Selçuk-Teknik Dergisi **ISSN 1302-6178** *Journal of Selcuk-Technic*
Özel Sayı 2018 (ICENTE'18) *Special Issue 2018 (ICENTE'18)*

TEMPORAL EXTENSIONS TO RDF

Di WU[1], Abdullah Uz TANSEL[2, *,+]

[1]Department of Computer Science, Graduate Center, Cuny, USA

[2]Pattaya Campus, Thammasat University, Thailand

[*]On leave from Department of Information Systems and Statistics, Baruch College and the Graduate Center, Cuny, USA

wudi99@gmail.com, Abdullah.Tansel@baruch.cuny.edu

Abstract

The Semantic Web is based on Resource Description Framework (RDF) which is widely used in practice. RDF represents information by only binary predicates. This simple representation scheme forms the basis of elaborate layers of methodologies, called Semantic Web Layer Cake. Though simple, it is very powerful in modeling data and basic knowledge. However, it is very limited in representing their temporal variation. Reification is the method proposed in RDF for modeling temporal changes in data and knowledge. Moreover, reification is cumbersome since it requires at least four more triples to represent just one temporal fact. By their very nature, RDF repositories are large in general and reification causes them to explode in size. In this paper, we review Semantic Web techniques that are proposed for representing temporal data in RDF.

Keywords: Semantic Web, Temporal Data, Temporal Knowledge.

1. Introduction

The "Semantic Web", introduced by Tim Berners-Lee et al. in their 2001 article, is defined as "an extension of the current one [WWW], in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [1]. Along with the wide spread expansion of Internet, Semantic Web is adopted by major search engines, governments, and big companies.

The Semantic Web has been proposed two decades ago; and its methodologies such as

[+] This paper has been presented at the ICENTE'18 (International Conference on Engineering Technologies) held in Konya (Turkey), October 26-28, 2018.

Resource Description Framework (RDF) being implemented in many domains, albeit slowly. Semantic Web allows sharing data and knowledge that are dispersed on servers connected via Internet not only by human users but also by computers. It enriches the web pages by semantic information that can be automatically processed by computers. The semantic information in RDF is represented by only binary predicates, i.e., triples in the form of ⟨subject, predicate, object⟩. This simple representation scheme is the basis of elaborate layers of methodologies, called Semantic Web Layer Cake, to include more semantics that allow inferencing of new triples from the given ones. Though this simple representation is very powerful for modeling data and basic knowledge, it is very limited in representing their temporal variation. Reification is the method proposed in RDF for modeling temporal changes in data and knowledge. However, reification is cumbersome since it requires at least four more triples to represent just one temporal fact. By their very nature RDF repositories are large in general and reification causes them to explode in size. In this paper, we review various representative Semantic Web techniques that are proposed for representing temporal data in RDF, in an effort to lay the foundation for developing a better solution for representing temporal knowledge in RDF.

The Semantic Web Layer Cake introduces conceptual structures and richness for enhanced semantic. The stack of conceptual tools in the Semantic Web Layer Cake is downward compatible. The bottom layer, Universal Resource Identifier (URI), provides a unique Id for every subject, object, and relationship thus makes linkage and integration among different knowledge bases possible [1]. Extensible Markup Language (XML) and XML Schema provide a standard for writing structured web documents with a user-defined vocabulary. RDF is a basic data model that includes a set of statements which are triples. RDF Schema (RDFs) is based on RDF, but provides modeling primitives for building hierarchy of objects. On top of RDF and RDFs, ontology vocabularies add more power for representing more complex relationships among objects. Web Ontology Language (OWL) is a logic-based language which allows more inferencing, so machine agents are able to exploit knowledge expressed in OWL. OWL 2 as the most current version includes structures as classes, properties, individuals, and data values. RDF/XML is the only mandatory syntax for RDF, RDFs, and OWL 2. Other notations are proposed for ease of use, such as OWL/XML, Functional Syntax, Manchester Syntax, and Turtle, etc. [8, 12]. The upper layer of Semantic Web Layer Cake includes logic and proof, to derive information from the knowledge base represented by the bottom layers.

