SWPM 2010

Workshop on Semantic Web and Provenance Management

The Second International Workshop on the role of Semantic Web in Provenance Management

(at the 9th International Semantic Web Conference ISWC-2010)

November 07 2010, Shanghai International Convention Center, Shanghai, China.

Organization

Chairs

Amit Sheth
Juliana Freire

Organizing Committee/PC Co-Chairs

Satya S. Sahoo
Jun Zhao
Paolo Missier
Jose Manuel Gómez-Pérez

Program Committee

Alexander Passant, DERI, NUI Galway
Beth Plale, Indiana University
Chris Bizer, Freie Universität Berlin
Christine Runnegar, Internet Society (ISOC)
Deborah McGuinness, RPI
GQ Zhang, Case Western Reserve University
Ilkay Altintas, San Diego Supercomputer Center, UCSD
Irini Fundulaki, ICS Foundation for Research and Technology, Greece
James Cheney, University of Edinburgh
James Myers, NCSA
Kei Cheung, Yale University
Krishnaprasad Thirunarayan, Kno.e.sis Center, Wright State University
Nirmal Mukhi, IBM Research
Olaf Hartig, Humboldt Universität zu Berlin
Olivier Bodenreider, National Library of Medicine, NIH
Paul Groth, VU University, Netherlands
Paulo Pinheiro da Silva, University of Texas at El Paso
Roger Barga, Microsoft Research
Sam Coppens, Ghent University
Sarah Cohen-Boulakia, Universite Paris-Sud
Simon Miles, King's College London

Sudha Ram, Arizona State University
Yolanda Gil, Information Sciences Institute, USC
Yogesh Simmhan, Microsoft Research

© Copyright remains with the authors, and permission to reproduce material printed as SWPM10 workshop proceedings should be sought from them. Similarly, pursuing copyright infringements, plagiarism, etc. remains the responsibility of authors.

Introduction

The growing eScience infrastructure is enabling scientists to generate scientific data on an industrial scale. Similarly, using the Web as the platform, the Linked Open Data (LOD) initiative has created a vast amount of information that can be leveraged by Semantic Web application in a variety of real world scenarios. The importance of managing various forms of metadata has long been recognized as critical in the Semantic Web. In this workshop we focus specifically on metadata that describes the origins of the data. The term provenance from the French word "provenir", meaning "to come from", describes the lineage or origins of a data entity. Provenance metadata is essential to correctly interpret the results of a process execution, to validate data processing tools, to verify the quality of data, and to associate measures of trust to the data. The primary objective of this workshop is two-fold, (1) to explore the role of Semantic Web in addressing some of the critical challenges facing provenance management and (2) the role of provenance in real world Semantic Web applications. Specifically,

- Efficiently capturing and propagating provenance information as data is processed, fragmented and recombined across multiple applications on a Web scale, for example in the LOD cloud.
- A common representation model or vocabulary for provenance for processing and analysis by both agents and humans.
- Interoperability of provenance information generated in distributed environments.
- Tools leveraging the Semantic Web for visualization of provenance information.

We thank the keynote speakers, all members of the program committee, authors, invited speakers, participants and local organizers for their efforts.

We look forward to a successful workshop!

Satya S. Sahoo, Jun Zhao, Paolo Missier, Jose Manuel Gómez-Pérez

Annotation algebras for RDFS

Peter Buneman
University of Edinburgh
Email: opb@inf.ed.ac.uk

Egor V. Kostylev
University of Edinburgh
Email: ekostyle@inf.ed.ac.uk

***Abstract*—Provenance and annotation are intimately connected. On the one hand provenance can be regarded as a form of annotation, on the other hand, in order to understand how to convey annotations from source data to derived data, we need an account of *how* the data was derived – its provenance. Following a successful line of database research in which elements of a database are annotated with algebraic terms that describe the provenance of those elements, we develop an algebra of annotations for RDFS that differs from that developed for relational databases. We show how such an annotation algebra can be used for computing annotations on inferred triples that provide information on belief, trust and temporal aspects of data as well as providing a framework for default reasoning.**

I. Introduction

With the increasing interest in provenance in both databases and ontologies, there have been a number of proposals and systems for annotation of the underlying data with information concerning time, belief, and various aspects of provenance. The question that has been repeatedly posed in databases [17], [5], [2] is what happens to these annotations when a query is applied? Are they ignored or are they somehow passed through the query? In fact generic prototypes [4] and systems that are specific to some domain [6] have been developed for propagating annotations through queries. Suppose, for example, we take two tables S and T in which the individual tuples have been annotated. How should we annotate the tuples in the join of S and T? An obvious answer is to put on any output tuple the union of the annotations on the two contributing tuples. This does not always make sense; for example if the input tuples are annotated with the set of people who believe that this tuple is true or with the set of database versions for which the tuple is actually in the database, it might be more appropriate to annotate a join tuple with the intersection of the relevant annotations.

This problem has resulted in a variety of proposals for propagating annotations through queries. Notably, by using a semiring model for annotations, in [8] a tuple is annotated with a term in a semiring algebra that describes the provenance of the tuple – how it was formed by the operations of the relational query that constructed it. By suitable instatiatons operations of the semiring, one can realize various extensions to relational databases such as probabilistic databases, multiset semantics and certain kinds of constraint databases. The semiring model also generalizes a number of other models for provenance [5], [2].

Turning to ontologies, proposals have also been suggested for the annotation of RDF [13]. Named graphs [3] and temporal RDF [10] propose methods of adding annotations to RDF triples to express belief, trust, or temporal properties. Can we simply follow the work for relational databases in developing a general model for such annotations? Here we have to start by looking not at query languages for RDF, but at the inference rules for the ontologies such as those of RDFS [1]. Given annotations on the base triples, what should be the annotations on an inferred triple? Indeed in [15], [16], [12], with an initial goal applying fuzzy logic to RDFS proposes an algebra similar in many respects to that of [8]. In this paper we propose a somewhat more general – and possibly simpler – algebra to serve this purpose. Our proposals differ from the semirings in [8] in two ways. First, there are situations in which we do not want commutativity and second, while the number of triples inferred in an RDFS graph is always finite, the derivations can be unbounded. We therefore need an extra condition to prevent "infinite annotations". This condition precludes the possibility of bag semantics, which is useful for relational algbra but appears to be inapplicable to ontologies. Also in [8] there is a compact representation – a polynomial – of terms in the semiring algebra. In the algebra we develop, the compact representation is somewhat different.

To introduce annotation algebras, we give two simple examples of annotating an RDF graph shown in Fig. 1(a) (a dashed arrow represents a triple, which can be inferred from the rest of the graph).

Example 1: A temporal extension of RDF was introduced in [10], [9]. It consists of attaching to every RDF triple a set intervals that represents the times at which the triple is valid. An example of a temporal annotation of the graph is shown in Fig. 1(b): Picasso worked as a cubist from 1908 to 1919, paints are created by painters at least since engravings in Chauvet-Pont-d'Arc cave from about 29,000BC, and so on. The point here is that an annotation for the inferred triple was obtained by an intuitive calculation $(\{1908-1919\}\cap\{1906-1921\})\cup(\{1937\}\cap\{-29,000-Now\}) = \{1908-1919, 1937\}$.

Example 2: In [15] fuzzy annotations of RDF are used to describe degree of trust. To every triple a real in the range $[0,1]$ is attached. Such an annotation is shown in Fig. 1(c). The annotation for the inferred triple was calculated as $max(0.8*0.4, 0.3*1) = 0.32$.

There is an obvious similarity between these examples: the calculations are both of the general form $(a \otimes b) \oplus (c \otimes d)$, and the calculations for the inferred triple are performed by suitably instantiating the operators $\oplus$ and $\otimes$. In fact [16], both of the annotation domains form so-called *BL-algebras*

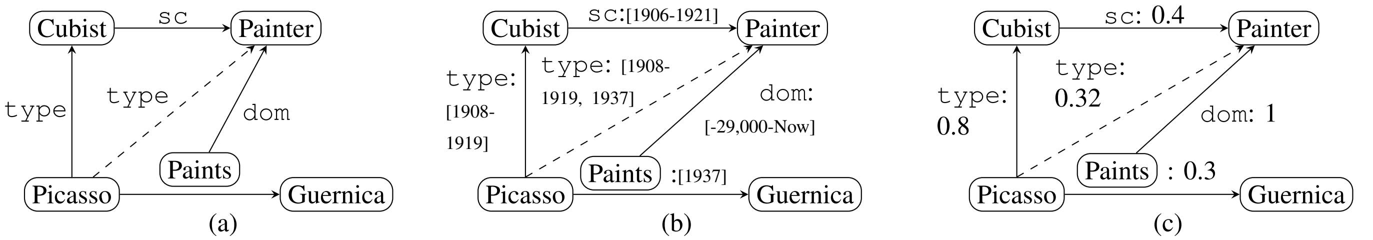


Fig. 1. Standard and annotated RDF graphs

[11], which in general preserve the properties of the deductive system [16].

Before describing these constructions in more detail we should briefly connect this work with the general issue of provenance. Our first observation is that an algebraic annotation or a triple of the form we have described is a synopsis of how that triple was derived – which is surely part of its provenance. The second is that exogenous provenance information such as who created a triple or when it was created can be added following the proposals of [3]. We would also like to compute such annotations for an inferred triple. Our proposals provide a method for tranferring the provenance annotations of explict triples to those that are derived.

Outline. This work has two aims. The first is to determine what algebraic structure is necessary for an annotation domain to keep the behavior of the deductive system the same as in the standard case. The second is to find "the most general" of such structures, which allows annotation of RDF graphs by elements of the general structure, apply inference rules, and then obtain annotations from specific domains on demand. In the following sections we review RDFS, introduce a new annotation algebra and provide some evidence that this is the appropriate algebra for RDFS. We give a freeness result that allows us to represent the terms of this annotation algebra and give some examples of the use of the algebra.

II. Preliminaries

Given a set of *RDF URI references* $\mathbf{U}$[1], let T be the set of *RDF triples* of the form $(s, p, o) \in \mathbf{U} \times \mathbf{U} \times \mathbf{U}$. Here s, p and o are called *subject*, *predicate* and *object* correspondingly. An *RDF graph* (or simply *graph*) is a finite set of triples $G \subseteq T$.

The RDF specification[13] includes RDF Schema (RDFS) [1] which is a vocabulary of reserved words designed to describe relationships between resources and properties. In this work we use the $\rho df = \{\mathtt{sp}, \mathtt{sc}, \mathtt{type}, \mathtt{dom}, \mathtt{range}\}$ fragment of RDFS [14]. The elements of ρdf represent sub-property, sub-class, domain, and range properties correspondingly. It is widely accepted that ρdf is a stable core of RDFS.

An *interpretation* of an RDF graph is a tuple $\mathcal{I} = (\Delta_R, \Delta_P, \Delta_C, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^I)$ where

- Δ_R is a nonempty set of *resources*,
- Δ_P is a set of *property names* (not necessarily disjoint from Δ_R),
- $\Delta_C \subseteq \Delta_R$ is a set of *classes*,
- $P[\![\cdot]\!] : \Delta_P \to 2^{\Delta_R \times \Delta_R}$ is a *property extension* function,
- $C[\![\cdot]\!] : \Delta_C \to 2^{\Delta_R}$ is a *class extension* function,
- $\cdot^I : \mathbf{U} \to \Delta_R \cup \Delta_P$ is an *interpretation mapping*.

