Linking Justifications in the Collaborative Semantic Web Applications

Rakebul Hasan and Fabien Gandon
INRIA Sophia Antipolis – Wimmics
2004 rt des Lucioles, BP93
Sophia Antipolis, 06902, France
{hasan.rakebul,fabien.gandon}@inria.fr

ABSTRACT
Collaborative Semantic Web applications produce ever changing interlinked Semantic Web data. Applications that utilize these data to obtain their results should provide explanations about how the results are obtained in order to ensure the effectiveness and increase the user acceptance of these applications. Justifications providing meta information about why a conclusion has been reached enable generation of such explanations. We present an encoding approach for justifications in a distributed environment focusing on the collaborative platforms. We discuss the usefulness of linking justifications across the Web. We introduce a vocabulary for encoding justifications in a distributed environment and provide examples of our encoding approach.

Categories and Subject Descriptors
H.5 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*computer-supported cooperative work, web-based interaction*; D.2.5 [**Software Engineering**]: Testing and Debugging—*distributed debugging, tracing*

General Terms
Design, Reliability

Keywords
Explanation, Justification, Trust, Linked Data, Collaborative Semantic Space

1. INTRODUCTION
Semantic Web-based collaborative platforms such as semantic wikis [15], DBPedia[1], Freebase[2], or YAGO[3] are continuously producing a growing amount of Semantic Web data.

---

[1]http://dbpedia.org/
[2]http://www.freebase.com/
[3]http://www.mpi-inf.mpg.de/yago-naga/yago/

The data produced by the collaborative applications are continuously changing and evolving, and therefore, new information derived from these data continuously need updates as well, resulting in ever changing interlinked data sources which Passant [13] termed as Collaborative Semantic Space. Federated applications in this Collaborative Semantic Space, which utilize the available data from these highly changing and evolving distributed data sources, should allow tracing the origin of the resulted information in order to impart an understanding of how the resulted information came to their existence and hence allow users' trust on the results [5]. Providing explanation about why a particular piece of information is derived from the distributed data sources is one way enable details of the origin of information.

Semantic Web applications should provide explanations about how the results are obtained in order to ensure their effectiveness and increase their user acceptance [11]. The interoperating Semantic Web applications, especially applications in collaborative settings, should not only provide explanations about how the answers were obtained, they should also explain and allow users to follow the flows of information between them [12]. Generation of such explanations requires additional metadata about why a conclusion has been drawn. This kind of additional information about the derivation of a conclusion is commonly known as justification. In the Collaborative Semantic Space, the RDF data distributed across the Web contain the triples representing ground facts generated from the community contribution and triples inferred from the ground facts. Moreover, justifications for such distributed inferred knowledge themselves are distributed across the Web. Federated applications and their reasoning processes in this dataspace use this data and perform new inferences. Such a scenario leads to the requirement of explicitly linking related justifications a distributed setting – *i.e.* applying the very approach of linked data [3] to the representation of justifications themselves. Explanations generated from these linked justifications enable navigation between the explanations of related information distributed across the Web providing details about their origin.

Linked justifications are also useful for truth maintenance in a distributed environment. This is very interesting especially in the case of collaborative Semantic Web applications as their knowledge bases continuously change and evolve. As a real world example, consider the case of DBPedia Live[4] and the chains of inferences that are dependent on the data

---

[4]http://wiki.dbpedia.org/DBpediaLive

produced by DBPedia Live. DBPedia Live keeps DBPedia always in synchronization with Wikipedia. DBpedia Live updates its knowledge base in response to the changes in the Wikipedia articles. The chains of applications which are dependent on the DBPedia Live data will benefit from a truth maintenance platform that enables tracing the origin of inferences. Linked justifications will provide the basis for such a platform by enabling to determine the triples that might be affected if a triple is modified or removed, and thus avoid the inefficient incremental approach of removing and recomputing all the inferred triples [9].

In this paper, we introduce the concept of linked justifications in the Collaborative Semantic Space. We provide outlines for encoding and linking justifications focusing on the collaborative platforms. These collaborative platforms allow the end users to incrementally develop their knowledge bases, and collaborate among themselves in a distributed setting in order to enrich and complement their knowledge. The rest of this paper is organized as follows: We introduce a motivating scenario in section 2. In section 3, we describe our proposed approach to encoding and linking justifications. Then we give examples of our encoding and linking approach in section 4. In section 5, we discuss the related work. Finally, we conclude and provide outlines for future work in section 6.

2. A MOTIVATING SCENARIO
Our running example includes the following three applications in the Collaborative Semantic Space:

- A semantic wiki, called *AcadWiki*, which allows creation of a knowledge base of academicians in a collaborative way.

