, to realize this information extraction step we have used the tool named CSD-IE (Open Information Extraction via Contextual Sentence Decomposition) presented in (Bast & Haussmann, 2013).

3.2 Matching to IfcOWL

IfcOWL, the OWL DL version of standard IFC reuses as much as possible entities, relationships, data types, general organization of IFC 4. Through the alignment of natural languages phrases to IfcOWL, we guarantee the reuse of an existing standard to formalize the definitions.

[2] The prefix **rdf** stands for <http://www.w3.org/1999/02/22-rdf-syntax-ns#> and the prefix **ifc** for <http://buildingsmart.org/ontology/IFC#> used in the IfcOWL ontology (Terkaj & Pauwels, 2014).

We perform this matching task for each triple. Prosaically, we have to identify respectively the IfcOWL entities which the subject, the object and the predicate of the triple refer to.

3.2.1 Matching of the Subject and the Object of a Triple

Using the subject of the triple, we compute the string matching score between each concept and individual of IfcOWL. The entity with the highest matching score is considered as the entity the subject refers to. When this matching score is under a given threshold, we decide to create a new entity. We repeat this process to assign an entity to the object of the triple.
To illustrate this sub task, let us consider the two triples of the preceding steps, we obtain the following connections:

- triple$_1$ (*An aluminium door, is, any door*)
  - An aluminium door (subject) ⟶ `new:AluminiumDoor`
  - Any door (object) ⟶ `IfcDoor`
- triple$_2$ (*Any door, is made of, aluminium*)
  - Any door (subject) ⟶ `IfcDoor`
  - Aluminium (object) ⟶ `ALUMINIUM`

3.2.2 Formal Relation between the Subject and Object Entities

Now we have identified the two entities linked within each triple, we have to provide the formal link between them. For this relation identification, we face two main cases. This relation can be immediate of based on properties of IfcOWL.

i. In the first case the relation can be either a traditional subsumption also known as 'is a' ($\subseteq$) or the equivalence ($\equiv$). Here *subsumption* and *equivalence* are hinted by a set of precise key phrases found within the triple. For instance, 'is a', 'is an', 'are a' indicate a subsumption. Phrases like 'is any', 'is all', 'are every' are hints for an equivalence.

ii. In the second case the relation is built on top of one or more properties found in the ontology. To identify these properties, we answer the following question: 'In the RDF graph denoted by the ontology, which are the *possible paths* of *length* n between the two identified entities'? A path of length n is one involving n properties (i.e. n steps to move from the subject entity to the object one). In practice we first try to answer this question for $n = 1$. The set of retrieved properties is ranked using the matching score of their labels with the predicate of the triple. If and only if we have no candidate properties, we increment n to 2 and so on. We stop this search at $n = 5$. Beyond this threshold, we simply conclude to an error in the formalization process. For a fixed value of n, this question corresponds to a SPARQL query. For example, with $n = 1$, the appropriate SPARQL query is[3]:

```
SELECT ?p WHERE
{subjectEntity rdfs:subClassOf* ?Dp . ?p rdfs:domain ?Dp.
 objectEntity rdfs:subClassOf* ?Rp . ?p rdfs:range ?Rp}
```

When at least one of the subject and the object are new entities, i.e. not present in IfcOWL, we do not make use of this task. We decide on the creation of a new relationship new:IsRelatedTo, to express the connection between the subject and the object.

Taking our running example, we obtain the following results:

- triple$_1$ (*An aluminium door, is, any door*)
  - "is any" ⟶ equivalence ($\equiv$)
  - Formal(triple$_1$) = `new:AluminiumDoor` $\equiv$ `IfcDoor`
- triple$_2$ (*Any door, is made of, aluminium*)
  - Identified entities: `IfcDoor` (subject) and `ALUMINIUM` (object)
  - The path between the two entities is clarified by Figure 1. To identify this path, it is necessary to take advantage of the hierarchy (and thus the inheritance mechanisms)

[3] The prefix **rdfs** corresponds to <http://www.w3.org/2000/01/rdf-schema#>

of the entities involved. In this case, it is necessary to make use of three properties to link `IfcDoor` and `ALUMINIUM`.

- Formal(triple$_2$) = `IfcDoor` **and** `IsTypedBy` **some** (`RelatingType` **some** (`ConstructionType` **value** `ALUMINIUM`))

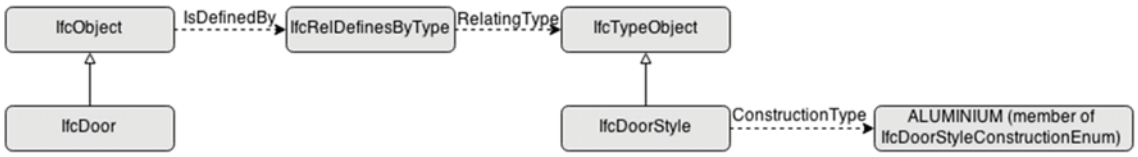

Figure 1 Path between the entities IfcDoor and ALUMINIUM

3.3 Merging the Formal Triples

Now we have formalized all the triples, we need to put them together to obtain a single expression. We perform this operation by using entities which appear in many formal expressions as connection nodes between the triples[4]. In our illustration case, the junction is realized through the concept `IfcDoor`. Indeed, it appears as the object in triple$_1$ and acts as the subject in triple$_2$. In the second triple, `IfcDoor` is constraint to be made of ALUMINIUM. This feature will be included in triple$_1$ by means of the conjunction ***and*** to obtain

new:AluminiumDoor ≡ IfcDoor **and** IsTypedBy **some** (RelatingType **some** (ConstructionType **value** ALUMINIUM))

4 Preliminary Evaluation and Discussion

4.1 Evaluation

From the RSMeans dictionary (RSMeans, 2010), we have built a corpus made of $N = 25$ definitions for a preliminary evaluation of this approach. The RSMeans dictionary is an illustrated source for construction terms and concepts. It is regularly updated and is used by numbers of professional.