The interpretation $\mathcal{I}$ is a *model* for a graph G over ρdf, denoted by $\mathcal{I} \models G$, iff the conditions in Tab. I hold[2]. A graph G *entails* G', denoted $G \models G'$, if every model of G is a model of G'.

A sound and complete deductive system for entailment [14] is presented in Tab. II. An *instantiation* of an inference rule r from the system is a replacement of the variables $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{X}$, and $\mathcal{Y}$, occurring in the rule, by references from $\mathbf{U}$. If there is an instantiation $\frac{R}{R'}$ of r such that $R \subseteq G$, then the graph $G' = G \cup R'$ is the result of an *application* of r to G. A graph G' is *inferred* from G, denoted by $G \vdash G'$, iff G' is obtained from G by successively applying rules in Tab. II. This entailment can be checked by computing the *closure* of G, which is the maximal graph which can be inferred from G. It can be done in quadratic time [14].

III. Annotated RDFS

Definition 1: Given an algebra $\mathcal{K}$ with an elements set K containing a distinguished element $\perp$ a $\mathcal{K}$*-annotated RDF graph* (or simply an *a-graph*, if $\mathcal{K}$ is clear) is a function $G : T \to 2^K$ such that for each $t \in T$ holds $\perp \in G(t)$ and the set $Supp(G) = \{t : v \mid v \in G(t), v \neq \perp\}$ is finite.

If $v \in G(t)$ we write $t : v \in G$ and call it an *a-triple*. An a-graph G is *schema-acyclic*, if the subgraphs $\{(s, \mathtt{e}, o) : v \mid (s, \mathtt{e}, o) : v \in Supp(G)\}, \mathtt{e} = \mathtt{sc}, \mathtt{sp}$, do not contain non-trivial loops. The semantics for a-graphs is given in the following definitions.

Definition 2: A $\mathcal{K}$*-annotated interpretation* is a tuple $\mathcal{I} = (\Delta_R, \Delta_P, \Delta_C, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^I)$ where $\Delta_R, \Delta_P, \Delta_C$ and $\cdot^I$ are the same as for the standard interpretation and $P[\![\cdot]\!], C[\![\cdot]\!]$ are defined as follows:

- for each $p \in \Delta_P$ holds $P[\![p]\!] : \Delta_R \times \Delta_R \to K$,
- for each $c \in \Delta_C$ holds $C[\![c]\!] : \Delta_R \to K$.

To define models for annotated RDFS we need more structure on the algebra $\mathcal{K}$. Let $\oplus$ and $\otimes$ be binary operations on K and $\perp$, $\top$ be distinct constants in it. For every $a, b \in K$

[1] For the sake of simplicity we do not consider blank nodes and literals. Their inclusion does not change the results of this work.

[2] The form of conditions in our definition of model is slightly different from that in [14], but they are equivalent *per se*. It is done to simplify the comparison with notion of model for annotated RDFS given in Sect. III.

(1) Simple interpretation:
- for each $(s, p, o) \in G$ holds $p^I \in \Delta_P$ and $(s^I, o^I) \in P[\![p^I]\!]$.

(2) Properties and classes:
- for each $\mathtt{e} \in \rho df$ holds $\mathtt{e}^I \in \Delta_P$;
- if $(x, y) \in P[\![\mathtt{sp}^I]\!]$ then $x, y \in \Delta_P$;
- if $(x, y) \in P[\![\mathtt{sc}^I]\!]$ then $x, y \in \Delta_C$;
- if $(x, y) \in P[\![\mathtt{type}^I]\!]$ then $y \in \Delta_C$;
- if $(x, y) \in P[\![\mathtt{dom}^I]\!]$ then $x \in \Delta_C$ and $y \in \Delta_P$;
- if $(x, y) \in P[\![\mathtt{range}^I]\!]$ then $x \in \Delta_C$ and $y \in \Delta_P$.

(3) Sub-property:
- $(p, p) \in P[\![\mathtt{sp}^I]\!]$;
- if $(p, q), (q, r) \in P[\![\mathtt{sp}^I]\!]$ then $(p, r) \in P[\![\mathtt{sp}^I]\!]$;
- if $(x, y) \in P[\![p]\!]$ and $(p, q) \in P[\![\mathtt{sp}^I]\!]$ then $(x, y) \in P[\![q]\!]$.

(4) Sub-class:
- $(c, c) \in P[\![\mathtt{sc}^I]\!]$;
- if $(c, d), (d, e) \in P[\![\mathtt{sc}^I]\!]$ then $(c, e) \in P[\![\mathtt{sc}^I]\!]$;
- if $x \in C[\![c]\!]$ and $(c, d) \in P[\![\mathtt{sc}^I]\!]$ then $x \in C[\![d]\!]$.

(5) Typing:
- $(x, c) \in P[\![\mathtt{type}^I]\!]$ iff $x \in C[\![c]\!]$;
- if $(x, y) \in P[\![p]\!]$ and $(c, p) \in P[\![\mathtt{dom}^I]\!]$ then $x \in C[\![c]\!]$;
- if $(x, y) \in P[\![p]\!]$ and $(c, p) \in P[\![\mathtt{range}^I]\!]$ then $y \in C[\![c]\!]$.

TABLE I
RDFS SEMANTICS

(1) Sub-property:

(a) $\dfrac{(\mathcal{A}, \mathtt{sp}, \mathcal{B})\ (\mathcal{B}, \mathtt{sp}, \mathcal{C})}{(\mathcal{A}, \mathtt{sp}, \mathcal{C})}$;

(b) $\dfrac{(\mathcal{X}, \mathcal{A}, \mathcal{Y})\ (\mathcal{A}, \mathtt{sp}, \mathcal{B})}{(\mathcal{X}, \mathcal{B}, \mathcal{Y})}$.

(2) Sub-class:

(a) $\dfrac{(\mathcal{A}, \mathtt{sc}, \mathcal{B})\ (\mathcal{B}, \mathtt{sc}, \mathcal{C})}{(\mathcal{A}, \mathtt{sc}, \mathcal{C})}$;

(b) $\dfrac{(\mathcal{X}, \mathtt{type}, \mathcal{A})\ (\mathcal{A}, \mathtt{sc}, \mathcal{B})}{(\mathcal{X}, \mathtt{type}, \mathcal{B})}$.

(3) Typing:

(a) $\dfrac{(\mathcal{X}, \mathcal{A}, \mathcal{Y})\ (\mathcal{A}, \mathtt{dom}, \mathcal{B})}{(\mathcal{X}, \mathtt{type}, \mathcal{B})}$;

(b) $\dfrac{(\mathcal{X}, \mathcal{A}, \mathcal{Y})\ (\mathcal{A}, \mathtt{range}, \mathcal{B})}{(\mathcal{Y}, \mathtt{type}, \mathcal{B})}$.

(4) Sub-class Reflexivity:

(a) $\dfrac{(\mathcal{A}, \mathtt{sc}, \mathcal{B})}{(\mathcal{A}, \mathtt{sc}, \mathcal{A})\ (\mathcal{B}, \mathtt{sc}, \mathcal{B})}$;

(b) $\dfrac{(\mathcal{X}, \mathtt{e}, \mathcal{A})}{(\mathcal{A}, \mathtt{sc}, \mathcal{A})}$ for $\mathtt{e} \in \{\mathtt{dom}, \mathtt{range}, \mathtt{type}\}$.

(5) Sub-property Reflexivity:

(a) $\dfrac{(\mathcal{X}, \mathcal{A}, \mathcal{Y})}{(\mathcal{A}, \mathtt{sp}, \mathcal{A})}$;

(b) $\dfrac{}{(\mathtt{e}, \mathtt{sp}, \mathtt{e})}$ for $\mathtt{e} \in \rho df$;

(c) $\dfrac{(\mathcal{A}, \mathtt{sp}, \mathcal{B})}{(\mathcal{A}, \mathtt{sp}, \mathcal{A})\ (\mathcal{B}, \mathtt{sp}, \mathcal{B})}$;

(d) $\dfrac{(\mathcal{A}, \mathtt{e}, \mathcal{X})}{(\mathcal{A}, \mathtt{sp}, \mathcal{A})}$ for $\mathtt{e} \in \{\mathtt{dom}, \mathtt{range}\}$.

TABLE II
RDFS DEDUCTIVE SYSTEM

we write $a \preceq b$ iff there exists $c \in K$ such that $a \oplus c = b$. The addition operation $\oplus$ will be used to combine annotations for the same triple and the $\bot$ constant represents the fact, that there is no information about a triple. The product operation $\otimes$ will be used to join annotations when applying inference rules and $\top$ represents the maximal annotation.

Definition 3: Let $\mathcal{K} = \langle K, \oplus, \otimes, \bot, \top \rangle$ be an algebra of type $(2, 2, 0, 0)$. The $\mathcal{K}$-annotated interpretation $\mathcal{I}$ is a *model* for an a-graph G, denoted $\mathcal{I} \models G$, iff the conditions in Tab. III hold. An a-graph G *entails* H, denoted $G \models H$, if for every $\mathcal{I} \models G$ holds $\mathcal{I} \models H$.

By these definitions, each (non-annotated) RDF graph G can be considered as an a-graph $G' = \{t : \top \mid t \in G\} \cup E$, if $K = \{\bot, \top\}$ and $E = \{t : \bot \mid t \in T\}$. In this case, the definition of model for an a-graph coincides with the standard.

IV. A DEDUCTIVE SYSTEM FOR ANNOTATED RDFS AND DIOIDS

Definition 4: An algebra $\mathcal{K} = \langle K, \oplus, \otimes, \bot, \top \rangle$ is a *dioid* iff it is an idempotent semi-ring, i.e.

(1) $\langle K, \oplus, \bot \rangle$ is a semilattice, i.e. for each a, b, and c hold: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ (associativity), $a \oplus b = b \oplus a$, (commutativity), $a \oplus \bot = a$ (neutral element), $a \oplus a = a$ (idempotence);

(2) $\langle K, \otimes, \top \rangle$ is a monoid, i.e. for each a, b, and c hold: $(a \otimes b) \otimes c = a \otimes (b \otimes c)$ (associativity), $a \otimes \top = a = \top \otimes a$ (neutral element);

(3) $\otimes$ is left and right distributive over $\oplus$, i.e. for each a, b, and c hold: $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$, $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$;

(4) $\otimes$ is $\bot$-annihilating, i.e. for each a holds $\bot \otimes a = \bot = a \otimes \bot$.

A dioid is $\top$-*dioid* iff

(5) $\oplus$ is $\top$-*annihilating*, i.e. for each a holds $\top \oplus a = \top$.

Note, that $\top$-annihilation entails idempotence from (1).

An *instantiation* of a rule from the deductive system in Tab. IV is a replacement of variables $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{X}$, and Y by elements of $\mathbf{U}$, and variables v, v_1, v_2, and v_3 by elements of K, such that all relations for annotations hold. An *application* of a rule to an a-graph G and a *deduction* of an a-graph G' from G, denoted by $G \vdash G'$, is defined exactly the same way as for the standard case.