A temporal database has temporal data, and is able to deal with insertion, deletion, and query of temporal data. There are two major types of time: valid time and transaction time. Valid Time is a time period during which the data is true (valid) in the real world. Transaction Time is a time period during which the data is recorded in the database. Bitemporal Time covers both valid and transaction time. Based on the type of time support, temporal databases have three forms: historical databases that support only valid time, rollback databases that support only transaction time, and bitemporal databases that support both. Naturally, previous research in temporal databases provides a solid basis for adding temporality to Semantic Web.

RDF triples are statements without temporal attributes; and they are assumed to be true at present. However, many applications need temporal knowledge and data. To build a temporal extension for the Semantic Web, researchers proposed extending RDF with a temporal component. There are various proposals that extend RDF to express temporal knowledge. In this study, we review the major extensions to RDF to handle temporal knowledge. It is not comprehensive, however, we believe it is a satisfactory examination of the literature that would provide the reader sufficient background to understand temporal aspects of Semantic Web.

In the remaining of this paper, we first briefly explain the basic RDF model. Then we examine the proposals that add time dimension to RDF in detail. We summaries these proposals and develop a taxonomy that classifies them to provide further insight into representing temporal knowledge.

2. RDF

RDF is a framework for expressing information about resources, which include documents, people, physical objects, and abstract concepts. RDF is designed for applications to process information that are included in the webpages by providing a common framework by which applications can exchange information without loss of meaning. The subject and object in an RDF triple represent two resources and the predicate represents the relationship between the subject and the object. This relationship is directional from the subject to the object, and is also called an RDF property. RDF triples can be represented as a directional graph that has nodes for subjects and objects and directed edges for predicates. These three elements in a triple can be Internationalized Resource Identifier (IRI), blank nodes, and literals.

An IRI is a Unicode string that can be used instead of Universal Resource Identifier (URI) to identify resources [4, 5]. For example, an IRI "http://example.com" denotes a website; "http://example.com/data" denotes a subcategory of the website; and "http://example.com/data/list.html" denotes a particular web page. IRIs have global scope and IRIs are absolute and may additionally have a fragment identifier (#). Two IRIs are equivalent only when they are equivalent strings. Thus, two IRIs may be not equivalent but refer to the same resource. For simplicity, namespaces are introduced to the syntax to represent repeated portion of the IRIs. In this paper, we use namespace "ex:" for "http://example.com/" and another useful namespace "rdf:" as "http://www.w3.org/1999/02/22-rdf-syntax-ns#".

A blank node represents any resource that does not have an IRI but has meanings in triples if represents existential quantification. For example, if "a person has something to do", although we cannot express that "something" with an IRI, but we can use a blank node as a place holder. Blank nodes have some unique characteristics: 1) Blank nodes have local scope and cannot be externally referenced; 2) Blank nodes are used to represent some resources that do not need or cannot be assigned with IRIs, i.e. existential quantifications; 3) Blank nodes have blank node identifiers that are local. In RDF, literals can appear only as object and predicate can only be an IRI [3, 12].

Consider the RDF triple:

ex:William ex:livesIn ex:NYC .

The subject is the IRI referring to a person "William", and predicate is the IRI that refers to a relation "livesIn", and the object is the IRI that refers to a city, NYC. This simple triple is represented in an RDF graph as below:

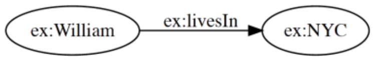

Figure 1: An Example RDF Triple

Note that this triple simply represents a true binary predicate: "William lives in NYC", livesIn(William, NYC). There is no temporal information at all. We don't have information about when this statement is recorded, nor when the statement becomes valid and expired. It is usually assumed to represent current (present) knowledge.

3. Temporal Extensions to RDF

3.1 Reification

Reification is a logic construct and is W3C working group recommendation [8, 12]. The RDF vocabulary, rdf:Statement, rdf:subject, rdf:predicate, and rdf:object, are used for reification purpose.

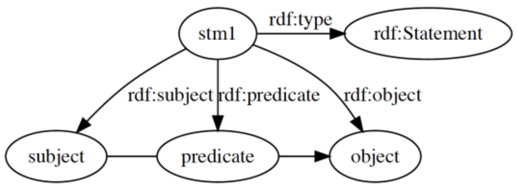

Figure 2: RDF Reification.

