Easing the Definition of N–Ary Relations for Supporting Spatio–Temporal Models in OWL

Alberto G. Salguero, Cecilia Delgado, and Francisco Araque

Dpt. of Computer Languages and Systems
University of Granada, Granada (Andalucía), Spain
{agsh,cdelgado,faraque}@ugr.es

Abstract. There are many issues to overcome when integrating different data sources due to the number of variables that are involved in the integration phase. However we are interested in the integration of temporal and spatial information due to the nature of modern Information Systems. We have previously developed a model, called STOWL, which is a spatio-temporal extension of OWL. We use this model as the common model for defining the schemes of the data sources in order to ease their integration. This paper presents the part of STOWL which has to do with the definition of n-ary relations.

Keywords: Ontology, data sources, data integration, OWL.

1 Introduction

OWL is the language adopted by the W3C for defining ontologies and supporting the Semantic Web. A growing number of Semantic Web applications, which can interact between them and cooperate to find better solutions, is being developed [2], [5], [6]. It is a common belief that Semantic Web technology would significantly impact the use of the Web, essentially in terms of increased task delegation to intelligent software agents, and a subsequent amelioration of the information overload effect [3].

This work focuses on the integration of spatial information. This kind of information has grown in importance in recent years due to the proliferation of GIS-based applications and the global positioning system (GPS). Increasingly, companies rely on such information to develop their business or enhance their productivity. The work is restricted to spatio-temporal information for reducing complexity in terms of integration possibilities: it is easier to perform the integration if the working environment is considerably reduced. It would be impractical to propose a model which considers all aspects of the real world. Moreover, we believe that this type of information is general enough to solve most of the problems that we can find nowadays. It could be possible to develop a financial-oriented model but the number of supported situations would be fewer than using a more general model. The problem is that OWL does not support some of the desired characteristics of a spatio-temporal data model.

One of these missing characteristics is the possibility of defining n-ary relations between concepts. It is common when dealing with geographic information which evolves in time to mark the instant when a measurement has been made, but using

R. Moreno-Díaz et al. (Eds.): EUROCAST 2009, LNCS 5717, pp. 271–278, 2009.

OWL only binary relations can be defined. Although the W3C on its own has described how to represent n-ary relations using OWL the resulting code is long and not easily understandable by a person.

In this paper an extension of OWL is presented in order to support the definition of n-ary relations in the context of geographic information systems. We call this extension STOWL [1].

The remaining part of this paper is organized as follows. In the following section the differences between OWL and STOWL are illustrated; in section 3 our model is presented; finally, section 4 summarizes the conclusions of this paper.

2 Differences between OWL and STOWL Ontologies

STOWL is the name of the common data model which is proposed in this work for describing the schemes of the different data sources. In this point, the differences between OWL and STOWL are explained.

The use of a data model based on ontologies is proposed as a common data model to deal with the data sources schemes integration. Although it is not the first time the ontology model has been proposed for this purpose [5], [6], [7], [8], to our knowledge, this is the first time an ontology language has been used to improve the Information System data refreshment process design.

Both languages allow the description of ontologies based on the OWL ontology. The OWL language can be used for describing spatio-temporal data repositories. The main problem is that this language lacks some desired features which makes difficult to express certain type of knowledge which is common when dealing with spatial and temporal information:

- *Description of exhaustive decompositions*. As well as in the DAML+OIL ontology (the OWL ontology derive from), in the OWL ontology is possible to express exhaustive decompositions. For this is used the primitive *owl:oneOf*. In spite of that, it is important to note that the classes described using this primitive are exhaustive decompositions of instances of other classes. The problem is that, in certain cases, it is necessary to describe exhaustive decompositions of classes, not specific instances.
- *Description of partitions*. Unlike the DAM+OIL model, in OWL is not possible to express partitions of concepts. This is because, in the case it is needed, a partition can be expressed combining the primitives *owl:disjoint With* and *owl:unionOf*. Although valid, the code needed for describing partitions with this approach is long and complex to follow.
- *N-ary relations*. OWL only allows the definition of binary relations. In case of higher arity relations need to be defined, the W3C suggests the creation of artificial classes (or concepts) for representing those relations. As well as with the previously described properties, although it is possible to express n-ary relations using the OWL language the resultant code is difficult to follow. The STOWL language has been defined to solve this issue.
- *N-ary functions*. OWL has not been designed for supporting functions. In case of necessity, the primitive *owl:FunctionalProperty* can be used as a mechanism for defining binary relations. It is not possible to express functions in

OWL with higher arity. It is therefore not possible to define layered information, a common type of data managed by the geographical information systems (terrain height, terrain usage…).