- A semantic wiki, called *GeoWiki*, which allows creation of a geographical knowledge base in a collaborative way.

- A federated application, called *Academician Locator*, which makes use of the data available from both semantic wikis to compute its results.

Semantic wikis have the ability to derive new facts from the base facts. For example, when someone is described as a computer scientist, *AcadWiki* makes the inference that this person is also a scientist. Similarly in *GeoWiki*, when London is described as a part of England and England is described as part of United Kingdom, *GeoWiki* makes the inference that London is part of United Kingdom. The facts in these two semantic wikis are interlinked. For example, a computer scientist described in *AcadWiki* can have a property *birthPlace* to specify his/her birth place that is described in *GeoWiki*. A statement such as "Bob was born in London" links two resources in the two different semantic wikis. *AcadWiki* and *GeoWiki* both make their knowledge bases, which include base facts and derived facts, available in RDF and publish them by following the linked data principles. In addition, these two applications make their data accessible via SPARQL endpoints. The *Academician Locator* federated application utilizes the data made available by the two semantic wikis, derives new facts, publishes them by

following the linked data principles and makes its data accessible via a SPARQL endpoint. Figure 1 shows an extract of the RDF data from the two semantic wikis. We omit the namespaces throughout this paper for better readability. Two dashed boxes separates the RDF data in the two semantic wikis. The dashed arrows show the inferred triples.
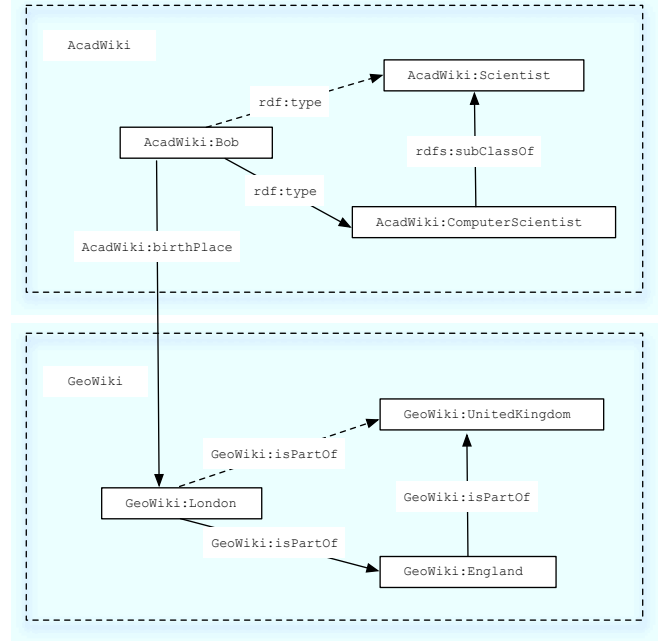

Figure 1: Extract of the RDF graphs from *AcadWiki* and *GeoWiki*

In *AcadWiki*, the fact that *AcadWiki:Bob* is a member of the *AcadWiki:Scientist* class is inferred by propagating *rdf:type* relationship [1]. In *GeoWiki*, the fact that *GeoWiki:London* is part of *GeoWiki:UnitedKingdom* is inferred because the *GeoWiki:isPartOf* property is defined as a transitive property. The federated application utilizes the available data from the two semantic wikis and makes its own inferences by applying its own rules. Figure 2 shows the two new inferences made by the federated application using the data available from the two semantic wikis.

In a scenario such as the one presented above, one might want to know why a particular inference has been made. For example, one might want to know why the federated application thinks that Bob is a Scientist born in United Kingdom. In the remaining of the paper, we present our approach to encoding and linking justifications which enables answering such questions.

3. ENCODING APPROACH
We represent the justifications following the linked data principles. We assign identifiers to all the resources in our approach. We use resolvable HTTP URLs as identifiers. We intentionally avoid using blank nodes as suggested in [6]. In our approach, we make statements about statements, e.g. statement about a triple that its assertion is justified by
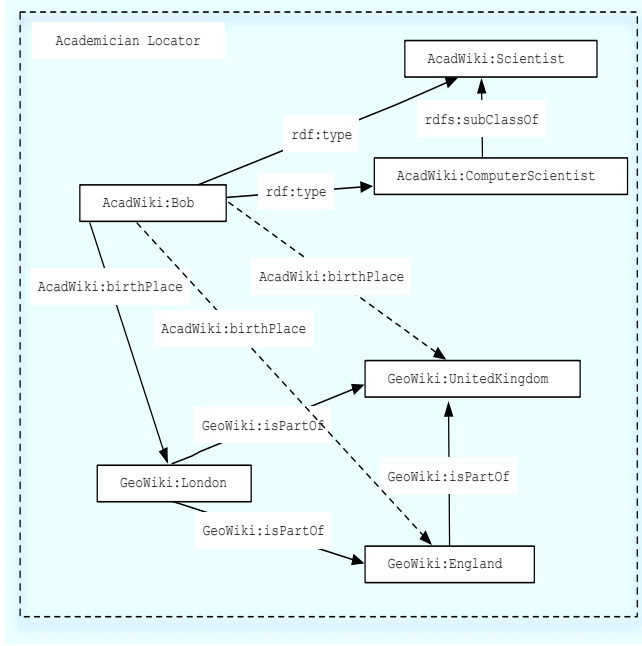
Figure 2: The new inferences made by the federated application shown by the dashed arrows