For example, In order to introduce the temporal information, that the triple ⟨ex:William, ex:livesIn, ex:NYC⟩ is valid in 2004, we need to create a blank node "_:x" representing a statement and 4 additional triples will be created to express the additional temporal information.

_:x rdf:type rdf:Statement .

_:x rdf:subject ex:William .

_:x rdf:predicate ex:livesIn.

_:x rdf:object ex:NYC .

_:x ex:isValid 2004.

If there is another statement, "William lives in Boston in 2018", using Reification approach, another blank node "_:y" and additional triples have to be created as well. Although "William" in "_:x" is the same "William" in "_:y", we still have to create two separate triples. We can see from this example that Reification has a significant data redundancy problem. Not only we need to use four triples to express one temporal fact, but also the four triples are hard to be reused. Furthermore, if we need to express a bitemporal information, a second layer of reification has to be introduced and many more triples have to be added.

3.2 Temporal RDF

Gutierrez et al. propose "temporal RDF" to add temporal information into standard RDF [6]. Temporal RDF adopts the labeling solution to handle evolving RDF data. The definition of Temporal RDF starts with a Temporal Triple: *(s, p, o): [t]*, where *(s, p, o)* is an RDF triple, and *t* is the temporal label. In this definition, *t* is a natural number that represents a time point. To represent a time interval, the temporal triple is expressed as $(s, p, o): [t_1,t_2]$ as the union of all *(s, p, o): [t]* where $t_1 \leqslant t \leqslant t_2$. A Temporal Graph is a set of Temporal Triples. This definition has several advantages: 1) It avoids the complex format of time. A natural number is used to represent a time point, which is very simple and straightforward; 2) It integrates point-based and interval-based temporal information; 3) It combines an existing RDF triple with a temporal label so that modification is minimized; 4) The temporal label can be further extended for not only valid time, but also transaction time; 5) It is entirely within the standard definition of RDF.

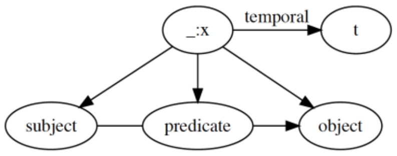

Figure 3: temporal RDF

Gutierrez et al. further extend the temporal RDF model to include implicit temporal labels (anonymous time) [7]. In their original paper, a temporal label *t* associates with either a time point which is a natural number, or a time interval that composed by two time points [6]. The advantages of having anonymous time are: 1) In the case that triples lack precise temporal information, anonymous time can specify them naturally. With anonymous time, RDF triples can easily be converted to a temporal graph. 2) Anonymous time itself has many uses. For example, a triple A occurs at time t_1, and another triple B occurs at the same time as A does. Although we don't know the explicit time point of the "same time" that triple B occurs at, we can infer by both triples A and B that the "same time" is actually t_1.

3.3 Named Graph

The named graph approach is developed by Carroll et al. [2] and Tappolet and Bernstein [13]. Carroll et al. define a named graph as an RDF graph with a URIref. Let *U* be URIs, *B* be Blank Nodes, *L* be Literals, then all nodes in an RDF graph are $V = U \cup B$

$\cup L$. The set $T = V \times U \times V$ is the set of all RDF triples (or an RDF graph). The set of all RDF graphs G is the power set of T. A named graph *ng* is a pair (n, g) with $n \in U$ and $g \in G$ [2].

Named graph is adopted by W3C Recommendations in the RDF dataset definition as a standard RDF feature: The dataset is a default RDF graph (with no name) and some (zero or more) named graphs may also be included. Or more formally: An RDF dataset is a set of $\{G, (u_1, G_1), (u_2, G_2), . . . (u_n, G_n)\}$ where G is the default graph, and (u_i, G_i) are named graphs where u_is are URIref and G_is are graphs. Each u_i is distinct [3].