Note, that the system in Tab. IV differs from the one in Tab. II only by the presence of annotations and the generalisation rule $(*)$ which combines annotations for the same triple. Thus, that is natural to expect the new system to behave the same as the standard one. Particularly, they should coincide if $K = \{\bot, \top\}$. To obtain it we need some properties of $\mathcal{K}$.

Definition 5: Let $\mathcal{R}$ be the set of all inference rules of the form $\frac{\mathcal{T}_1\ \mathcal{T}_2}{\mathcal{T}}$ from Tab. IV and $Ins(r)$ the set of all instantiations of a rule $r \in \mathcal{R}$.

(1) A set of rules $\mathcal{R}' \subseteq \mathcal{R}$ is *associative*, iff for every $r, r' \in \mathcal{R}'$, $\frac{\tau_1\ \tau_2}{\tau_4}, \frac{\tau_1\ \tau_4'}{\tau_5'} \in Ins(r)$, and $\frac{\tau_4\ \tau_3}{\tau_5}, \frac{\tau_2\ \tau_3}{\tau_4'} \in Ins(r')$ holds $\tau_5 = \tau_5'$.

(2) A rule $r \in \mathcal{R}$ is *commutative*, iff for every $\frac{\tau_1\ \tau_2}{\tau} \in Ins(r)$ holds $\frac{\tau_2\ \tau_1}{\tau} \in Ins(r)$.

(3) A rule $r \in \mathcal{R}$ is *idempotent*, if $\frac{\tau\ \tau}{\tau} \in Ins(r)$ for every τ.

(4) A set of rules $\mathcal{R}' \subseteq \mathcal{R}$ is *left distributive* over $r \in \mathcal{R}$ if for every $r' \in \mathcal{R}'$, $\frac{\tau_2\ \tau_3}{\tau_4}, \frac{\tau_4'\ \tau_4''}{\tau_5'} \in Ins(r)$, and

(1) Simple interpretation:
- for each $(s, p, o) : v \in G$ holds $p^I \in \Delta_P$ and $v \preceq P[\![p^I]\!](s^I, o^I)$.

(2) Properties and classes:
- for each $\mathtt{e} \in \rho df$ holds $\mathtt{e}^I \in \Delta_P$;
- $P[\![\mathtt{sp}^I]\!](x, y)$ is defined only for $x, y \in \Delta_P$;
- $P[\![\mathtt{sc}^I]\!](x, y)$ is defined only for $x, y \in \Delta_C$;
- $P[\![\mathtt{type}^I]\!](x, y)$ is defined only for $y \in \Delta_C$;
- $P[\![\mathtt{dom}^I]\!](x, y)$ is defined only for $x \in \Delta_C$ and $y \in \Delta_P$;
- $P[\![\mathtt{range}^I]\!](x, y)$ is defined only for $x \in \Delta_C$ and $y \in \Delta_P$.

(3) Sub-property:
- $P[\![\mathtt{sp}^I]\!](p, p) = \top$,
- $P[\![\mathtt{sp}^I]\!](p, q) \otimes P[\![\mathtt{sp}^I]\!](q, r) \preceq P[\![\mathtt{sp}^I]\!](p, r)$;
- $P[\![p]\!](x, y) \otimes P[\![\mathtt{sp}^I]\!](p, q) \preceq P[\![q]\!](x, y)$.

(4) Sub-class:
- $P[\![\mathtt{sc}^I]\!](c, c) = \top$,
- $P[\![\mathtt{sc}^I]\!](c, d) \otimes P[\![\mathtt{sc}^I]\!](d, e) \preceq P[\![\mathtt{sc}^I]\!](c, e)$;
- $C[\![c]\!](x) \otimes P[\![\mathtt{sc}^I]\!](c, d) \preceq C[\![d]\!](x)$.

(5) Typing:
- $P[\![\mathtt{type}^I]\!](x, c) = C[\![c]\!](x)$;
- $P[\![p]\!](x, y) \otimes P[\![\mathtt{dom}^I]\!](c, p) \preceq C[\![c]\!](x)$;
- $P[\![p]\!](x, y) \otimes P[\![\mathtt{range}^I]\!](c, p) \preceq C[\![c]\!](y)$.

TABLE III
ANNOTATED RDFS SEMANTICS

(1) Sub-property:

$$\text{(a)}\ \frac{(\mathcal{A}, \mathtt{sp}, \mathcal{B}) : v_1\ (\mathcal{B}, \mathtt{sp}, \mathcal{C}) : v_2}{(\mathcal{A}, \mathtt{sp}, \mathcal{C}) : v_1 \otimes v_2}; \qquad \text{(b)}\ \frac{(\mathcal{X}, \mathcal{A}, \mathcal{Y}) : v_1\ (\mathcal{A}, \mathtt{sp}, \mathcal{B}) : v_2}{(\mathcal{X}, \mathcal{B}, \mathcal{Y}) : v_1 \otimes v_2}.$$

(2) Sub-class:

$$\text{(a)}\ \frac{(\mathcal{A}, \mathtt{sc}, \mathcal{B}) : v_1\ (\mathcal{B}, \mathtt{sc}, \mathcal{C}) : v_2}{(\mathcal{A}, \mathtt{sc}, \mathcal{C}) : v_1 \otimes v_2}; \qquad \text{(b)}\ \frac{(\mathcal{X}, \mathtt{type}, \mathcal{A}) : v_1\ (\mathcal{A}, \mathtt{sc}, \mathcal{B}) : v_2}{(\mathcal{X}, \mathtt{type}, \mathcal{B}) : v_1 \otimes v_2}.$$

(3) Typing:

$$\text{(a)}\ \frac{(\mathcal{X}, \mathcal{A}, \mathcal{Y}) : v_1\ (\mathcal{A}, \mathtt{dom}, \mathcal{B}) : v_2}{(\mathcal{X}, \mathtt{type}, \mathcal{B}) : v_1 \otimes v_2}; \qquad \text{(b)}\ \frac{(\mathcal{X}, \mathcal{A}, \mathcal{Y}) : v_1\ (\mathcal{A}, \mathtt{range}, \mathcal{B}) : v_2}{(\mathcal{Y}, \mathtt{type}, \mathcal{B}) : v_1 \otimes v_2}.$$

(*) Generalisation:

$$\frac{(\mathcal{X}, \mathcal{A}, \mathcal{Y}) : v_1\ (\mathcal{X}, \mathcal{A}, \mathcal{Y}) : v_2}{(\mathcal{X}, \mathcal{A}, \mathcal{Y}) : v_1 \oplus v_2}.$$

(4) Sub-class Reflexivity:

$$\text{(a)}\ \frac{(\mathcal{A}, \mathtt{sc}, \mathcal{B}) : v}{(\mathcal{A}, \mathtt{sc}, \mathcal{A}) : v\ (\mathcal{B}, \mathtt{sc}, \mathcal{B}) : v}; \qquad \text{(b)}\ \frac{(\mathcal{X}, \mathtt{e}, \mathcal{A})}{(\mathcal{A}, \mathtt{sc}, \mathcal{A})} \quad \text{for } \mathtt{e} \in \{\mathtt{dom}, \mathtt{range}, \mathtt{type}\}.$$

(5) Sub-property Reflexivity:

$$\text{(a)}\ \frac{(\mathcal{X}, \mathcal{A}, \mathcal{Y}) : v}{(\mathcal{A}, \mathtt{sp}, \mathcal{A}) : v}; \quad \text{(b)}\ \frac{}{(\mathtt{e}, \mathtt{sp}, \mathtt{e}) : \top} \quad \text{for } \mathtt{e} \in \rho df; \quad \text{(c)}\ \frac{(\mathcal{A}, \mathtt{sp}, \mathcal{B}) : v}{(\mathcal{A}, \mathtt{sp}, \mathcal{A}) : v\ (\mathcal{B}, \mathtt{sp}, \mathcal{B}) : v}; \quad \text{(d)}\ \frac{(\mathcal{A}, \mathtt{e}, \mathcal{X}) : v}{(\mathcal{A}, \mathtt{sp}, \mathcal{A}) : v} \quad \text{for } \mathtt{e} \in \{\mathtt{dom}, \mathtt{range}\}.$$

TABLE IV
ANNOTATED RDFS DEDUCTIVE SYSTEM

$\frac{\tau_1\ \tau_4}{\tau_5}, \frac{\tau_1\ \tau_2}{\tau_4'}, \frac{\tau_1\ \tau_3}{\tau_4''} \in Ins(r')$ holds $\tau_5 = \tau_5'$.

(5) A set of rules $\mathcal{R}' \subseteq \mathcal{R}$ is *right distributive* over $r \in \mathcal{R}$ if for every $r' \in \mathcal{R}'$, $\frac{\tau_1\ \tau_2}{\tau_4}, \frac{\tau_4'\ \tau_4''}{\tau_5'} \in Ins(r)$, and $\frac{\tau_4\ \tau_3}{\tau_5}, \frac{\tau_1\ \tau_3}{\tau_4'}, \frac{\tau_2\ \tau_3}{\tau_4''} \in Ins(r')$ holds $\tau_5 = \tau_5'$.

(6) A rule $r \in \mathcal{R}$ is *v-neutral*, $v \in K$, if for every $\frac{t_1 : v\ t_2 : v_2}{t_3 : v_3} \in Ins(r)$ holds $v_2 = v_3$ and for every $\frac{t_1 : v_1\ t_2 : v}{t_3 : v_3} \in Ins(r)$ holds $v_1 = v_3$.

(7) A rule $r \in \mathcal{R}$ is *v-annihilating*, $v \in K$, if for every $\frac{t_1 : v\ t_2 : v_2}{t_3 : v_3} \in Ins(r)$ holds $v_3 = v$ and for every $\frac{t_1 : v_1\ t_2 : v}{t_3 : v_3} \in Ins(r)$ holds $v_3 = v$.

Proposition 1: The set of inference rules $\mathcal{R}' = \{(1a), (1b), (2a), (2b), (3a), (3b)\}$ in Tab. IV is associative, the set of rules $\{(8)\}$ is associative, the rule (8) is commutative, idempotent and $\bot$-neutral, the set $\mathcal{R}'$ is left and right distributive over the rule (8), and each of the rules from $\mathcal{R}'$ are $\top$-neutral and $\bot$-annihilating iff $\mathcal{K}$ is a dioid.

Theorem 1 (Soundness and completeness): Given a dioid $\mathcal{K}$ and a-graphs G and H hold.

(1) If $G \vdash H$ then $G \models H$.
(2) If G is schema-acyclic and $G \models H$ then for every $t : v \in H$ there exists $v' \succeq v$ such that $G \vdash t : v'$.
(3) If $\mathcal{K}$ is $\top$-annihilating and $G \models H$ then for every $t : v \in H$ there exists $v' \succeq v$ such that $G \vdash t : v'$.

Hence, the deductive system behaves the same as the standard one iff $\mathcal{K}$ is a dioid for schema-acyclic a-graphs and a $\top$-dioid in general case.

As in the standard case, the important notion is the *closure* $cl(G) = \{\tau \mid G \vdash \{\tau\}\}$ of an a-graph G. To compute it we need a *representation* of an a-graph G, which is a finite set of a-triples R_G such that $R_G \cup E = G$, $E = \{t : \bot \mid t \in T\}$. The set $Supp(G)$ by the definition is a representation of G, but we also want to have a possibility to work with representations which contain an finite number of $\bot$-annotated triples.