a justification. A justification itself is a collection of some statements. Therefore, we needed a mechanism which allows making reference to triples and making statements about triples. We use the named graphs data model proposed by Carroll *et al.* [4] which allows naming an RDF graph containing a collection of RDF triples. The names of the named graphs are resolvable HTTP URLs in our case as required by the linked data principles. This makes it possible to refer to the named graphs distributed across the Web and also to get useful information about the named graphs via HTTP GET requests. Each justification is a named graph. A justification contains a set of triples which justify the assertion of a triple. We provide triple level granularity for triples representing ground facts and the inferred triples. Each triple is encoded into its own named graph with only one triple that is the triple itself. This allows making statements about triples and enable triple level granularity. We use a lightweight vocabulary in combination with the named graphs data model to describe justifications.

3.1 The Ratio4TA Vocabulary

Ratio4TA (inte*r*linked justific*atio*ns *for t*riple *a*ssertions)[5] is a lightweight vocabulary for encoding justifications using named graphs.

As shown in Figure 3, the initial version of our vocabulary includes the following classes and properties:

Assertion class describes an asserted triple. The *Assertion* class is a subclass of the *rdfg:Graph* class.

InferredAssertion is a subclass of the *Assertion* class.
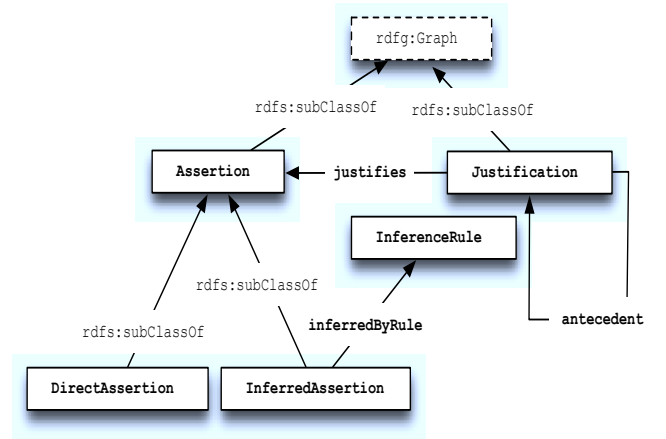

Figure 3: The classes and properties of the *Ratio4TA* vocabulary

The *InferredAssertion* class describes an asserted triple that is inferred from other triples.

DirectAssertion is a subclass of the *Assertion* class. The *DirectAssertion* class describes a directly asserted triple that represents a ground fact.

Justification class describes a justification. A justification can justify the assertion of an inferred triple or the assertion of a triple representing a ground fact. The *Justification* class is a subclass of the *rdfg:Graph* class.

InferenceRule class represents an inference rule that has been enforced to infer a triple. How rules will be encoded are not restricted to a particular encoding on purpose to accommodate different kinds of rule based systems distributed across the Web.

antecedent property links a justification to the justifications for the assertions of triples from which the inferred triple of linking justification has been derived.

justifies property expresses the relation that a justification justifies the assertion of a triple.

inferredByRule property relates an inferred assertion with a rule that has been enforced to infer the triple.

We define the *Assertion* class and the *Justification* class as named graphs by extending the *rdfg:Graph* class, shown in the dashed box in Figure 3, defined by Carroll *et al.* [4]. An *Assertion* named graph contains a triple, which makes the triple referenceable. For instance, the triple `AcadWiki:Bob AcadWiki:birthPlace GeoWiki:UnitedKingdom` in *Academician Locator* is encoded in a named graph. Assume that the identifier of this named graph is *aloc:t1*. So the triple is now referenceable using the named graph identifier *aloc:t1*. A *Justification* named graph contains a group of triples that justify the assertion of a triple. A *Justification* named graph for an inferred triple contains the following components:

1. A triple expressing the fact that the justification named graph is a member of the *Justification* class. Assume

that the identifier of a justification named graph is *aloc:j1*. This named graph will contain the triple `aloc:j1 rdf:type r4ta:Justification` to express the fact that *aloc:j1* is of type *Justification*[6].

