Representing Contextual Information as Fluents

José M. Giménez-García[1(✉)], Antoine Zimmermann[2], and Pierre Maret[1]

[1] Univ Lyon, UJM-Saint-Etienne,
CNRS, Laboratoire Hubert-Curien UMR 5516, 42023 Saint-Étienne, France
jose.gimenez.garcia@univ-st-etienne.fr

[2] École Nationale Supérieure des Mines, FAYOL-ENSMSE,
Laboratoire Hubert Curien, 42023 Saint-Étienne, France

Abstract. Annotating semantic data with metadata is becoming more and more important to provide information about the statements. While there are solutions to represent temporal information about a statement, a general annotation framework which allows representing more contextual information is needed. In this paper, we extend the 4dFluents ontology by Welty and Fikes to any dimension of context.

1 Introduction

In the Semantic Web, it is often necessary to characterize the context associated to a statement, *e.g.*, when it was generated, or who said it. In RDF and OWL this can only be represented natively using binary relations. There are generic approaches for representing statements about statements, such as using reification[1], N-ary relations[2] or N-Quads[3], and the Singleton Property [4]. However, each of them has its own drawbacks. When using reification, inference is prevented; in the case of N-ary relations, the structure of the domain ontology needs to be changed (not always possible when reusing external ontologies). For N-Quads there are no formal semantics, and its usage for named graphs has been standardized. The Singleton Property on the other hand, requires to extend the formal semantics of RDF and, due to the explosion in the number of properties, has proved not to be efficient in current knowledge bases [3].

On the other hand, Welty and Fikes [6] propose a model to represent temporal validity of an entity by considering it a perdurant, without any of the previous disadvantages. This work has been used by the community and further extended by other authors to address the proliferation of slices [7], or representing spatio-temporal information [1]. While Welty [5] proposes a generalization of the model by substituting the temporal part of an entity by a contextual projection, he does not address the possibility of using different dimensions of context in the same dataset.

In this work, we propose an extension of Welty and Fikes model to a generic ontology that can be extended to implement any number of concrete dimensions

[1] http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/#reification.
[2] https://www.w3.org/TR/swbp-n-aryRelations.
[3] https://www.w3.org/TR/n-quads.

P. Ciancarini et al. (Eds.): EKAW 2016 Satellite Events, LNAI 10180, pp. 119–122, 2017.
DOI: 10.1007/978-3-319-58694-6_13

of context (Sect. 2). In addition, we address different issues and decisions to make when modeling a knowledge base using out approach (Sect. 3). Finally, we give some final remarks and possible lines of future work (Sect. 4).

2 NdFluents

Welty and Fikes [6] address the problem of representing *fluents*, *i.e.*, relations that hold only within a certain time. They address the issue by using the perdurantist view. According to it, entities are four dimensional constructs, and instead of making statements about them, one should make the assertions about their temporal parts. Instead of making an assertion about some entities, such as *"Paris is the capital of France"*, one should make the assertion about their temporal parts: *"A temporal part of Paris (since 508 up to now) is the capital of a temporal part of France (since 508 up to now)"*.

The temporal part of an entity can be viewed as an individual context dimension of the entity. A similar approach can then be used to represent different dimensions, such as provenance. Continuing with our running example, we can articulate that fact as *"Paris (according to Wikipedia) is the capital of France (according to Wikipedia)"*. Different context dimensions of an entity could then be combined, allowing to represent complex information: *"According to Wikipedia, Paris has been the capital of France since 508"*.

We use this idea to generalize the 4dFluents ontology for any context dimension in the *NdFluents* ontology [2]. The ontology, shown down below, is a direct extension from temporal parts to any contextual parts. While approaches that reify the predicate hinder OWL reasoning, NdFluents allows for OWL inference of OWL property axioms within the same contexts. The ontology, the extensions for temporal and provenance dimensions, and a use case where both dimensions are used (the estimated evolution of Earth population according to different sources) is published in http://www.emse.fr/~zimmermann/ndfluents.html

```
Declaration( Class( nd:Context ) )
Declaration( Class( nd:ContextualPart ) )
DisjointClasses( nd:Context nd:ContextualPart )
Declaration( ObjectProperty( nd:contextualProperty ) )
ObjectPropertyDomain( nd:contextualProperty nd:ContextualPart )
ObjectPropertyRange( nd:contextualProperty nd:ContextualPart )
Declaration( DataProperty( nd:contextualDatatypeProperty ) )
DataPropertyDomain ( nd:contextDataProperty nd:ContextualPart )
Declaration( ObjectProperty( nd:contextualExtent ) )
ObjectPropertyDomain( nd:contextualExtent nd:ContextualPart )
ObjectPropertyRange( nd:contextualExtent nd:Context )
Declaration( ObjectProperty( nd:contextualPartOf ) )
FunctionalObjectProperty( nd:contextualPartOf )
ObjectPropertyDomain( nd:contextualPartOf nd:ContextualPart )
ObjectPropertyRange( nd:contextualPartOf ObjectComplementOf( nd:Context ))
```

3 Modeling a Knowledge Base with NdFluents

In this section we present possible issues that may arise from the approach and what can be done to solve them when modeling data using the NdFluents ontology. While the first two points are common to Welty and Fikes [6] approach, the last ones are only relevant when using more than one dimension.