Proposition 2: Let G be an a-graph and $\mathcal{K}$ a $\top$-dioid. For every representation R_G of G there exists an representation $R_{cl(G)}$ of $cl(G)$ which can be computed in polynomial time in the size of R_G if the complexities of $\oplus$ and $\otimes$ are polynomial bounded.

Let $\mathcal{K}$ and $\mathcal{K}'$ be two algebras. For any function $h : \mathcal{K} \to \mathcal{K}'$ and $\mathcal{K}$-annotated graph G denote $h(G)$ the set of $\mathcal{K}'$-annotated triples formed from G by applying h to each annotation.

Proposition 3: Let $h : \mathcal{K} \to \mathcal{K}'$ and $\mathcal{K}, \mathcal{K}'$ be $\top$-dioids. For every $\mathcal{K}$-annotated graph G the set $h(G)$ is a $\mathcal{K}'$-annotated graph and holds $cl(h(G)) = h(cl(G))$ iff h is a dioid homomorphism.

V. STRING DIOIDS FOR RDFS ANNOTATION

Prop. 3 enables us to obtain an a-graph from another one without recomputing annotations for inferred triples. The next step is to develop a "universal" annotation, i.e. an annotation from which we can obtain any other one by applying a corresponding dioid homomorphism.

Given an alphabet Σ define an *subsequence order* on the set of words Σ^*: $u \le u'$ iff for some $u_1, \ldots, u_n, w_1, \ldots w_{n-1} \in \Sigma^*$ holds $u = u_1 u_2 \ldots u_n$ and $u' = u_1 w_1 u_2 w_2 \ldots w_{n-1} u_n$. A finite set $m \subseteq \Sigma^*$ is an *antichain* if for all $u \le u'$, $u, u' \in m$, holds $u = u'$. Let $\min(m)$ be the set of minimal elements (w.r.t. $\le$) of m. On the set of antichains $M[\Sigma]$ we define:

$$m_1 + m_2 = \min(m_1 \cup m_2),$$
$$m_1 \times m_2 = \min(\{w_1 w_2 \mid w_1 \in m_1, w_2 \in m_2\}),$$

Call $\mathcal{M}[\Sigma] = \langle M[\Sigma], +, \times, \emptyset, \{\epsilon\}\rangle$, where ϵ is the empty string, a *string dioid over generators* Σ. Note, that a string dioid is a $\top$-dioid. The following proposition says, that it is "the most general" of all $\top$-dioids.

Proposition 4: (1) Given a set of generators Σ the string dioid $\mathcal{M}[\Sigma]$ is the free $\top$-dioid on Σ, i.e. for any $\top$-dioid $\mathcal{K}_f = \langle K, \oplus, \otimes, \bot, \top\rangle$ and a valuation $\phi : \Sigma \to K$ there exists a unique homomorphism $\mathit{Eval}_\phi : M[\Sigma] \to K$ such that for each $a \in \Sigma$ holds $\mathit{Eval}_\phi(a) = \phi(a)$.
(2) The operations of the string dioid $+$ and $\times$ can be computed in polynomial time.

Hence, string dioids consititute an important and general subclass of $\top$-dioids. We now show how they can be applied to annotate RDF graphs. Let G be a $\mathcal{K}$-annotated graph and X a set of triple ids of triples from $\mathit{Supp}(G)$. We associate to G an "abstract" version which is a $(X \cup \emptyset)$-annotated graph $\bar{G}$ consisted of the same triples as G, but the annotation of each of them is its id if it has one, and $\emptyset$ otherwise.

Theorem 2: For every $\mathcal{K}$-annotated graph G holds $cl(G) = \mathit{Eval}_\phi \circ cl(\bar{G})$, where $\phi : X \to K$ is a valuation which associates the annotation of an a-triple in G to its id.

This theorem gives rise to the following strategy for annotating RDF graphs. Given a graph G we are to annotate it with elements from several domains. However, we would like to infer triples and their annotations only once, without recomputing the annotations for each annotation domain. In this case we construct an "abstract" version $\bar{G}$ of G by annotating triples with their ids. Then we can apply inference rules and obtain an abstract annotation from the string dioid over the set of ids for each triple. Finally, as soon as we need to get annotations from a specific domain we need to make sure that it is a $\top$-dioid, define a homomorphism by attaching specific annotations to triples in G and then apply it to abstract annotations of previously inferred triples.

VI. Applications to specific models

In this section we introduce several annotation domains for RDF graphs those are $\top$-dioids.

Temporal RDF [16]. This model treats the Ex. 1. The *temporal domain* $\mathcal{K}_T = \langle K_T, \oplus_T, \otimes_T, \bot_T, \top_T\rangle$ is defined as follows. Consider *temporal intervals* $[\alpha_1, \alpha_2]$ where $\alpha_{1,2} \in \mathbb{P} = \mathbb{Z} \cup \{-\infty, +\infty\}$, $\alpha_1 < \alpha_2$. Two intervals $[\alpha_1, \alpha_2]$ and $[\alpha_3, \alpha_4]$ are *adjacent* iff $\alpha_2 + 1 = \alpha_3$. The set K_T is the set of all pairwise disjoint and non-adjacent sets of intervals. On K_T a partial order is defined: $\gamma_1 \preceq \gamma_2$ iff for each $I_1 \in \gamma_1$ there exists $I_2 \in \gamma_2$ such that $I_1 \subseteq I_2$. For $\mathcal{K}_T$ we have: $\gamma_1 \oplus_T \gamma_2 = \inf\{\gamma \mid \gamma \succeq \gamma_i, i = 1, 2\}$, $\gamma_1 \otimes_T \gamma_2 = \sup\{\gamma \mid \gamma \preceq \gamma_i, i = 1, 2\}$, $\bot_T = \emptyset$ and $\top_T = \mathbb{P}$. The domain $\mathcal{K}_T$ forms an BL-algebra and hence a $\top$-dioid.

Fuzzy RDF [15] treats the Ex. 2. The domain here is $\mathcal{K}_F = \langle [0,1], \max, \otimes_F, 0, 1\rangle$, where $\otimes_F$ is any t-norm of BL-algebra. If $\otimes_F$ is an ordinary multiplication as in Ex. 2, then the domain becomes probabilistic. As $\mathcal{K}_T$ domain, $\mathcal{K}_F$ is a $\top$-dioid.

Default RDF. In both of the previous domains the product operation in the dioid is commutative. Next we introduce an example of $\otimes$-noncommutative annotations.

Suppose we want to represent – in an RDF graph – information about an attribute attached to resources modelled by the graph. The straightforward way is to introduce a new property to the vocabulary of the graph and handle it as usual. Nevertheless, in some situations this way can be not optimal: If we have a broad system of classes and values of the attribute for subclasses and elements of a class are usually the same, then it is natural to keep the default attribute value for the class and the value for a resource only if it differs from usual one of the class it belongs to.

For a substantiation, consider an RDF graph denoted in Fig. 2 representing a (part of) botanical taxonomy[3]. The triples of this graph have annotations which store values of an attribute **PublishedBy** for every taxon in the graph. The division *Pinophyta* was introduced by Carl Linnaeus, so the triple $(\mathit{Plantae}, \mathtt{sc}, \mathit{Pinophyta})$ is annotated by *Linnaeus*. The subsequent taxons down to family *Araucariaceae* keep this attribute value, so we annotate all sc-triples between *Pinophyta* and *Araucariaceae* with a special value $\star$ which means "derived from above". The same value keeps for genus *Araucaria* and species *Araucaria Araucana*. But genus *Wollemia* was introduced by David Noble as well as its only species *Wollemia Nobilis*, so the annotations for the triples $(\mathit{Araucariaceae}, \mathtt{sc}, \mathit{Wollemia})$ and $(\mathit{Wollemia}, \mathtt{sc}, \mathit{W.Nobilis})$ are Noble and $\star$ correspondingly. Thus, to find out from the graph who introduced a taxon we need to go up from its node along the tree until we find an edge with an annotation differs from $\star$. An annotation value for a triple here represents not the value of the attribute by itself, but its *difference* with the value of higher triple in the tree. Hence, it is natural to phisically keep such annotations in a memory only it they differ from $\star$. That is why if the value of an attribute for a resource in most cases is determined by a class it belongs to and the number of attributes are large, the advantage in memory can be sufficient.

Before we define the default domain formally, note, that the situation in general case can be more complicated than in the latter example, because a ρdf subgraph of a graph does not nesessarily have a tree structure. Therefore, in certain cases we need to combine different annotations for a triple. To this effect we require a set of attirbute values to have a union operation and the lowest element, i.e. to be a semilattice. In many cases (as in the taxonomy example) this set has no structure, but it is possible to enrich it with the lowest element `Unknown` and the greatest element `Error`. The latter is justified by situations

[3] This is just an example and the actual information may be incorrect.

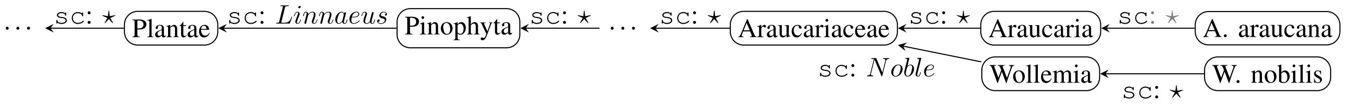


Fig. 2. The RDF graph representing botanical taxonomy

when from one source we get an attribute value for a resource, but from another source we get a different value for the same resource which contradicts with the first one. The union of this information is inconsistent and requires a manual intervention, which can be flagged by `Error` annotation.

Let $\mathbf{A}$ be an attribute with a domain $\langle A, \sqcup, 0\rangle$ which is a semilattice with union $\sqcup$ and the lowest element 0. Consider an $\mathbf{A}$-*default* domain $\mathcal{K}_{\mathbf{A}} = \langle A^{\star} \cup \{\bot^{\star}, \top^{\star}\}, \oplus_{\mathbf{A}}, \otimes_{\mathbf{A}}, \bot^{\star}, \top^{\star}\rangle$, where $A^{\star} = (A \times \{True, False\})$ and $\bot^{\star}$, $\top^{\star}$ are new special symbols. Note, that the meanings of $\bot^{\star}$ and an annotation with 0 are different: The first one says that a triple is not in the support of a graph and the second says that the value of the attribute is minimal according to $\sqcup$. The second boolean component of $A^{\star}$ corresponds to $\star$ in the taxonomy example above, i.e. it is $True$ in an annotation for a triple iff the actual value of the attribute is the union of the first component of the annotation with values those can be derived from triples above, and $False$ overwise. The operations are defined as follows:

$$(a_1, b_1) \oplus_{\mathbf{A}} (a_2, b_2) = (a_1 \sqcup a_2, b_1 \vee b_2),$$
$$(a_1, b_1) \otimes_{\mathbf{A}} (a_2, b_2) = \begin{cases} (a_1 \sqcup a_2, b_2) \text{ iff } b_1 = True, \\ (a_1, b_1) \text{ iff } b_1 = False. \end{cases}$$

These operations extend to elements $\bot^{\star}$ and $\top^{\star}$ in a way to keep the properties of $\bot$ and $\top$ in the dioid. Next, we describe the meaning of this operations. The addition $\oplus_{\mathbf{A}}$ is applied when we union annotations about the same triple. Hence, the values a_1 and a_2 are joined by the semilattice operation $\sqcup$ and the possibility to derive values from a ρdf structure above exists only if it exists in any of the considering annotations. The product $\otimes_{\mathbf{A}}$ is applied when we infer triples by transitivity of `sc` or similar inference rule from Tab. IV. If $b_1 = True$ we union the current value a_1 with the derived value a_2 and the possibility of father deriving depends on b_2. If $b_1 = False$, we just keep the annotation (a_1, b_1) which override any annotation from a structure above. Finally, $\mathcal{K}_{\mathbf{A}}$ can be easily checked to be $\otimes$-noncommutative $\top$-dioid.