2. A triple expressing the fact that the justification justifies the assertion of a triple. The *justifies* property is used to express this relation. Assume that *aloc:j1* justifies the assertion of the triple *aloc:t1*. This fact will be expressed by the triple `aloc:j1 r4ta:justifies aloc:t1`.

3. A set of triples specifying the antecedent justifications of the current justification. An inferred triple is inferred from other triples. The justifications of these other triples are the antecedents of the justification of the inferred triple. The triple *aloc:t1* in *Academician Locator* is inferred from the triple `AcadWiki:Bob AcadWiki:birthPlace GeoWiki:London` in *AcadWiki* and the triple `GeoWiki:London GeoWiki:isPartOf GeoWiki:UnitedKingdom` in *GeoWiki*. Assume that the assertions of these two triples are justified by *AcadWiki:j4* and *GeoWiki:j1* subsequently. Therefore, the justifications *AcadWiki:j4* and *GeoWiki:j1* are the antecedent justifications for the justification *aloc:j1*. These two antecedent relations will be captured by the triple `aloc:j1 r4ta:antecedent AcadWiki:j4` and the triple `aloc:j1 r4ta:antecedent GeoWiki:j1`.

4. A triple expressing the fact that the assertion of the justified triple is an inferred triple. This fact is expressed by specifying the inferred triple as a member of the *InferredAssertion* class. In our example, the inferred triple *aloc:t1* is specified as a member of the *InferredAssertion* by the triple `aloc:t1 rdf:type r4ta:InferredAssertion`.

5. A triple expressing the fact that the inferred triple is inferred by enforcing an inference rule. The *inferredByRule* property is used to relate an *InferredAssertion* with an *InferenceRule*. In our example, assume that the inferred triple *aloc:t1* is inferred by enforcing the inference rule *aloc:pobRule*. Therefore, the triple `aloc:t1 rdf:InferenceRule aloc:pobRule` will be added in the justification named graph.

A justification for the assertion of a triple representing a ground fact is encoded in the same way. The difference is that such a justification does not contain any antecedent. The asserted triple that is justified by the justification is specified as a member of the *DirectAssertion* class. In addition, no inference rule is specified since no inference is performed for a ground fact.

In our approach, we explicitly link related justifications. Justifications are generated by the reasoners that are distributed across the Web. For example, justifications generated by *AcadWiki*, *GeoWiki*, and *Academician Locator* reside in different data sources. Therefore, the related justifications distributed across the Web should have the possibility to express their relation. The *antecedent* property

in our vocabulary expresses the antecedent dependency between justifications. The consumers of the justifications can determine the triples from which a given triple is inferred by exploiting the link structure of the justifications. In essence, linking justifications allow maintenance of related facts and justifications in a distributed settings. In addition, our encoding allows explicitly describing which are the inferred triples and which are the triples representing the ground facts. This separation is an important feature in regards to writing efficient algorithms in order to exploit these justification data.

The encoding of rules is out of the scope of this paper. We focus on encoding and linking justifications. However, our proposal is to use SPIN[7] for representing SPARQL rules in RDF. This will allow to write the rules once, then enforcing them to make inferences; linking them from the justifications as they are also RDF resources with identifiers; and finally providing human understandable abstraction of them for explanation.

We opted for our own vocabulary because the existing vocabulary, the Proof Markup Language (PML) [10], has high complexity [14] and limitations with regard to our approach. PML uses an RDF container like concept called *NodeSetList* to specify the antecedent justifications. RDF containers are described using blank nodes to connect a sequence of items [1]. In our approach, we make all the resources referenceable by avoiding blank nodes. For this reason, PML is not compatible to our approach and hence we opted for our own vocabulary.

3.2 Design Decisions

We choose the named graphs data model over RDF reification for making statements about statements. The advantages of named graphs over RDF reification are discussed in [4, 16]. Furthermore, we use named graphs to group together the justification related statements for a triple assertion in a graph so that we can make reference to those statements together. Alternatively, we could have defined a justification resource, and resources representing the other components of a justification, and then link the justification resource and the resources representing the other components using appropriate properties. However, this approach would have had added complexities because one had to traverse through several links to get all the components for a justification. In contrast, our approach of grouping all the justification related statements for a triple assertion in a named graph provides a simpler way to manage justifications. For instance, one can obtain a justification for a triple assertion just by obtaining the statements in the justification named graph for that triple assertion.

We directly link the antecedent justification named graphs from the justification named graphs for inferred triple assertions using the *antecedent* property. Other design choice we had was to directly link the antecedent triples instead of the antecedent justification named graphs. However, one would have had to traverse more links to navigate through related justifications in that case. For instance, while navigating through the justifications for a chain of inferred assertions,

---

each time he had to follow a link to an antecedent triple and then follow another link from the antecedent triple to the justification graph for that triple. In contrast, our approach allows to follow only one link and navigate to an antecedent justification for a justification. Furthermore, directly linking justifications from justifications enables a better separation of data and metadata. This separation allows to better manage the data and the justification related metadata.