Adapting Non-Contextual Ontologies to NdFluents: When reusing existing ontologies, it is not always possible to add a subproperty relationship between an already defined property and a contextual property, because most restrictions will not hold for the contextual parts of an entity. For instance, let us suppose that the property `capitalOf` has domain `City`, and we use it as a fluent property. Then, it would be inferred that `Paris@1` is a city, instead of `Paris` being inferred as a city.

Dealing with Terminological Statements: In general, contexts are used just for assertions (the *ABox*). While using contexts with terminological statements (the *TBox*) is possible, it is important to take into account that new properties will not benefit from the standard inferences associated with `subClassOf`.

Relations between Different Contextual Parts: The NdFluents ontology allows to model relations among different contextual parts of different context dimensions (*i.e.*, a temporal part of Paris could be the capital of a provenance part of France). If it is needed for a contextual property to be related to contextual parts of the same dimension of context, it will be necessary to add the appropriate axioms to the ontology:

```
Declaration( ObjectProperty( 4d:fluentProperty ) )
SubObjectPropertyOf( 4d:fluentProperty nd:contextualProperty )
ObjectPropertyDomain( 4d:fluentProperty 4d:TemporalPart )
ObjectPropertyRange( 4d:fluentProperty 4d:TemporalPart )
Declaration( DataProperty( 4d:fluentDataTypeProperty ) )
SubDataPropertyOf( 4d:fluentDataTypeProperty nd:contextualProperty )
DataPropertyDomain( 4d:fluentProperty 4d:TemporalPart)
```

Combining Different Context Dimensions: An important scenario where NdFluents becomes relevant is when the necessity of combining two or more dimensions of context arises. There are different possibilities to model them in NdFluents.

- *Contexts in Context:* Relating a contextual part to another contextual part. This approach can be taken when the "first level" contextual parts are relevant facts of the knowledge base, and we want to state additional information about them. To improve reasoning, `contextualPartOf` property needs to be transitive, which can be achieved by adding the following axiom: `TransitiveObjectProperty( nd:contextualPartOf )`
- *Use Multiple Contexts for each Contextual Part:* Only one contextual part is created for a combination of context dimensions. This contextual part is then related to all the related contextual information. This model is easier to model: Relating the contextual part with the context dimensions is straightforward. It also avoids ambiguity when modeling contextual information related to more than one contextual dimension, and reduces the number of resources in the ontology (*i.e.*, while the previous model needed one contextual part for each context dimension involved, this approach only requires one contextual part).

Note that `contextualPartOf` is a functional property, which means that there cannot be a contextual part of more than one entity.
- *Relations between Different Contextual Parts:* Creating contextual extents that combine two or more context dimensions, and enforcing a limit of only one contextual extent per contextual part. This model adds a layer of complexity to the previous approach, but it can be useful to require a specific combination of context dimensions on a set of contextual parts. This can be achieved by adding the axiom `FunctionalObjectProperty( nd:contextualExtent )`

4 Conclusion

Representing contextual information in different dimensions is a current challenge in OWL. We have proposed NdFluents, a multi-domain contextual representation, generalizing the 4dFluents ontology to any number of context dimensions. NdFluents allows for a more complete OWL inference than other generic approaches, and allows to retrieve easily all the information within a context for the same entity.

Acknowledgments. This project is supported by funding received from the European Union Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 642795.

References

1. Batsakis, S., Petrakis, E.G.M.: Representing temporal knowledge in the semantic web: the extended 4D fluents approach. In: Hatzilygeroudis, I., Prentzas, J. (eds.) Combinations of Intelligent Methods and Applications. SIST, vol. 8, pp. 55–69. Springer, Heidelberg (2011)
2. Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: A Multi-dimensional Contexts Ontology. Univ. Jean Monnet, Technical report (2016)
3. Hernández, D., Hogan, A., Krötzsch, M.: Reifying RDF: what works well with wikidata? In: Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems, pp. 32–47 (2015)
4. Nguyen, V., Bodenreider, O., Sheth, A.P.: Don't like RDF reification?: making statements about statements using singleton property. In: WWW, pp. 759–770. ACM Press (2014)
5. Welty, C.: Context slices: representing contexts in OWL. In: Workshop on Ontology Patterns. CEUR Workshop Proceedings, vol. 671, p. 59. CEUR-WS.org (2010)
6. Welty, C., Fikes, R.: A reusable ontology for fluents in OWL. In: Proceedings of the First International Conference of Formal Ontology in Information Systems (FOIS). Frontiers in Artificial Intelligence and Applications, vol. 150, pp. 226–236. IOS Press (2006)
7. Zamborlini, V., Guizzardi, G.: On the representation of temporally changing information in OWL. In: Enterprise Distributed Object Computing Conference Workshops (EDOCW), pp. 283–292. IEEE (2010)