To use the $\mathbf{A}$-default dioid $\mathcal{K}_{\mathbf{A}}$ for storing values of an attribute $\mathbf{A}$ for resources we assume $(0, True)$-annotation for a triple by default and do not keep it in a memory. As soon as we need the real value of the attribute for a resource a we infer a triple $(a, \texttt{type}, r)$ and get the first component of annotation for it. (Here r is the *root* of ρdf subgraph of the considered graph; if it does not exist, we can always introduce it.)

VII. Conclusions

Although annotation algebras have been studied for database query languages [8] and have also recently been investigated for RDF query languages [7], we have suggested an alternative and more natural algebra for the annotation of RDFS, and we have given some examples of its use. There may be some mileage to be gained by combining these two algebras. One of the applications of the proposed algebra is a system for default reasoning about certain annotations on RDF resources, which may also prove to be a useful mechanism for physically storing those annotations. We would like to find similar mechanisms for efficiently storing default annotations on triples.

We are grateful to Marcelo Arenas, Boris Motik, Floris Geerts and Alex Simpson for helpful discussions. This work was supported by a grant from the UK Engineering and Physical Sciences Research Council.

References

[1] D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation. http://www.w3.org/TR/rdf-schema/, Feb. 2004.

[2] P. Buneman, S. Khanna, and W. Tan. Why and Where: A Characterization of Data Provenance. In *ICDT 2001*, volume 1973 of *LNCS*, pages 316–330. Springer, 2001.

[3] J. J. Carroll, C. Bizer, P. J. Hayes, and P. Stickler. Named graphs, provenance and trust. In *WWW*, pages 613–622, 2005.

[4] L. Chiticariu, W.-C. Tan, and G. Vijayvargiya. DBNotes: a post-it system for relational databases based on provenance. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 942–944, New York, NY, USA, 2005. ACM.

[5] Y. Cui, J. Widom, and J. L. Wiener. Tracing the lineage of view data in a warehousing environment. *ACM Trans. Database Syst.*, 25(2):179–227, 2000.

[6] R. D. Dowell, R. M. Jokerst, A. Day, S. R. Eddy, and L. Stein. The Distributed Annotation System. *BMC Bioinformatics*, 2:7, 2001.

[7] G. Flouris, I. Fundulaki, P. Pediaditis, Y. Theoharis, and V. Christophides. Coloring RDF Triples to Capture Provenance. In *International Semantic Web Conference*, pages 196–212, 2009.

[8] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *PODS '07: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 31–40, New York, NY, USA, 2007. ACM.

[9] C. Gutierrez, C. A. Hurtado, and A. Vaisman. Introducing Time into RDF. *IEEE Trans. on Knowl. and Data Eng.*, 19(2):207–218, 2007.

[10] C. Gutiérrez, C. A. Hurtado, and A. A. Vaisman. Temporal RDF. In *ESWC*, pages 93–107, 2005.

[11] P. Hájek. *Metamathematics of Fuzzy Logic (Trends in Logic)*. Springer, 1 edition, November 2001.

[12] A. Hogan, G. Lukacsy, N. Lopes, A. Polleres, U. Straccia, A. Zimmermann, and S. Decker. RDF Needs Annotations. In *Proceedings of W3C Workshop — RDF Next Steps*. W3C, 2010.

[13] F. Manola, E. Miller, and B. McBride. RDF Primer, W3C Recommendation. http://www.w3.org/TR/REC-rdf-syntax/, Feb. 2004.

[14] S. Muñoz, J. Pérez, and C. Gutiérrez. Minimal Deductive Systems for RDF. In *ESWC*, pages 53–67, 2007.

[15] U. Straccia. A Minimal Deductive System for General Fuzzy RDF. In *Proceedings of the 3rd International Conference on Web Reasoning and Rule Systems (RR-09)*, number 5837 in Lecture Notes in Computer Science, pages 166–181. Springer-Verlag, 2009.

[16] U. Straccia, N. Lopes, G. Lukacsy, and A. Polleres. A General Framework for Representing and Reasoning with Annotated Semantic Web Data. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*. AAAI Press, 2010.

[17] Y. R. Wang and S. E. Madnick. A Polygen Model for Heterogeneous Database Systems: The Source Tagging Perspective. In *VLDB*, pages 519–538, 1990.

Calculating the Trust of Event Descriptions using Provenance

Davide Ceolin, Paul Groth, Willem Robert van Hage
VU University Amsterdam
Amsterdam, The Netherlands
Email: dceolin,pgroth,wrvhage@few.vu.nl

***Abstract*—Understanding real world events often calls for the integration of data from multiple often conflicting sources. Trusting the description of an event requires not only determining trust in the data sources but also in the integration process itself. In this work, we propose a trust algorithm for event data based on Subjective Logic that takes into account not only opinions about data sources but also how those sources were integrated. This algorithm is based on a mapping between a general event ontology, the Simple Event Model, and a model for describing provenance, the Open Provenance Model. We discuss the results of applying the algorithm to a use case from the maritime domain.**

I. Introduction

The hijacking of a freighter in the Gulf of Aden, a goal not given in the semi-final of the World Cup and the sudden rise of the stock market, understanding these *events* requires the integration of data from multiple data sources using complex data integration routines. For example, to build a description of why a goal was not given there may be the report of the referee, the comments of managers and players, and video from different camera angles. The veracity of the resulting *description of the event* is dependent not only upon the trust one has in the original data sources (e.g. players, referees, cameras) but also in trust one has in the process used to create the event description.

Therefore, in this work, we investigate the generation of trust ratings for event descriptions. These trust ratings are calculated with respect to not only the original sources but also to the data integration process itself. Thus, the trust calculations consider the whole of an event description's *provenance*. The trust algorithms presented here rely on the novel combination of two existing representations, the Simple Event Model (SEM) for event representations and the Open Provenance Model (OPM) for representing the data integration process itself. Based on a mapping of these models, we develop a trust algorithm using subjective logic. We apply our trust algorithm to a use case from maritime shipping. The contributions of this paper are twofold:

1) A mapping of SEM to OPM.
2) An algorithm for computing trust ratings for event descriptions based on their provenance.

The rest of this paper is organized as follows. We begin with a description of a use case for data integration for event descriptions, which we use as a running example. This is followed by a discussion of both OPM and SEM and a presentation of the mapping between these models. Based on this mapping, we then present an algorithm for producing trust ratings for event descriptions. After this we present initial results applied to the use case. We end with a discussion of related work and a conclusion.

II. Use Case

Our use case comes from the maritime domain. It is of vital importance for the coast guard, harbors and ships to know where ships are and their vicinity to one another. Being able to track ships helps avoid collisions, manage traffic in crowded harbors, respond to emergency, and facilitate navigation. To enable this tracking, a common system has been developed called The Automatic Identification System (AIS) has been developed.[1] The International Maritime Organization requires that the system be installed on all ships over 300 tons. AIS works by exchanging messages between local ships and radar stations. This messages provide a range of information about the ship including its geoposition, navigation status, speed, radio call sign, the ship's unique registered id (MMSI - Maritime Mobile Service Identity), a permanent id (IMO - International Maritime Organization Number) and the ship's dimensions. Such messages are subject to manipulation, corruption, and errors impacting their reliability [1]. For example, the unique registered id may be falsely programmed into the system, the message may be corrupted during radio transmission, or users may fail to update their navigation status.

An AIS message or series of AIS messages describe the event of a ship's movement or change in status. Often, one would like to extract information about that event. Here, we use a simple example of extracting what nation the ship is registered to. This is known as the *flag* of the ship. This is actually a difficult problem as both the MMSI number as well as the IMO number report the country of origin and these may disagree because the MMSI can change when the ship is reregistered. Indeed, one report identified 26 vessels using the same MMSI number [1]. In addition, country information may be garbled or incorrectly entered. Thus, if part of the event description is a flag then it is important to be able to determine whether to trust that flag information based on the information sources and how those sources were combined. SEM is already

[1] http://www.uais.org

being used to represent ship movement events based on AIS messages [2]. However, we need to add additional information to represent the provenance of the description. For this, we turn to a model designed specifically for provenance, namely, OPM.

III. Mapping SEM and OPM

In order to connect the description of an event to how that description was created, we need to be able to interpret the event description with respect to its provenance. To do so, we provide a mapping from the model used for event descriptions (SEM) to the model used for describing provenance (OPM). To facilitate the explaination of this mapping, we first briefly introduce both SEM and OPM.

A. SEM, the Simple Event Model

SEM [2], [3], [4] is a schema for the semantic representation of events. It does not deal with the way data about events is stored, but only with the events themselves. SEM focuses on modeling the most common facets of events: who, what, where, and when. These are represented respectively by the SEM core classes sem:Actor, sem:Place, sem:Object and sem:Time. SEM is a model that takes into account the inherent messiness of the Web by making as little semantic commitment (e.g. disjointness statements, functional properties) as possible. Every instance of one of the core classes can be assigned types from domain vocabularies. For example, the sem:Event instance ex:world_cup_2010 can be assigned a sem:eventType dbpedia:FIFA_Club_World_Cup. Any property of SEM, including the type properties, is optional and duplicable. SEM and Simple Knowledge Organization System (SKOS) [5] mappings to related models (DOLCE-Lite, CIDOC-CRM, SUMO, LODE, F, Dublin Core, FOAF, and the CultureSampo and Queen Mary's event models) can be accessed online.[2]. Additionally, through sem:View an event can have multiple, perhaps conflicting, descriptions.

B. OPM, the Open Provenance Model

OPM is a community developed model for the exchange of provenance information [6]. It stems from a series of interoperability challenges (Provenance Challenges) held by the provenance research community to understand and exchange provenance information between systems. While not as comprehensive as some other provenance models such as ProPreO [7] , OPM provides a common technology-agnostic layer of agreement between systems. OPM was used by 15 teams during the Third Provenance Challenge [6]. These teams used a variety of provenance management systems ranging from those focused on workflow systems to those concentrating on operating systems. Thus, by using OPM, we aim to be able to apply our trust algorithm to a variety of systems.

OPM represents the provenance of an object as a directed acyclic graph with the possibility for annotations on the graph. The graph is interpreted as being causal. An OPM graph captures the past execution of a process. The graph consists of three types of nodes:

SEM	SKOS relation	OPM
sem:Event	skos:closeMatch	opm:Process
sem:Actor	skos:closeMatch	opm:Artifact
sem:Actor	skos:broadMatch	opm:Agent
sem:Place	skos:closeMatch	opm:Artifact
sem:Place	skos:broadMatch	opm:Agent
sem:Role	skos:closeMatch	opm:Role
sem:View	skos:closeMatch	opm:Account

TABLE I
Mapping between OPM and SEM

- An *opm:Artifact*, which is an immutable piece of state, for example, a file.
- An *opm:Process*, which is perform actions upon artifacts and produce new artifacts. An example of a process would be the execution of the Unix command cat on two files to produce a new concatenated file.
- An *opm:Agent*, which controls or enables a process. An example of an agent would be the operating system that a process runs in or the person who started the process.