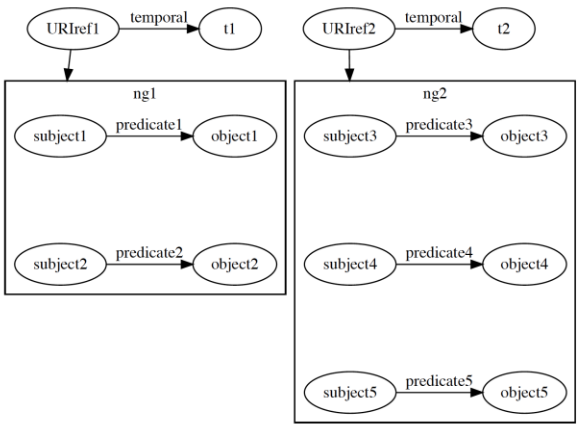

Figure 4: Named Graph

Tappolet and Bernstein adopt the named graph approach for adding temporal interval into RDF triples [13]. Because named graph is adopted as part of the RDF dataset definition, the difficulty of using this approach for adding temporal information is minimal. Since each named graph *ng* has a name which is a URIref and has a set of triples, the temporal information *t* can be attached to the URIref so that all the triples of that named graph share the same temporal information *t*.

3.4 4D Fluents

McCarthy and Hayes define a fluent as a function that maps from objects and situations to truth [9]. Thus, fluents are relations that hold within some time interval and not in others. Welty and Fikes further develop the 4D Fluents approach to describe information changes over time [15]. The basic problem is how to logically account for the fact that "same" entity appears to be "different" at different times. 3D view consider endurants that are wholly present at all times and perdurants that have temporal parts that only exist during the time the entities exist. 4D view (also called perdurantist view) maintains that all entities are perdurants. Thus, all entities have temporal parts and can be thought of intuitively as four dimensional.

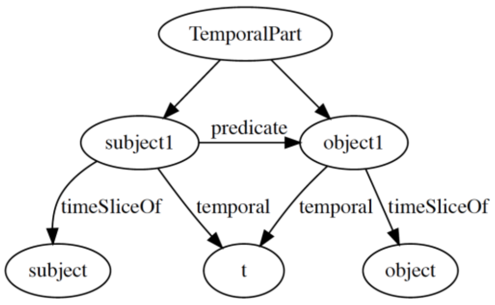

Figure 5: 4D Fluents

One consequence of a 4D approach is that time is "bundled in" with the temporal parts themselves and binary fluents can be represented as a simple binary relations between them. The time interval of a temporal part is defined to be the duration when a fluent holds. A single fluent in the 4D approach requires two extra objects, a temporal part and additional six triples, which is more than needed in reification. The most significant aspect of 4D Fluents approach is that it benefits the defined capabilities of OWL, such as transitive, inversive etc.

*3.5 RDF**

RDF* differs from RDF by allowing a triple in place of subject and object resources in a triple: such triples are called RDF* triples. Thus, the definition of an RDF* triple allows nesting: an RDF* triple may have another RDF* triple as its subject or object. The depth of the nesting is denoted as *k*. Thus, if *k* = 0, the triple is an ordinary RDF triple; and if *k* > 0, it is a nested RDF* triple.

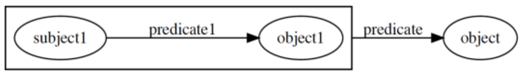

Figure 6: RDF*

Since the definition of RDF* allows nesting, the unfolding process is defined recursively: If the unfolded RDF graph contains RDF* triples, then the transformation process will continue until all triples are ordinary RDF triples. Turtle* is provided by adding three additional terms: tripleX for an RDF* triple, subjectX and objectX for referring the subject and object of a tripleX. Also, the standard Turtle grammar is updated to include the definition of subject and object in tripleX. SPARQL* is also developed as a metadata extension of SPARQL. This extension enables querying for RDF* graphs. The specification of SPARQL* grammar has an embedded triple pattern as a new syntax element.

3.6 Singleton Property

Nguyen et al. [10] make a general extension to RDF by introducing singleton property. Temporality is one special kind of singleton property in this approach. The singleton property is based on the notion that "the nature of every relationship is universally unique". A triple in RDF is considered as a relation that connects two entities. In every context, the relation is unique. There may be many relations between two entities, such as time, location, source, certainly, etc. The authors give examples to show how generic properties together with their sub-properties that represent different types of information about statements. For example, "isMarriedTo" relationship is considered as a generic property. All relevant information about "isMarriedTo" relationship is represented as its sub-properties. If there are two triples that have same "isMarriedTo" relationship between the same entities, then two sub-properties will be created, and each corresponding to its unique source. If there is temporal information, then it will be attached to the singleton property that represents the temporal relationship [10].