3.3 Consuming Linked Justifications

The consumers of linked justifications can transform the justifications to a human understandable presentation in order to provide explanations about how answers are derived by the Semantic Web applications using data from different data sources. The justifications can be transformed and abstracted to human understandable representations such as explanation in natural language to make the reasoning processes transparent. Navigations support between the related explanations generated from the distributed justifications can be provided by exploiting the link structure of linked justifications. Linked justifications enable providing explanation in a distributed environment.

Furthermore, as previously discussed, justifications can be also used for truth maintenance in a continuously changing and evolving scenario such as the Collaborative Semantic Space. Justifications can be used to determine the triples that might be affected if a triple is modified or removed thus avoid the inefficient incremental approach of removing and recomputing all the inferred triples [9]. Linked justifications will provide the basis for a truth maintenance platform such as the one we discuss.

In the following section, we describe our approach to encoding and linking justifications with examples from *AcadWiki*, *GeoWiki*, and *Academician Locator*.

4. EXAMPLES OF ENCODING

We use the TriG notation [4], which is an extension of the Turtle notation [2], to describe our encoding examples.

The *AcadWiki* application propagates the *rdf:type* property in its RDF data up the subclass hierarchy. The definition of *rdfs:subClassOf* specifies that the meaning of "A is a subclass of B" is "every member of class A is also a member of class B" [1, 7]. The *AcadWiki:ComputerScientist* class is a subclass of the *AcadWiki:Scientist* class and *AcadWiki:Bob* is a member of the *AcadWiki:ComputerScientist* class. Therefore, it is inferred that *AcadWiki:Bob* is a member of the *AcadWiki:Scientist* class expressed by the inferred triple `AcadWiki:Bob rdf:type AcadWiki:Scientist`.

Listing 1 shows the encoding for the triples and the justifications in *AcadWiki*. The inferred triple is encoded in the named graph *AcadWiki:t1*. The *AcadWiki:t2* named graph and the *AcadWiki:t3* named graph contain the triples from which the triple *AcadWiki:t1* is inferred. The *AcadWiki:t4* named graph contains the triple that interlinks *AcadWiki* and *GeoWiki* by specifying *GeoWiki:London* as the birthplace of *AcadWiki:Bob*. The justification for the assertion of the inferred triple is encoded in the *AcadWiki:j1* named graph. The justification for the assertion of the triple *Acad-Wiki:t2* is encoded in the justification *AcadWiki:j2*. The

justification for the assertion of the triple *AcadWiki:t3* is encoded in the justification *AcadWiki:j3*. The justification for the assertion of the triple *AcadWiki:t4* is encoded in the justification *AcadWiki:j4*

The first triple in the *AcadWiki:j1* named graph specifies *AcadWiki:j1* as a member of the *r4ta:Justification* class. The fact that the justification justifies the assertion of the inferred triple *AcadWiki:t1* is expressed by relating the justification *AcadWiki:j1* with the triple *AcadWiki:t1* using the *r4ta:justifies* property. Two more triples specify the antecedent justifications of the justification *AcadWiki:j1*, namely, *AcadWiki:j2* and *AcadWiki:j3*, using the *r4ta:antecedent* property. The fact that the triple *AcadWiki:t1* is an inferred triple is expressed by specifying *AcadWiki:t1* as a member of the *r4ta:InferredAssertion* class. The last triple in the justification named graph *AcadWiki:j1* specifies that the triple *AcadWiki:t1* is inferred by enforcing the inference rule *AcadWiki:typeProp*. The encoding of the inference rule *AcadWiki:typeProp* is not shown in Listing 1 as we are not addressing how to encode rules.