These nodes are linked by five kinds of edges representing dependency between nodes. An opm:Process used and generated opm:Artifacts, represented by opm:used and opm:wasGeneratedBy edges. These artifacts can be given an opm:Role with respect to an opm:Process distinguishing it from other artifacts. Note, an opm:Process can only produce one opm:Artifact. Dependency between opm:Artifacts is represented using opm:wasDerivedFrom while dependency between opm:Processes is represented using the opm:wasTriggeredBy edge. Finally, the control of an opm:Process by an opm:Agent is expressed using the opm:wasTriggeredBy edge.

Each part of an OPM graph can be labeled with an *account*, which allows the same execution to be explained from different perspectives. For example, one could describe the generation of an event description with more or less detail.

C. Mapping

Given an event description in SEM, we would like to determine how its facets should map to OPM so that we can describe the facet's provenance using OPM. For example, if an event occurred at a sem:Place, we could consider that place an opm:Artifact. This idea is in-line with the notion of sub-typing within OPM [6]. We could say that a particular opm:Artifact has a type of sem:Place. To represent the mapping, we use SKOS, a W3C standard for describing and mapping vocabularies (i.e. concept schemes). The use of SKOS follows the practice of the W3C Provenance Incubator Group in defining a set of Provenance Vocabulary Mappings [8]. We refer the readers to [5] for the exact definitions of skos:closeMatch, skos:relatedMatch and skos:broadMatch.

Our mapping focuses on the nodes within the OPM graph and not the edges, because our aim is to describe the provenance of both the event description and its facets. We now discuss the mapping shown in Table I in more detail.

[2] http://semanticweb.cs.vu.nl/2009/11/sem/

For sake of space, we report only a mapping at class level. A more comprehensive mapping detailed with justifications is available on the web.[3]

Each sem:Event is an action with some duration, this maps very closely with the notion of an opm:Process. SEM has the notion of an sem:Actor, the entities or people *who* take part or are involved in an event. If an sem:Actor is directly a cause or is vital for an event to take place, we would model this as an opm:Artifact used by an opm:Process. For people who were not directly involved but enabled the event to take place, the sem:Actor would be mapped to an opm:Agent. By way of example, the crew on board a ship would be modeled as opm:Artifacts while the CEO of the shipping company can be seen as an opm:Agent controlling the event of sending an AIS message. Similar reasoning applies to mapping sem:Place to OPM.

The sem:Role signifies the role a particular SEM facet plays in an event, just as an opm:Role signifies the role a particular opm:Artifact plays with respect to an opm:Process. Additionally, an sem:View allows for multiple descriptions of the same event, which maps naturally to an opm:Account describing different descriptions of the same execution. Finally, the time of an sem:Event can be easily mapped to the time annotations present on OPM edges.

IV. Trust Rating Algorithm

We now describe our trust rating algorithm. The algorithm works upon OPM graphs. We assume that the provenance of each facet of an event description is captured. Before applying the algorithm, the above mapping is applied in order to view the facets of the SEM event description in OPM.

A. Subjective Logic

Subjective logic [9] is a probabilistic logic that provides the basis for the evidential reasoning part of our trust model. Subjective logic's probabilities are based on the Beta probability distribution [10]. These probabilities represent the level of belief, disbelief and uncertainty about each proposition we encounter, according to the evidence we own and are represented by means of "opinions" about such propositions.

This logic provides also operators for combining such opinions in order to handle the combination of opinions that reflect the application of propositional logic operators to the proposition which are objects of such opinions.

B. Opinions

The key concept of Subjective Logic logic is the concept of "opinion", which is the probability of correctness of a proposition according to a certain source. An opinion according to source x about proposition y is represented as ω_y^x. More precisely, opinions are depicted as follows:

$$\omega_x^y(b, d, u, a)$$

which is a representation equivalent to the Beta probability distribution, where :

$$b = \frac{positive\_evidence}{total\_evidence + n} \; d = \frac{negative\_evidence}{total\_evidence + n}$$

$$u = \frac{n}{total\_evidence + n} \; a = \frac{1}{n}$$

b,d,u are, respectively, *belief*, *disbelief* and *uncertainty*. *a* is the *a priori probability*, that is the probability that the proposition is correct, in absence of evidence. *n* is the cardinality of the set of possible outcomes, so it may be equal to 2, in case of a boolean outcome, or higher.

The expected value of the probability distribution represented by an opinion is given by:

$$E = b + a \times u$$

The expected value E will be used as trust value about propositions. *E is the "trust value"*. Given the evidence that we have collected about a certain proposition, E represents the probability that the proposition is true. Therefore it numerically quantifies our trust in the proposition.

Consider the following example. There are 249 countries in the world. Thus, the number of possible outcomes for a flag is 249. For sake of simplicity, we consider the 35 most used flags, which cover 99% of ships.

Here we consider three sources of information about the flag. Two sources say the flag is Italy. One source says the flag is the USA. Each of these opinions is secure according to each source, therefore they assume the pattern $\omega_y^x\left(1, 0, 0, \frac{1}{n}\right)$.

$$\omega_{italy}^{s_1}\left(1, 0, 0, \frac{1}{35}\right) \; \omega_{italy}^{s_2}\left(1, 0, 0, \frac{1}{35}\right) \; \omega_{usa}^{s_3}\left(1, 0, 0, \frac{1}{35}\right)$$

These are the opinions about the three sources, where $n = 2$ because, unlike previous opinions that represent the probability that a given value is correct (in a multivalued distribution), these opinions represent the probability that the source is reliable (therefore in this case the probability distribution is binomial):

$$\omega_{s_1}^x\left(\frac{8}{12}, \frac{2}{12}, \frac{2}{12}, \frac{1}{2}\right) \; \omega_{s_2}^x\left(\frac{9}{12}, \frac{1}{12}, \frac{2}{12}, \frac{1}{2}\right)$$

$$\omega_{s_3}^x\left(\frac{5}{12}, \frac{5}{12}, \frac{2}{12}, \frac{1}{2}\right)$$

Procedure *opinion_source(A_i)* of Algorithm Fig. 1 (Lines 26 - 30) builds opinions for given Artifact A_i.

C. Weighting (discounting) operators

Subjective Logic allows to build networks of opinions. The logic allows opinions to be transitive, but such opinions are weighted on the reputation of the source when evaluated by third parties. Given the opinion of z on y (ω_y^z), and the opinion of x on z (ω_z^x), the opinion that x derives from z about y is represented by $\omega_y^{x:z}$. The operator for weighting opinions is:

$$\omega_z^x \otimes \omega_y^z = \omega_y^{x:z}(b_z^x b_y^z, b_z^x d_y^z, d_z^x + u_z^x + b_z^x u_y^z, a_y^z)$$

[3] http://bit.ly/c8A3A7

Following the previous example, the weighted opinions become:

$$\omega^{x:s_1}_{italy}\left(\frac{8}{12}, 0, \frac{4}{12}, \frac{1}{35}\right)\ \omega^{x:s_2}_{italy}\left(\frac{9}{12}, 0, \frac{3}{12}, \frac{1}{35}\right)$$

$$\omega^{x:s_3}_{usa}\left(\frac{5}{12}, 0, \frac{7}{12}, \frac{1}{35}\right)$$

All the disbeliefs have value zero as consequence of starting from secure opinions.

On line 31 of Algorithm of Figure 1, procedure opinion_sources(A_i) returns opinions about artifact A_i weighted on reputation of the sources.

D. Fusion operator

Finally, the logic provides a range of operators which allow us to combine opinions about the same proposition (fusion). The fusion of n opinions given by sources $x_1, ..., x_n$ about the same proposition y is represented as $\omega_y^{x_1 \diamond \ldots \diamond x_n}$. The operator works as follows:

$$\omega^{s_i}_y \oplus \omega^{s_j}_y = \omega^{s_i \diamond s_j}\Big(\frac{b^{s_i}_y \times u_B + b^{s_j}_y \times u^{s_i}_y}{u^{s_i}_y + u^{s_j}_y - u^{s_i}_y \times u^{s_j}_y},$$

$$\frac{d^{s_i}_y \times u^{s_j}_y + d^{s_j}_y \times u^{s_i}_y}{u^{s_i}_y + u^{s_j}_y - u^{s_i}_y \times u^{s_j}_y}, \frac{u^{s_i}_y \times u^{s_j}_y}{u^{s_i}_y + u^{s_j}_y - u^{s_i}_y \times u^{s_j}_y}, a^{s_i}_y\Big)$$

Since s_i's and s_j's opinion have the same object, their a priori probability is the same ($a^{s_i}_y = a^{s_j}_y$).

$\oplus$ is an operator that returns cumulative fusion of opinions [11] (since we assume that they are independent opinions, evidence that these opinions resemble are cumulated).

Continuing our example, by merging the previous opinions regarding the two outcomes (Italy and USA), we obtain:

$$\omega^{x:s_1 \diamond x:s_2}_{italy}(0.77, 0.14, 0.09, 0.5)\ \omega^{x:s_3}_{usa}(0.42, 0.42, 0.16, 0.5)$$

Line 21 of algorithm of Figure 1 iteratively merges opinions about the Artifact of interest.

E. Trust Rating Algorithm

Here we present an algorithm for calculating the trust value of an event facet, represented by artifacts. However, because of its recursive nature, the algorithm is directly applicable to event descriptions.