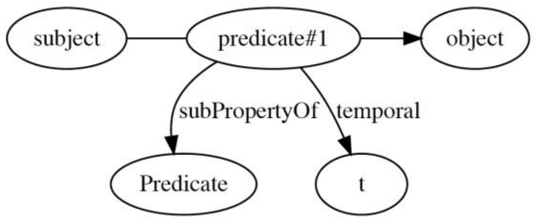

Figure 7: Singleton Property

Thus, by definition, for a triple *(s, p, o)*, *p* is a generic property, and *p* has multiple sub-properties as singleton property, denoted as *p#i*. Consequently, the relationship that connects *s* and *o*, is not *p*, but *p#i* and it defines a unique context. Each *p#i* defines multiple relationships for instantiating different values for *i*. It is important to keep the singleton property unique to avoid inconsistencies. Universally Unique Identifier (UUID) can be used to address this problem. UUID is supported by SPARQL and programming languages.

There are two types of queries: for data and for metadata that includes singleton properties. Querying for data basically follows standard RDF notation whereas querying metadata needs to include multiple triples in the query pattern. Nguyen et al. compare and evaluate five approaches: the Singleton Property, standard RDF reification, and three flavors of Provenance Context Entity (PaCE) in an experiment. In the comparison, three factors are considered: number of triples, query length, and time for query execution. The result shows the Singleton Property approach has better performance [10] .

3.7 N-ary Relations

N-ary Relations approach addresses three major issues in RDF/OWL binary representation of relationships. Since a binary relation can only relate two individuals, then there are three issues that need to be addressed: 1) How to describe the relation itself; 2) How to relate more than two individuals; and 3) How to state the linked individuals are ordered. The sample scenarios for each problem are: 1) A patient with a disease and its probability; 2) A transaction that has buyer, seller, price, purpose, etc.; and 3) A flight that has multiple ordered airports on its route. The proposed solutions as n-ary relations for the three use cases are: 1) Introduce a new class of the particular relation, add the disease and probability as property of that class, then use an instance of that relation class to replace the old relation; 2) Introduce a new class of the product that has all other individuals as properties; 3) Use a list of arguments in a relation and use constraint on cardinality for arguments to represent city-on-the-road and destination [11].

Noy et al. distinguish n-ary relations and reification in RDF [11]. The most significant difference between these two is: reification focuses on the statement which has 3 parts *(s, p, o)*; while n-ary relations focuses on the property, which is one part of the statement.

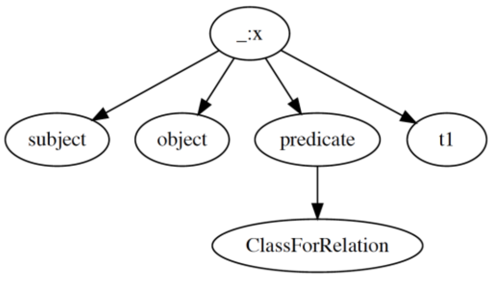

Figure 8: N-ary Relations

3.8 Annotated RDF

Udrea et al. develop the Annotated RDF (aRDF) as a generic extension to the standard RDF model. This approach uses aRDF as a single uniform framework to support various extensions, such as uncertainty, pedigree, time, etc. [14]

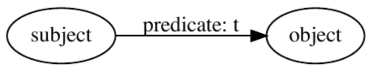

Figure 9: Annotated RDF

Annotated RDF, short for aRDF, has two parts. The first part is an ordinary RDF triple, *(s, p, o)*; and second part is an annotation *a* from a partially ordered set *A*. An annotated RDF triple thus, is in the form of *(s, p:a, o)*. For a temporal extension, *A* can be a set of non-negative integers representing time instants. *A* can also be a pair of ordered integers *x*, *y* as time intervals [14]. In addition, the annotation is attached to the predicate *p*, not the entire triple *(s, p, o)*. Also, aRDF is similar to temporal RDF. In fact, we can consider temporal RDF as a variation of aRDF. However, temporal RDF uses arbitrary labeling t on ordinary RDF triples as *(s, p, o)[t]* while aRDF uses annotations from a partially ordered set on predicates as *(s, p:a, o)*.