Listing 1: Encoding of asserted triples and justifications in *AcadWiki*

```
#graph for triple 1
AcadWiki:t1 {
  AcadWiki:Bob rdf:type AcadWiki:Scientist.
}
#graph for triple 2
AcadWiki:t2 {
  AcadWiki:Bob rdf:type AcadWiki:ComputerScientist.
}
#graph for triple 3
AcadWiki:t3 {
  AcadWiki:ComputerScientist rdfs:subClassOf
      AcadWiki:Scientist.
}
#graph for triple 4
AcadWiki:t4 {
  AcadWiki:Bob AcadWiki:birthPlace GeoWiki:London.
}
#graph justifying the assertion of triple 1
AcadWiki:j1 {
  AcadWiki:j1 rdf:type r4ta:Justification.
  AcadWiki:j1 r4ta:justifies AcadWiki:t1.
  AcadWiki:j1 r4ta:antecedent AcadWiki:j2.
  AcadWiki:j1 r4ta:antecedent AcadWiki:j3.
  AcadWiki:t1 rdf:type r4ta:InferredAssertion.
  AcadWiki:t1 r4ta:inferredByRule AcadWiki:typeProp
    .
}
#graph justifying the assertion of triple 2
AcadWiki:j2 {
  AcadWiki:j2 rdf:type r4ta:Justification.
  AcadWiki:j2 r4ta:justifies AcadWiki:t2.
  AcadWiki:t2 rdf:type r4ta:DirectAssertion.
}
#graph justifying the assertion of triple 3
AcadWiki:j3 {
  AcadWiki:j3 rdf:type r4ta:Justification.
  AcadWiki:j3 r4ta:justifies AcadWiki:t3.
  AcadWiki:t3 rdf:type r4ta:DirectAssertion.
}
#graph justifying the assertion of triple 4
AcadWiki:j4 {
  AcadWiki:j4 rdf:type r4ta:Justification.
  AcadWiki:j4 r4ta:justifies AcadWiki:t4.
  AcadWiki:t4 rdf:type r4ta:DirectAssertion.
}
```

The triples *AcadWiki:t2*, *AcadWiki:t3*, *AcadWiki:t4* represent ground facts. Therefore, the justifications for their assertions make it explicit that they represent ground facts by declaring the corresponding triple as a member of the *r4ta:DirectAssertion* class.

Listing 2: Encoding of asserted triples and justifications in *GeoWiki*

```
#graph for triple 1
GeoWiki:t1 {
  GeoWiki:London GeoWiki:isPartOf GeoWiki:
      UnitedKingdom.
}
#graph for triple 2
GeoWiki:t2 {
  GeoWiki:London GeoWiki:isPartOf GeoWiki:England.
}
#graph for triple 3
GeoWiki:t3 {
  GeoWiki:England GeoWiki:isPartOf GeoWiki:
      UnitedKingdom.
}
#graph justifying the assertion of triple 1
GeoWiki:j1 {
  GeoWiki:j1 rdf:type Justification.
  GeoWiki:j1 r4ta:justifies GeoWiki:t1.
  GeoWiki:j1 r4ta:antecedent GeoWiki:j2.
  GeoWiki:j1 r4ta:antecedent GeoWiki:j3.
  GeoWiki:t1 rdf:type r4ta:InferredAssertion.
  GeoWiki:t1 r4ta:inferredByRule GeoWiki:
      transitivity.
}
#graph justifying the assertion of triple 2
GeoWiki:j2 {
    GeoWiki:j2 rdf:type Justification.
    GeoWiki:j2 r4ta:justifies GeoWiki:t2.
    GeoWiki:t2 rdf:type r4ta:DirectAssertion.
}
#graph justifying the assertion of triple 3
GeoWiki:j3 {
    GeoWiki:j3 rdf:type Justification.
    GeoWiki:j3 r4ta:justifies GeoWiki:t3.
    GeoWiki:t3 rdf:type r4ta:DirectAssertion.
}
```

In *GeoWiki*, the *GeoWiki:isPartOf* property is a transitive property. This means that if *A* is a part of *B* and *B* is a part of *C* then *A* is also a part of *C*. In *GeoWiki*, *GeoWiki:London* is described as a part of *GeoWiki:England* and *GeoWiki:England* is described as a part of *GeoWiki:UnitedKingdom*. Therefore, it is inferred that *GeoWiki:London* is a part of *GeoWiki:UnitedKingdom*. Listing 2 shows the encoding of the triples and justifications for their assertions in *GeoWiki*. The triple *GeoWiki:t1* is the inferred triple and its assertion is justified in the justification *GeoWiki:j1*. *GeoWiki:t2* and *GeoWiki:t3* represent the other triples and their justifications are encoded in *GeoWiki:j2* and *GeoWiki:j3* subsequently.

In our example scenario, the *Academician Locator* federated application enforces its own rules on these factual data available from both semantic wikis. For the inferences shown in Figure 2, *Academician Locator* enforces the rule: "if *A* has birth place *B* and *B* is a part of *C* then *A* has birth place *C*". Listing 3 shows the encoding of an inferred triple along with the justification for its assertion by the *Academician Locator* application.