Given an artifact to calculate the trust value of, our first step is determine the opinion of any source that directly generates the artifact's value. The following steps are:

- take the amount of evidence given by each source about each possible value for the artifact. Usually each source gives one output, but if more are available, then the resulting opinion is stronger (see subsect. IV.B).
- weight the opinions given by the sources according to the opinion on the source itself (in turn, based on previous evidence about its trustworthiness, see subsection IV.C)
- merge all the opinions (see subsection IV.D)

```
(1)  proc tv ($A_i$) $\equiv$
(2)    $res := null$
(3)    for $P_k : A_i\ opm : wasGeneratedBy\ P_k$ do
(4)        for $A_j : P_k\ opm : used\ A_j$ do
(5)            if $A_i\ opm : wasDerivedFrom\ A_j$
(6)              then
(7)                   if $res = null$
(8)                     then $res := tv(A_j)$
(9)                     else $res := F(P_k)(res, tv(A_j))$
(10)                  fi
(11)           fi
(12)       od
(13)   od
(15)   comment: res = $\omega_{v(A_i)}^{\forall_{A_j} x:tv(A_j)}$
(16)   for $s_i : \exists v_{s_i}(A_i) \neq \emptyset$ do
(17)       if $res = null$
(18)         then $res := opinion\_sources(A_i)$
(19)         else $res := res \oplus opinion\_sources(A_i)$
(20)       fi
(21)   od
(22)   return $res$
(23) end
(24) proc opinion_source($A_i$)
(25)   for $s_i : v_{s_i}(A_i) \neq null$ do
(26)       record_evidence($v_{s_i}(A_i)$)
(27)   od
(28)   return $\omega^{x:s_i}_{v(A_i)}$
(29) end
(30) proc $\pi(t, s_i, A_i)$
(31)   $e : e \in domain \wedge$ dist$(e, v_{s_i}(A_i) =$
(32)   $=$ min$_{\forall e' \in domain}$(dist$(e', v_{s_i}(A_i))$)
(33)   $d :=$ dist$(e, v_{s_i}(A_i))$
(34)   record $\omega^{s_i}_{v_{s_i}(A_i)=e}(b'_{s_i} \cdot \frac{1}{d}, 0, (d'_{s_i} + u'_{s_i}) \cdot (1 - \frac{1}{d}), a'_{s_i})$
(35)   comment: $b'_{s_i}, d'_{s_i}, u'_{s_i}, a'_{s_i}$ are the
(36)   comment:  projections of $b_{s_i}, d_{s_i}, u_{s_i}, a_{s_i}$
(37) end
(38) proc dist
(39)   comment: distance between two points
(40)   comment:  (e.g. Euclidean).
(41) proc record_evidence
(42)   comment: stores evidence in memory .
(43) proc record
(44)   comment: stores opinion in memory.
(45) proc $\omega$
(46)   comment: returns an opinion
(47)   comment:  based on stored evidence.
(48) comment: Possible values for F:
(49) $F(concat) = \wedge$
(50) $F(lookup(t)) = \wedge \cdot \pi(t)$
```

Fig. 1. Trust Rating Algorithm

Generalizing, we can say that:

- given an artifact A;
- given a set of sources: s_1, ... s_n
- given a function $v(s_i, A) = v_{s_i}(A)$
- given opinions on the sources $\omega^x_{s_i}(b_{s_i}, d_{s_i}, u_{s_i}, a_{s_i})$

We compute the opinion on a event facet from each source:

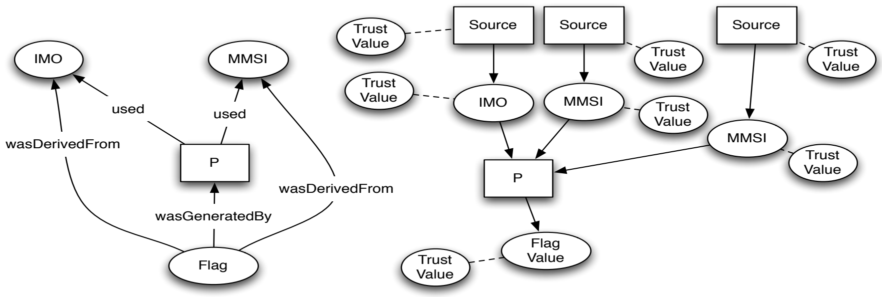


Fig. 2. Provenance and Trust graphs about the flag value of a ship. The left graph reconstructs the provenance of the flag field. The graph on the right, starting from the first ancestors of the flag field, collects all the evidence about all the artifacts involved in the provenance trail (of the left graph) and gradually merges them.

$$\omega^{x:s_i}_{v_{s_i}(A)}(b_{s_i}, 0, d_{s_i} + u_{s_i}, a_{s_i})$$

Once we have the opinions about the values from each source, we merge them in order to obtain an opinion for each value from all sources:

$$\bigoplus_{v_{s_i}} \omega^{x:s_i}_{v_{s_i}(A)}(b_{s_i}, 0, d_{s_i} + u_{s_i}, a_{s_i})$$

F. Integration process

We want to consider not only sources that directly provide the artifact value but also which process is used during integration to generate the artifact. Therefore, in case the artifact is not a leaf node, then we need to merge the (eventual) opinions computed taking into account the provenance of the artifact. For example, considering the example of Fig. 2, we see that the trust level of the root node depends on the trust levels of the leaf nodes, combined according to how the process manipulates them. Therefore, we should use a functor that, allows us to apply proper functions to the trust values of the input artifacts, according to the kind of process that manipulates them.

Two examples are provided in Algorithm 1: in case of a concatenation process (that takes as inputs two strings and outputs their concatenation), then all the trust value equally contribute to determining the outcome and therefore they are merged by conjunction. In case of a lookup process (that takes as inputs a key and a value table, and outputs the value in the table corresponding to the key), then before calculating the conjunction of the trust values, we project them into the space of the possible values, possibly smaller than the space of plausible ones. Moreover, in case the value we face does not fall into the range of possible values, then we consider the value or values closer to it and belonging to the sset of possible values. Clearly, we weight these contributions according to the distance to the given value.

V. Applying the algorithm

We now discuss how, by taking advantage of both provenance and background knowledge, the trust algorithm can produce more precise trust ratings.

One important feature of the algorithm is that, by means of provenance, we encorporate in our algorithm also semantic information.

This way, we restrict the domain of possible value for each field to the range of real, meaningful values. For instance, if the nationality field of a MMSI is a 3 digit code, then there are 10^3 possible values, since any cypher would be equally probable in each of the 3 positions. By taking into account the meaning (semantics) of the MMSI, the cardinality of the set of the plausible values would restrict to 35 (considering the countries which own 99% of the ships). This means that if we own 10 positive evidence and we restrict the plausibilty set from 1000 to 35, then the trust value rises from $E = \frac{10}{1010} + \frac{1}{1000} \times \frac{1000}{1010} = 0,0189...$ to $E = \frac{10}{45} + \frac{1}{35} \times \frac{35}{45} = 0,3143.....$ Note that the MMSI field is retrieved via traversing the provenance graph.

Another important feature of the algorithm is the usage of provenance information. Because of this, we enlarge the availability of evidence at disposal for calculating trust values. In fact, we don't limit to the use of direct evidence about the facets we have to evaluate, but we consider also evidence about elements used in the process that lead us to our facets. Therefore, we check whether these initial elements were correct and whether they were combined properly in order to produce the facet we are analyzing. Once we have this result, we can compare it with evidence directly referred to the facet we are evaluating, obtaining an improvement of the precision of the trust value.

Continuing the previous example, if we have also sources that provide a value for the nation, knowing that the national code is determined by looking it up into a trusted table, then by applying the Trust Ranking Algorithm, we obtain the following trust value: $E = \frac{20}{45} + \frac{1}{35} \times \frac{35}{45} = 0,4667.....$

If we adopt a conservative approach and accept only facets which trust value is above a certain threshold, then this change reduces the amount of errors due to false negatives.

VI. Related Work

Trust is a widely explored topic within a variety of areas within computer science including security, intelligent agents, software engineering and distributed systems. Here, we focus on those works directly touching upon the junction of trust, provenance and the Semantic Web. For a readable overview of trust research in artificial intelligence, we refer readers to Sabater and Sierra [12]. For a more specialized review of trust research as it pertains to the Semantic Web see Artz and Gil [13]. Finally, Golbeck provides a longer review of trust research as it relates to the Web [14].

Our work is closest to the work on using provenance for information quality assessment on the Semantic Web. In the WIQA framework [15], policies can be expressed to determine whether to trust a given information item based on both provenance and background information expressed as Named Graphs [16]. Hartig and Zhao follow a similar approach using annotated provenance graphs for a given information item to perform the quality assessment and thus generate a trust value [17]. However, their work uses a more complex provenance representation similar to OPM that captures not only the data origins but also the processing steps involved. Similarly, IWTrust generates trust values for answers produced by a question answering system based on a combination of source data, provenance information, and user ratings [18]. Our work differs from these approaches in three respects: First, we concentrate on event descriptions and not generic data items. Second, our work takes advantage of a priori knowledge about the likelihood of data items in order to correct for possible data errors. Finally, we use Subjective Logic to allow for multiple (possibly conflicting) opinions about data sources to be taken into account, but unlike [19], we use it in combination with provenance.

Recent work has focused on querying trust using SPARQL [20]. We see our work as complementary in that it could facilitate the population of the trust values to query over. Finally, other work has considered using provenance and ontologies to determine the trust in electronic contracts [21]. Our work differs, in that they use provenance as a source of experience for calculating opinion values whereas we focus on the combination of current opinion values to produce a final trust value.

VII. Conclusion

Here, we presented a trust algorithm for determining trust of event descriptions based on provenance. We provide a novel mapping between an event ontology and a widely used provenance ontology. Secondly, we show how Subjective Logic can be used in combination with provenance to generate improved trust values in a maritime data integration domain. In the future, we will perform a comprehensive evaluation of the model and extend Subjective Logic to handle a contextualization of opinions and address some of its limitations (see [22]). Additionally, we aim to expand our work applying it to the problem of determining trust of event descriptions produced from data integrated from the Web.

References

[1] A. Harati-mokhtari, A. Wall, P. Brooks, and J. Wang, "Automatic identification system (ais): A human factors approach," 2008. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.127.1049

[2] W. R. van Hage, V. Malaisé, G. de Vries, G. Schreiber, and M. van Someren, "Combining ship trajectories and semantics with the simple event model (sem)," in *EiMM '09: Proceedings of the 1st ACM international workshop on Events in multimedia*. New York, NY, USA: ACM, 2009, pp. 73–80.

[3] N. Willems, W. R. van Hage, G. de Vries, J. Janssens, and V. Malaisé, "An integrated approach for visual analysis of a multi-source moving objects knowledge base," *IJGIS (to appear)*, 2010.

[4] W. R. van Hage, V. Malaisé, G. de Vries, and A. T. Schreiber, "Abstracting and reasoning over ship trajectories and web data with the simple event model (sem)," *MTAP (to appear)*, 2010.

[5] S. Bechhofer and A. Miles, "SKOS Simple Knowledge Organization System Reference," W3C, W3C Recommendation, Aug. 2009.

[6] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, and J. Myers, "The Open Provenance Model core specification (v1.1)," *Future Generation Computer Systems*, Jul. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.future.2010.07.005

[7] S. S. Sahoo, A. Sheth, and C. Henson, "Semantic provenance for eScience: Managing the deluge of scientific data," *IEEE Internet Computing*, vol. 12, pp. 46–54, 2008. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/MIC.2008.86

[8] S. Sahoo, P. Groth, O. Hartig, S. Miles, S. Coppens, J. Myers, Y. Gil, L. Moreau, J. Zhao, M. Panzer, and D. Garijo, "Provenance Vocabulary Mappings," 20010. [Online]. Available: http://www.w3.org/2005/Incubator/prov/wiki/Provenance_Vocabulary_Mappings

[9] A. Jøsang, "Probabilistic logic under uncertainty," in *Theory of Computing 2007. Proceedings of the Thirteenth Computing: The Australasian Theory Symposium (CATS2007). January 30 - Febuary 2, 2007, Ballarat, Victoria, Australia, Proceedings*, ser. CRPIT, J. Gudmundsson and C. B. Jay, Eds., vol. 65. Australian Computer Society, 2007, pp. 101–110.

[10] Wikipedia. (2010, Jun.) Beta distribution. [Online]. Available: http://en.wikipedia.org/wiki/Beta_distribution

[11] A. Jøsang and D. McAnally, "Multiplication and comultiplication of beliefs," *Int. J. Approx. Reasoning*, vol. 38, no. 1, pp. 19–51, 2005.