4. Taxonomy of Temporal Extensions to RDF

We examine various approaches that extend standard RDF for temporal information. Since all of them extend the standard RDF model and they are developed with the goal of minimizing the needed modifications they have common aspects and also differences. These approaches convert one or more element of the triples into a meta resource to carry extra information. Depending on the element converted, we can categorize these approaches: graph, triple, or element level.

Graph level: As an RDF dataset contains one or more RDF graphs, named graph approach uses the URIref of the named graph to carry temporal information. A named graph attaches the same time reference on all of its triples.

Triple level: Triple level extensions treat an RDF triple as metadata and convert the triple into a resource to carry temporal information. Reification explicitly converts an RDF triple to a statement; and temporal RDF implicitly attaches a temporal label to an RDF triple.

Element level: 4D fluents converts the subject and object in a triple to be a timeslice of a fluent object to carry temporal information. Singleton Property converts the predicate in a triple and defines a unique sub-property that only exists in a temporal context. N-ary Relations creates a new class for the relation to carry temporal information. Annotated RDF attaches a partially ordered annotation to an RDF triple, and in terms of its syntax, the annotation is attached to the predicate. RDF* allows the subject and object in a triple to be metadata to form nested triples.

5. Conclusion

Standard RDF model is built upon the simple *(s, p, o)* triple structure and thus is limited in expressing only binary relationships. Along with the need to embed temporal information to standard RDF, scholars have developed various approaches. We examine these temporal extensions to RDF; and categorize them into three groups depends on how they extend the standard RDF model. Graph level extensions have Named Graph; Triple level extensions have Reification and temporal RDF; Element level includes 4D fluent, Singleton Property, Annotated RDF, RDF* and N-ary Relations approaches. Naturally, each extension has performance differences in space and computation time. A detailed performance comparison of these extensions would be a beneficial study. We plan to explore this direction and also search for a better approach for handling temporal knowledge.

Acknowledgment

The research of second author is partially supported by the PSC CUNY Award, 60342-00-48.

References

[1] Tim Berners-Lee, Ora Lassila, and James Hendler. The semantic web. Scientific American Magazine, pages 29–37, May 2001.

[2] Jeremy J. Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named graphs. Web Semant., 3(4):247–267, December 2005.

[3] Richard Cyganiak, David Wood, and Markus Lanthaler. Rdf 1.1 concepts and abstract syntax. W3C Recommendation, Feb 2014.

[4] M. Duerst and M. Suignard. Internationalized resource identifiers (iris). https://tools.ietf.org/html/rfc3987. Accessed: 2018-06-16.

[5] M. Dürst and M. Suignard. Internationalized resource identifiers (iris). RFC, page 832–843, 2005.

[6] Claudio Gutierrez, Carlos A. Hurtado, and Alejandro Vaisman. Temporal RDF. In Proceedings of Second European Semantic Web Conference, pages 93–107, 2005.

[7] Claudio Gutierrez, Carlos A. Hurtado, and Alejandro Vaisman. Introducing Time into RDF. IEEE Trans. on Knowledge and Data Engineering, 19:207–218, February 2007.

[8] Frank Manola and Eric Miller. Rdf 1.0 primer. W3C Recommendation, 2004.

[9] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, Machine Intelligence 4, pages 463–502. Edinburgh University Press, 1969. Reprinted in McC90.

[10]Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. Don't like rdf reification?: Making statements about statements using singleton property. In Proceedings of the 23rd International Conference on World Wide Web, WWW '14, pages 759–770, New York, NY, USA, 2014. ACM.

[11]Natasha Noy, Alan Rector, Pat Hayes, and Chris Welty. Defining N-ary relations on the semantic web. W3C Working Group Note, 12:4, 2006.

[12]Guus Schreiber and Yves Raimond. Rdf 1.1 primer. W3C Working Group Note, 2014.

[13]Jonas Tappolet and Abraham Bernstein. Applied temporal rdf: Efficient temporal querying of rdf data with sparql. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl, editors, The Semantic Web: Research and Applications, pages 308–322, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[14]Octavian Udrea, Diego Reforgiato Recupero, and V. S. Subrahmanian. Annotated rdf. ACM Trans. Comput. Logic, 11(2):10:1–10:41, January 2010.

[15]Christopher Welty, Richard Fikes, and Selene Makarios. A reusable ontology for fluents in OWL. In Proceedings of FOIS, pages 226–236, 2006.