Listing 3: Encoding of the justification for an inferred assertion in the *Academician Locator* application

```
#graph for the inferred triple
aloc:t1 {
    AcadWiki:Bob AcadWiki:birthPlace GeoWiki:
        UnitedKingdom.
}
#graph justifying the assertion of the inferred
    triple
aloc:j1 {
    aloc:j1 rdf:type Justification.
    aloc:j1 r4ta:justifies aloc:t1.
    aloc:j1 r4ta:antecedent AcadWiki:j4.
    aloc:j1 r4ta:antecedent GeoWiki:j1.
    aloc:t1 rdf:type r4ta:InferredAssertion.
    aloc:t1 r4ta:inferredByRule aloc:pobRule.
}
```

The justification *aloc:j1* shows an example of linking distributed justifications. The triple *aloc:t1*, representing *AcadWiki:Bob* has birthplace *GeoWiki:UnitedKingdom*, is inferred from the triple *AcadWiki:t4* in *AcadWiki* which states that *AcadWiki:Bob* has birthplace *GeoWiki:London*, and the triple *GeoWiki:t1* in *GeoWiki* which states that *GeoWiki:London* is a part of *GeoWiki:UnitedKingdom*. The justification graph for the assertion of *aloc:t1* therefore includes an antecedent link to *AcadWiki:j4*, the justification for the assertion of *AcadWiki:t4*; and an antecedent link to *GeoWiki:j1*, the justification for the assertion of *GeoWiki:t1*. Note that the justifications *AcadWiki:j4* and *GeoWiki:j1* are generated and located in different locations than the *Academician Locator* application. This shows how our encoding allows linking distributed justifications.

5. RELATED WORK

Horridge *et al.* present two fine-grained subclasses of justifications called laconic justifications and precise justifications [8]. Laconic justifications are the justifications whose axioms do not contain any superfluous parts. Precise justifications are derived from laconic justifications and each of whose axioms represents a minimal part of the justification. The authors also present an optimised algorithm to compute laconic justifications showing the feasibility of computing laconic justifications and precise justifications in practice. In contrast to this work, we focus on a platform for justifications in a distributed environment. We do not focus on the theoretical aspects of the justifications such as the minimal parts of axioms in a justification which are required to hold an entailment. Rather, we focus on the aspects related to providing a platform for publishing and consuming justifications in a distributed environment.

McGuinness *et al.* [10] present an explanation interlingua called Proof Markup Language (PML). PML supports capturing provenance, information about information manipulation steps and trust. PML provides representational primitives for encoding conclusions, conclusion antecedents, and the information manipulation steps used to derive conclusions. As we pointed out earlier, PML uses containers to represent a set of antecedents. For this reason, the data described using PML contain blank nodes. This is a major drawback of PML with regard to our approach because we are completely avoiding blank nodes. In addition, we have a

narrower focus as we do not consider encoding manipulation steps, trust related information, or the generic provenance related information. Our focus is on the representation of inference dependencies between triples in form of justifications in a distributed environment.

Kotowski and Bry [9] argue that explanation complements the incremental development of knowledge bases in the frequently changing wiki environments. The authors present a semantic wiki called KiWi which takes a rule-based inconsistency tolerant reasoning approach that has the capability of explaining how a given piece of information was derived. The reasoning approach allows knowledge base updates in an efficient way by using reason maintenance. Justifications of all the derivations are stored and used for explanation and reason maintenance. In contrast to our work, Kotowski and Bry do not discuss publishing justifications in the Web for future reuse. Reason maintenance in a distributed environment is not discussed either. In our approach, we publish justifications following the Linked Data principles. The consumers can use these published justifications for providing explanation and enabling reason maintenance in a distributed environment.

Zhao *et al.* [17] discuss the management of biological data in terms of mapping links of data items from different data sources with the help of the provenance information of mapping links. Provenance information about the mapping links and about the changes in mapping links help providing reliable and accurate service. The authors provide design patterns to encode provenance information of the mapping links. The authors use named RDF graphs to represent the aspects of data provenance illustrating different levels of granularity of data provenance. The authors use HTTP URLs as identifiers of all data items, including the named graphs recoding provenance information, to accommodate federated queries over multiple datasets distributed across the Web. Our encoding approach is inspired by these design patterns. However, we do not consider a broader notion of provenance as Zhao *et al.* do. We focus on encoding justifications for the assertions of triples.

6. CONCLUSIONS AND FUTURE WORK

We have discussed the usefulness of linking justifications in the collaborative Semantic Web applications which have the capability of making inferences and publishing the inferred statements for future reuse. We have presented a motivating scenario and discussed our encoding approach with examples. Our proposed approach will enable the data consumers to determine the chains of the related statements from which a particular statement has been derived. These related statements can be distributed across the Web. Similarly, the justifications for the assertions of these statements can also be distributed across the Web. Our approach allows to explicitly link the related justifications in a distributed environment. This enables tracing the origin of inferences performed in a distributed environment in a simple manner.