- *Formal axioms*. None of the markup-based languages for describing ontologies, including OWL, support the definition of formal axioms.
- *Rules*. They are a special type of formal axioms, so they cannot be expressed in OWL. They can be used for inferring new knowledge. Geographic information system can use this rules for performing complex queries.
- *Integrity constraints*. The integrity constraints make the maintenance of semantic consistency of data easier. OWL only considers the one type of integrity constraint: the functional dependency, which can be expressed by mean of the primitive *owl:FunctionalProperty*. There are other types of integrity constraints which should be addressed: unique integrity constraint, assertions… The integrity constraints are related to the quality of data.

The rest of this work is focused in how the OWL language has been extended in order to consider n-ary relations.

3 Extending OWL with N-Ary Relations

STOWL is build on top of OWL language. Most of its features rely on OWL features On the other hand, there are spatio-temporal characteristics which can not be expressed in the OWL language because to the OWL ontology does not support those desired features. In this case the modification of the OWL ontology should be made in order to incorporate those features and the OWL language, consequently, has also to be modified in order to reflect those changes (giving STOWL as result).

In the former situation STOWL can be seen as a software layer that transform the new defined spatio-temporal features, which can be expressed straightforwardly in STOWL, in a more complex and equivalent definition in OWL. Due to this transformation the results continue being an ontology description expressed in OWL, so all the OWL tools can be used as usual (reasoners, editors...). This is the case of the proposed extension in this paper. N-ary relations in STOWL are transformed to concepts in OWL that represent those relations.

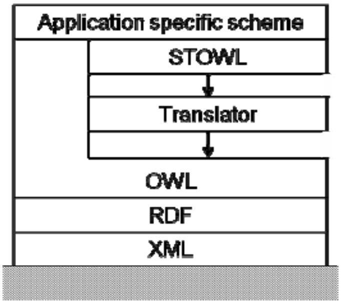

Fig. 1. STOWL functional diagram

3.1 Defining N-Ary Relations in OWL

N-ary relations cannot be expressed directly in OWL. The binary relations are the relations with the higher arity which can be defined using OWL. The W3C, being aware of this problem, has proposed some solutions in order to solve this issue. Their usage depends on the specific situation and the intended meaning for the n-ary relation [4]:

- Additional attributes describing relations.
- Different subjects about the same relation.
- N-ary relation with no distinguished participant.

In the former situation, for describing a relation is necessary to add an attribute to it. This is the case when, for instance, the definition of a relation for tracking the position of a person or any object is needed. In OWL is possible to specify that a person is located at some location but it is not possible to specify that the person were located at some location at some instant. For this usage pattern the W3C suggest the creation of a new artificial concept to represents the ternary relation (figure 2).

Although valid, this solution has some important drawbacks. One of those drawbacks is, for instance, the necessity of creating instances of the introduced artificial concepts representing the n-ary relations. Those instances are also artificial and do not provide useful knowledge from the specific problem point of view. Furthermore, as it can be seen in figure 3, its implementation could be relatively complex, even when the implicated concepts are relatively few (as in this case). From the developer point of view, its usage is difficult and error-prone. Much of the effort is dedicated to handling with the artificial concepts and instances and how they are related with the rest of the elements of the scheme (the really important elements).

The second of the usage patterns (different subjects about the same relation) is resolved in the same way as the former. This kind of relation appears when the relation to be represented has ranges of complex objects with multiple relations involved.

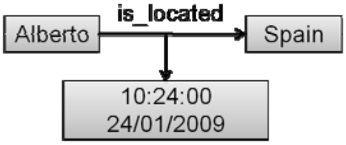

Fig. 2a. Example of ternary relation

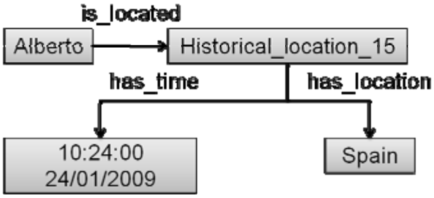

Fig. 2b. Example of artificial concept created for representing the ternary relation of figure 2a