For each sentence S_i we have:

- the set W_i (W for wish) of entities (new or present in IfcOWL) we wish to identify using our approach
- the set I_i (I for identified) of entities identified after the automatic formalization process
- its formal expressions F_i (F for formal) resulting from the process

For the N sentences, we compute the three following traditional quantities to quantify our evaluation process:

$$Precision = \frac{\sum_{i=1}^{N} |I_i \cap W_i|}{|I_i|}, \; Recall = \frac{\sum_{i=1}^{N} |I_i \cap W_i|}{|W_i|} \text{ and } F1\ measure^5 = \frac{2 \times precision \times recall}{precision + recall}$$

We measure the quality of the formal expressions by $\frac{|\{correct\ F_i,\ 1 \le i \le N\}|}{N}$

For these 25 sentences, we have the following results:

- Precision: 0.67
- Recall: 0.67
- F1 measure: 0.67
- Quality of formalization: 72% (18/25)

[4] Detailing the full algorithm to achieve this task is out of the scope of this paper.

[5] The F1 measure is the *harmonic mean* of the precision and recall

4.2 Discussion

- Even if the process for identifying entities performs quite well, some improvements need to be done. Currently we do not use semantic relations like synonymy to link natural language phrases to formal entities. Moreover we must be able to handle noise during this process. Indeed, some words appear in the original definitions but are useless for this task. For example the phrase "is made of" mentioned in our running example is not useful to identify the formal path between the entities `IfcDoor` and `ALUMINIUM`.
- We have seen that under certain conditions, we propose new entities to formalize the definition. Actually this process is out of the control of any user including experts. Thus, we need to provide an interactive tool based on this approach where the experts in Construction, would be able to easily validate or edit the result of the formalization.
- Actually this approach has been designed to formalize natural language definitions. It would be interesting to see how it can be adapted to handle rules.

5 Conclusion and future work

In this paper, we propose an approach to automatically convert a definition in natural language into a formal OWL DL expression. This expression is foremost built on top of entities already existing in the IFC vocabulary. Moreover it is able to match the conceptual schema of IFC. Some recent work has shown the interest of providing formal definitions of concepts using Semantic Web languages (Farias, et al., 2014). But we are the first, to the best of our knowledge, to provide a fully automatic solution for this problem. Actually our approach does not involve expert control which is essential for the integration of results in an IFC ontology like IfcOWL. We thus need to provide a plugin or an extension to be integrated in classical ontology manipulation environments (like Protégé[6]). Another main concern in the field of Construction is rules management especially the automatic control of the conformity of building engineering products. A prerequisite of such process is the formalization of rules which are mainly available as natural language text resources. Through our approach we are able to integrate into existing formal resources, new terms found in rules and we are able to envision the formalization of the whole rule itself. It is thus interesting to see how our approach can help to achieve this goal.

References

Bast, H. & Haussmann, E., 2013. *Open Information Extraction via Contextual Sentence Decomposition.* s.l., IEEE, pp. 154-159.

Bouzidi, K. R. et al., 2011. *An ontology for modelling and supporting the process of authoring technical assessments.* Sophia Antipolis, s.n., pp. 1-9.

Farias, M. T., Roxin, A. & Nicolle, C., 2014. *A Rule Based System for Semantical Enrichment of Building Information Exchange.* Prague, s.n., pp. 2-9.

Pauwels, P. et al., 2011. A semantic rule checking environment for building performance checking. *Automation in Construction,* 20(5), pp. 506-5018.

RSMeans, 2010. *RSMeans Illustrated Construction Dictionary.* 4e éd. s.l.:Wiley.

Terkaj, W. & Pauwels, P., 2014. *IfcOWL ontology file for IFC4.* [En ligne] Available at: http://linkedbuildingdata.net/resources/IFC4_ADD1.owl

W3C, 2004b. *OWL Web Ontology Language.* [En ligne] Available at: http://www.w3.org/TR/owl-features/

W3C, 2004. *RDF Vocabulary Description Language 1.0: RDF Schema.* [En ligne] Available at: http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

W3C, 2008. *SPARQL Query Language for RDF.* [En ligne] Available at: http://www.w3.org/TR/rdf-sparql-query/

Yurchyshyna, A. & Zarli, A., 2009. An ontology-based approach for formalisation and semantic organisation of conformance requirements in construction. *Automation in Construction,* 18(8), pp. 1084-1098.

Zhong, B. et al., 2012. Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking. *Automation in Construction,* Volume 28, pp. 58-70.

[6] http://protege.stanford.edu/