The knowledge created by the collaborative applications are continuously changing and evolving. Therefore, the inferences that these applications make change and evolve. The chains of applications that are depended on these collaboratively created knowledge continuously need to update the

inferences they make. In such a scenario, it is important to allow tracing back the origins of inferences in order to understand if a given inferred statement is up-to-date with regard to all the changes. Furthermore, a user in a collaborative knowledge management application such as a semantic wiki might be more knowledgeable about the schema, the facts and the inferences in the semantic wiki in which he is working. The user might be less knowledgeable about such information from the external data sources. He might encounter pieces of information that have been inferred from the information from external data sources. In such a situation, he might not be able to completely understand how such information have come into existence. A user might need additional explanation about the history of such inference. Explanations generated from the linked justifications provide the required additional information about the history of inferences and assist the users in the collaborative knowledge management process.

An explanation generated from the distributed and linked justifications do not necessarily have to present the whole derivation chain together to a user at one instance. A user might follow a link to the part of an explanation in which he feels interested. In this way a user will be able to browse through an explanation generated from justifications distributed in different locations. Such explanations would be generated from abstractions of justifications on demand, and therefore, there will be no need to collect and integrate all the relevant justifications in one place.

Our immediate future work would be to develop the infrastructure required for publishing and linking justifications following our proposed approach. The next future work would be to generate human understandable explanations and to build the browsers to navigate through the explanation generated from the linked justifications allowing the follow-your-nose principle[8]. With regard to the generation of explanations, what kind of presentations of explanations are suitable for different kinds of users and how to transform the justifications into a suitable human understandable presentation is an important direction for the future work. Other future directions include the discussed truth maintenance platform and exploring how user trust can be supported from the associated justifications.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] Rdf semantics. *W3C recommendation*, Feb 2004.
[2] D. Beckett and T. Berners-Lee. Turtle - terse rdf triple language.
[3] T. Berners-Lee. Linked data - design issues.
[4] J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs, provenance and trust. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 613–622, New York, NY, USA, 2005. ACM.

---

[8]http://inkdroid.org/journal/2008/01/04/following-your-nose-to-the-web-of-data/

[5] O. Görlitz and S. Staab. Federated data management and query optimization for linked open data. In A. Vakali and L. C. Jain, editors, *New Directions in Web Data Management*, pages 109–137. Springer Verlag, 2011.

[6] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edition, 2011.

[7] P. Hitzler, M. Krötzsch, and S. Rudolph. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 2009.

[8] M. Horridge, B. Parsia, and U. Sattler. Laconic and precise justifications in owl. In *Proceedings of the 7th International Conference on The Semantic Web*, ISWC '08, pages 323–338, Berlin, Heidelberg, 2008. Springer-Verlag.

[9] J. Kotowski and F. Bry. A perfect match for reasoning, explanation and reason maintenance: Owl 2 rl and semantic wikis. In *Proceedings of 5th Semantic Wiki Workshop, Hersonissos, Crete, Greece (31st May 2010)*, 2010.

[10] D. McGuinness, L. Ding, P. Da Silva, and C. Chang. Pml 2: A modular explanation interlingua. In *In AAAI 2007 Workshop on Explanation-aware Computing*, 2007.

[11] D. L. McGuinness and P. P. da Silva. Explaining answers from the semantic web: the inference web approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(4):397 – 413, 2004. International Semantic Web Conference 2003.

[12] D. L. McGuinness, V. Furtado, P. Pinheiro da Silva, L. Ding, A. Glass, and C. Chang. Explaining semantic web applications. In *Semantic Web Engineering in the Knowledge Society*. 2008. Http://www.igi-global.com/reference/details.asp?id=8276.

[13] A. Passant. A collaborative semantic space for enterprise. In *Proceedings of the Knowledge Web PhD Symposium*, 2007.

[14] P. Pinheiro da Silva, D. L. McGuinness, N. Del Rio, and L. Ding. Inference web in action: Lightweight use of the proof markup language. In *Proceedings of the 7th International Semantic Web Conference (ISWC'08)*, pages 847–860, 2008.

[15] S. Schaffert, F. Bry, J. Baumeister, and M. Kiesel. Semantic wikis. *Software, IEEE*, 25(4):8 –11, july-aug. 2008.

[16] E. Watkins and D. Nicole. Named graphs as a mechanism for reasoning about provenance. In X. Zhou, J. Li, H. Shen, M. Kitsuregawa, and Y. Zhang, editors, *Frontiers of WWW Research and Development - APWeb 2006*, volume 3841 of *Lecture Notes in Computer Science*, pages 943–948. Springer Berlin / Heidelberg, 2006.

[17] J. Zhao, A. Miles, G. Klyne, and D. Shotton. Linked data and provenance in biological data webs. *Briefings in bioinformatics*, 10(2):139–152, 2009.