On the other hand, the latter usage pattern (n-ary relation with no distinguished participant) is slightly different. In this case, there is no a main concept in the relation. In spite of this, the solution is equivalent to the former usage patterns: create an artificial concept which represents the n-ary relation.

```
<owl:Class rdf:ID="Person">
<rdfs:subClassOf>
            <owl:Restriction>
                        <owl:onProperty>
                                    <owl:ObjectProperty rdf:about="#is_located"/>
                        </owl:onProperty>
                        <owl:allValuesFrom>
                                    <owl:Class rdf:about="# Historical_location"/>
                        </owl:allValuesFrom>
            </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
…
<owl:Class rdf:ID="Historical_location">
<rdfs:subClassOf>
            <owl:Restriction>
                        <owl:someValuesFrom rdf:resource="#Location"/>
                        <owl:onProperty>
                                    <owl:FunctionalProperty rdf:about="#has_location"/>
                        </owl:onProperty>
            </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
            <owl:Restriction>
                        <owl:allValuesFrom rdf:resource="#TimePeriod"/>
                        <owl:onProperty>
                                    <owl:FunctionalProperty rdf:about="#has_time"/>
                        </owl:onProperty>
            </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

Fig. 3. Implementation of the example in figure 2b using OWL

There are some considerations which have to be taken into account when introducing a new concept as a relation:

- It is necessary to implement some kind of method for assigning *meaningful names* to instances of properties or to the artificial concepts used to represent instances of n-ary relations.
- Creating a class to represent an n-ary relation limits the use of many OWL constructs and creates a *maintenance problem.*
- Defining *inverse properties* with n-ary relations, using any of the patterns above, requires more work than with binary relations.

In a spatio-temporal environment it is very common to find such kind of relations and their management could results very difficult. It is very common, for instance, to find in any spatial-based IS a ternary relation such as “connect” (representing that two locations are connected by a segment). It would require the definition of a new concept representing the relation and two new properties relating that concept and two of the three related concepts. This operation can be performed when there are few n-ary relations and instances defined in the ontology but it is very difficult to maintain an ontology when the number of such kind of relations increases. This is precisely the case of geographical information. On the other hand, only one relation (and no new concepts) should be defined in order to represent the same situation with STOWL, capable of defining n-ary relations. As explained previously, STOWL is a software layer which tries to overcome the drawbacks stated before.

Following are detailed how OWL has been extended in order to consider n-ary relations.

3.2 STOWL Abstract Syntax

The first step consists on extends OWL in order to support n-ary relations. Actually, we consider that the best option for accomplishing this task is to directly modify RDFS, which OWL relies on. The changes introduced have to do with the possibility of defining multiple range values for the relations. They consisted in the definition of *n* subclasses of the class *rdfs:Range* (figure 4), being *n* the desired maximum arity. The changes are declared in a new namespace called RDFSN.

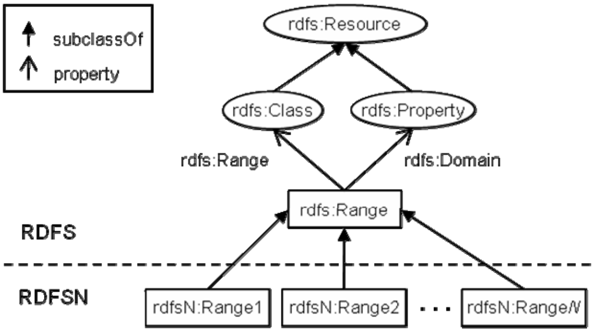

Fig. 4. Definition of RDFSN as an extension of RDFS for supporting properties with multiple ranges

The OWL abstract syntax does not have to be modified in order to consider multiple ranges. As illustrated in figure 5, the OWL abstract syntax is designed to allow multi-ranges properties.

```
axiom ::= 'DatatypeProperty(' datavaluedPropertyID ['Deprecated'] { annotation }
                { 'super(' datavaluedPropertyID ')' } ['Functional']
                { 'domain(' classID' ')' } { 'range(' dataRange ')' } ')'
        | 'ObjectProperty(' individualvaluedPropertyID ['Deprecated'] { annotation }
                { 'super(' individualvaluedPropertyID ')' }
                [ 'inverseOf(' individualvaluedPropertyID ')' ] [ 'Symmetric' ]
                [ 'Functional' | 'InverseFunctional' | 'Functional' 'InverseFunctional'
                  | 'Transitive' ]
                { 'domain(' classID ')' } { 'range(' classID ')' } ')'
```

Fig. 5. OWL abstract syntax relating to relations range description

3.3 STOWL RDF Syntax

Although the OWL abstract syntax allows the definition of properties with multiple ranges, the RDF model does not allow such kind of knowledge to be expressed. This is basically the reason why the subclasses of the concept *rdfs:Range* have been introduced in the previous section. When needed, they can be used to relate more than two classes in the same relation. The solution consists on replacing the property *rdfsn:Range* by as many as *rdfsn:RangeX* properties as concepts involved in the relation. The property *is_located* of figure 2, for instance, is defined in STOWL using two derived classes of *rdfs:Range*, as illustrated in figure 6.

```
<owl:ObjectProperty rdf:ID="is_located">
        <rdfs:domain rdf:resource="#Person"/>
        <rdfsn:range1 rdf:resource="#Location"/>
        <rdfsn:range2 rdf:resource="#Time"/>
</owl:ObjectProperty >
…
```

Fig. 6. Definition of property *is_located* in STOWL using n-ary relations

When defining restrictions about the n-ary relations is necessary to specify what of the concepts are involved. This can be performed, as illustrated in figure 7, adding the *rdfsn:RangeX* attribute to the property description to indicate the affected concept.

```
<owl:Class rdf:ID="Person">
<rdfs:subClassOf>
        <owl:Restriction>
                <owl:onProperty>
                        <owl:ObjectProperty rdf:about="#is_located" rdfs:range="#range1"/>
                </owl:onProperty>
                <owl:allValuesFrom>
                        <owl:Class rdf:about="#Location"/>
                </owl:allValuesFrom>
        </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

Fig. 7. Example of definition of property restrictions on n-ary relations in STOWL

As result it is obtained the possibility of defining n-ary relations straightforwardly, without having to manually create artificial classes or instances. As well as with the case of the restrictions, the unique difference with respect to OWL when defining instances are that it is necessary to specify the range of the property value. Figure 8 illustrated how the example in figure 2 can be expressed easily in STOWL without having to create artificial classes and instances.

```
<Person rdf:ID="Alberto">
        <is_located rdf:resource="Spain" rdf:range="range1">
        <is_located rdf:resource="10:24:00 24/01/2009" rdf:range="range2">
</Person>
```

Fig. 8. Example of definition of restrictions about n-ary relations in STOWL

4 Conclusions

In this paper we have presented our work relating to the definition of n-ary relations in OWL. Furthermore, we have described the elements in STOWL, a spatio-temporal extension of OWL we have developed in order to ease the integration of different data sources schemes, which have to do with the possibility of defining n-ary relations.

We have justified why this kind of knowledge is important when dealing with spatio-temporal information and presented the difficulties for implementing it directly in OWL. Although the W3C provides some valid solutions to overcome this issue they require a big effort. STOWL is a software layer on top of OWL which eases the definition of n-ary relations.

Acknowledgements

This work has been supported by the Andalucía Research Program under project GR2007/07-2 and by the Spanish Research Program under projects EA-2007-0228 and TIN2005-09098-C05-03.

References

1. Salguero, A., Araque, F., Delgado, C.: Using ontology meta data for data warehousing. In: Filipe, J., Cordeiro, J. (eds.) ICEIS 2008. LNBIP, vol. 19, pp. 28–35. Springer, Heidelberg (2009)
2. Kolas, D., Dean, M., Hebeler, J.: Geospatial semantic Web: architecture of ontologies. In: Rodríguez, M.A., Cruz, I., Levashkin, S., Egenhofer, M.J. (eds.) GeoS 2005. LNCS, vol. 3799, pp. 183–194. Springer, Heidelberg (2005)
3. Lytras, M.D., García, R.: Semantic Web applications: a framework for industry and business exploitation – What is needed for the adoption of the Semantic Web from the market and industry. International Journal of Knowledge and Learning 4(1), 93–108 (2008)
4. Hayes, P., Welty, C.: Defining N-ary Relations on the Semantic Web. W3C Working Group Note April 12 (2006), `http://www.w3.org/TR/swbp-n-aryRelations`
5. Goudos, S.K., Peristeras, V., Tarabanis, K.A.: Semantic Web Application for Public Administration using OWL for Public Domain Data Knowledge Representation. WSEAS Transactions on Information Science & Applicattions 4(4), 725–730 (2007)
6. Yang, S.Y.: How Does Ontology help Web Information Management Processing. WSEAS Transactions on Computers 5(9), 1843–1850 (2006)
7. Skotas, D., Simitsis, A.: Ontology-Based Conceptual Design of ETL Processes for Both Structured and Semi-Structured Data. International Journal on Semantic Web and Information Systems 3(4), 1–24 (2006)
8. Ale, M.A., Gerarduzzi, C., Chiotti, O., Galli, M.R.: Organizational Knowledge Sources Integration through an Ontology-Based Approach: The Onto-DOM Architecture. In: Lytras, M.D., Carroll, J.M., Damiani, E., Tennyson, R.D. (eds.) WSKS 2008. LNCS (LNAI), vol. 5288, pp. 441–450. Springer, Heidelberg (2008)