[12] J. Sabater and C. Sierra, "Review on Computational Trust and Reputation Models," *Artificial Intelligence Review*, vol. 24, no. 1, p. 33, 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1057866

[13] D. Artz and Y. Gil, "A survey of trust in computer science and the Semantic Web," *J. Web Sem.*, vol. 5, no. 2, pp. 58–71, 2007. [Online]. Available: http://www.isi.edu/∼gil/papers/artz-gil-jws07.pdf

[14] J. Golbeck, "Trust on the world wide web:a survey," *Foundations and Trends in Web Science*, vol. 1, no. 2, pp. 131–197, 2006. [Online]. Available: http://portal.acm.org/citation.cfm?id=1373449

[15] C. Bizer and R. Cyganiak, "Quality-driven information filtering using the WIQA policy framework," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 1, pp. 1–10, Jan. 2009.

[16] J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler, "Named graphs, provenance and trust," *International World Wide Web Conference*, 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1060745.1060835

[17] H. Olaf and J. Zhao, "Using Web Data Provenance for Quality Assessment," in *Proceedings of the 1st Int. Workshop on the Role of Semantic Web in Provenance Management (SWPM) at ISWC*, Washington, USA, 2009.

[18] I. Zaihrayeu, P. Pinheiro˜da Silva, and D. L. McGuinness, "IWTrust: Improving User Trust in Answers from the Web," in *Proceedings of 3rd International Conference on Trust Management (iTrust2005)*. Paris, France: Springer, 2005, pp. 384–392.

[19] D. Ceolin, W. R. van Hage, and W. Fokkink, "A trust model to estimate the quality of annotations using the web," in *WebSci10: Extending the Frontiers of Society On-Line*, 2010. [Online]. Available: http://journal.webscience.org/315/

[20] O. Hartig, "Querying Trust in RDF Data with tSPARQL," in *Proceedings of the 6th European Semantic Web Conference (ESWC)*, Heraklion, Greece.

[21] P. Groth, S. Miles, S. Modgil, N. Oren, M. Luck, and Y. Gil, "Determining the Trustworthiness of New Electronic Contracts," in *Proceedings of the Tenth Annual Workshop on Engineering Societies in the Agents' World, (ESAW-09)*, Utrecht, The Netherlands, 2009.

[22] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust." ACM Press, 2004, pp. 403–412.

Prov4J: A Semantic Web Framework for Generic Provenance Management

André Freitas, Arnaud Legendre, Seán O'Riain, Edward Curry
Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway
Galway, Ireland

Abstract—**Provenance is a cornerstone element in the process of enabling quality assessment for the Web of Data. Applications consuming or generating Linked Data will need to become provenance-aware, i.e., being able to capture and consume provenance information associated with the data. This will bring provenance as a key requirement for a wide spectrum of applications. This work describes Prov4J, a framework which uses Semantic Web tools and standards to address the core challenges in the construction of a generic provenance management system. The work discusses key software engineering aspects for provenance capture and consumption and analyzes the suitability of the framework under the deployment of a real-world scenario.**

Keywords- provenance management; semantic web.

I. INTRODUCTION

The Web is evolving into a complex information space where users have access to an unprecedent volume of information. The advent of Linked Data in the last years as the de-facto standard to publish data on the Web, and its uptake by early adopters[1][2], defines a clear trend towards a Web where users will be able to easily aggregate, consume and republish data. With Linked Data, Web information can be repurposed with a new level of granularity and scale. In this scenario, tracking the provenance of an information artifact will play a fundamental role on the Web, enabling users to determine the suitability and quality of a piece of information.

As a direct consequence, Linked Data applications will demand mechanisms to track and manage provenance information. This new common requirement is inherent to the level of data integration provided by Linked Data and it is not found in most systems consuming information from 'data silos', where the relationship among data sources and applications is, in general, more rigid.

Until now, provenance management has been a wide concern in the domain of scientific workflow systems [1, 2], enabling understandability and reproducibility in scientific experiments. Provenance on the Web introduces new and broader requirements for representing and managing provenance[3], as different communities are represented under the same space.

This work discusses provenance management from the perspective of this larger audience, describing Prov4J[4], a general-purpose open source provenance management system. The framework uses Semantic Web standards and tools to deploy a generic and standards-based solution. The paper also discusses key software engineering aspects in the process of designing the framework.

The central goal behind the design of the framework is to provide a set of core functionalities that enable users to develop provenance-aware applications, both from the consumption (discovery/query/access) and from the capture (logging/ publishing) perspectives.

The paper is structured as follows: section II introduces a motivational scenario; sections III and IV describe general aspects of provenance management and the architecture behind Prov4J; sections V and VI cover the consumption and capture cycles of provenance management, discussing the application of Semantic Web standards and tools in the construction of the framework. Section VII provides a brief analysis of the framework using a real world scenario based on the motivational scenario; section VIII present related work and section IX conclusions and future work.

This work concentrates its contributions: (1) in the description and analysis of a generic provenance framework for the Web using Semantic Web standards and tools; (2) in the analysis of the suitability of these standards and tools in the process of building this framework.

II. MOTIVATIONAL SCENARIO

Financial analysts in an investment company are using information from the Web to help make investment decisions. Business related data aggregated from different Web sources is filtered, curated and analyzed, and financial reports about companies or investment areas are generated. Each report is a data mash-up and the provenance of each statement in the report should be tracked to its sources. The ecosystem of Web applications used for aggregating, filtering, curating, analyzing and visualizing the data should be provenance-aware, i.e. the

[1] http://data.gov.uk, UK Government Data

[2] http://data.nytimes.com/, NYT Open Data

[3] http://www.w3.org/2005/Incubator/prov/wiki/User_Requirements, W3C Provenance Incubator Group

[4] http://prov4j.org

historical trail of all the entities and processes behind the transformation of the original data need to be recorded and users should be able to access the provenance of data.

III. A GENERIC PROVENANCE FRAMEWORK FOR THE WEB

The core goal behind Prov4J is the provision of a provenance management mechanism for the large set of applications which will increasingly need to capture and consume provenance information. As a result, Prov4J is targeted towards an application developer which needs to build provenance-aware applications.

According to Freire et al. [2], provenance management frameworks typically consist of three main components: a capture mechanism, a representational model and an infrastructure for storage, access and queries (provenance consumption). In Prov4J, the representational model is covered by the W3P provenance ontology[5], the capture mechanism is covered by *ProvLogger*, the component which is responsible for logging and publishing, and the provenance consumption is done by the *ProvClient* component.

W3P is a generic provenance ontology for tracking provenance on the Web. W3P is designed to be a lightweight provenance ontology, complementing and integrating vocabularies such as Dublin Core[6], and the ChangeSet[7] vocabulary. Other key features of W3P include the coverage of social provenance [3] and the maximization of the compatibility with the Open Provenance Model (OPM). Prov4J uses W3P as its default provenance model. Sections V and VI approach the consumption and capture cycles in Prov4J.

IV. ARCHITECTURE

In most applications, provenance represents a cross-cutting concern where the functionalities to capture and consume provenance are a complementary requirement to the core functionalities of an application. A cross-cutting concern is a common feature that is typically spread across objects in the application, being difficult to decompose from other parts of the system. Prov4J adopts a provenance architecture which reflects the separation between the core concerns of the application and the cross-cutting concern of provenance. The architecture maximizes the encapsulation of provenance capture and consumption functionalities in a separate layer (figure 1). The architecture behind Prov4J contains many elements in common with the general architecture proposed by Groth [4].

However, differently from classical examples of cross-cutting concerns (e.g. message logging and user authentication/access control), provenance capture and consumption is typically more tightly coupled with the logic structure of the calling application (process documentation perspective [2]) or to the data used in the application (data provenance perspective [5]), bringing challenges and practical limits to the isolation of the provenance concern inside the calling application.

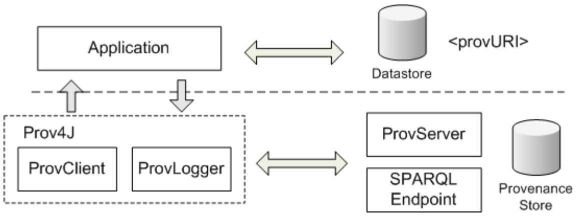


Figure 1: Generic provenance management architecture.

A common provenance scenario is the association of the information present in a procedural or object-oriented application to a data artifact in a generic data store (e.g. relational databases, XML or RDF data). The strategy used in Prov4J is to use RDF to represent provenance data and URIs to associate the described information resource in the core application layer with its provenance descriptor. This allows Prov4J to cope with both *data representation independency* and *separation of concerns*, important requirements for a generic provenance framework. A <*provURI*> is a connection point between the core application layer and the provenance layer, being an entry point into the provenance store. This allows an abstraction over the artifact type, which can be a relational tuple, RDF triples, a named graph, a XML element, a HTML element, etc.

The <*provURI*> mechanism also allows Prov4J to partially track data provenance. Data provenance is defined as the process of tracking the origins of data and its movement between databases [6]. Compared to the perspective of workflow provenance, data provenance approaches the problem under a database perspective, focusing on the relationships between data artifacts. A typical problem in this perspective is the representation of dependencies between data artifacts (i.e. on which artifacts a specific piece of data depends upon). Despite the fact that a complete data provenance tracking solution is highly dependent on the storage mechanism, Prov4J provides a basic functionality for mapping dependencies across data artifacts. This discussion is briefly detailed in section VI.

Prov4J also allows the discovery and consumption of provenance descriptors associated with different types of resources, including HTML pages, SPARQL endpoints, and RDF published as Linked Data. This allows Prov4J to respond to an important use case where an application is consuming provenance from third-party Web resources. Prov4J consists of two core components: *ProvClient* and *ProvLogger* (figure 2). ProvClient is responsible for the consumption cycle of the application, while ProvLogger provides an interface for provenance capture. A third element, *ProvServer*, is introduced in order to allow high performance provenance capture.

V. PROVENANCE CONSUMPTION

A. *Description*

The consumption cycle inside Prov4J starts with the specification of the information sources which will be

[5] http://prov4j.org/w3p/schema#
[6] http://dublincore.org
[7] http://vocab.org/changeset/schema.html

consumed: users can specify the location (URIs) of information resources that have associated provenance descriptors or the URIs of provenance data sources. There are three types of supported provenance sources: provenance stores (which are SPARQL endpoints), linked provenance data (RDF published using the Linked Data principles[8]) and provenance descriptors (RDF data embedded in different formats).

Each type of provenance data source has a different consumption approach. Data in provenance stores are consumed after a user query is defined over the API. Linked provenance data is consumed using a navigational approach [7], where provenance is queried by successive navigation over the provenance graph (de-referencing each of the provenance entities and loading the returned RDF into a memory model). Figure 2 shows the basic components inside the framework including the components for provenance discovery and RDF extraction under different publication protocols (Provenance Discovery and Parsers), the components for Linked Provenance Data navigation (Linked Data Navigator) and provenance store data consumption (Client).

The provenance graphs collected from different sources are then loaded into a memory model. The framework uses two basic internal provenance structures: a provenance graph (*ProvGraph*) and a provenance view (*ProvView*). A ProvGraph represents the basic fragment of provenance information associated with a data source. One or more different ProvGraphs can be loaded into a single model by using a ProvView. The ProvView is the model where users have a consolidated provenance view over a set of different provenance data sources.

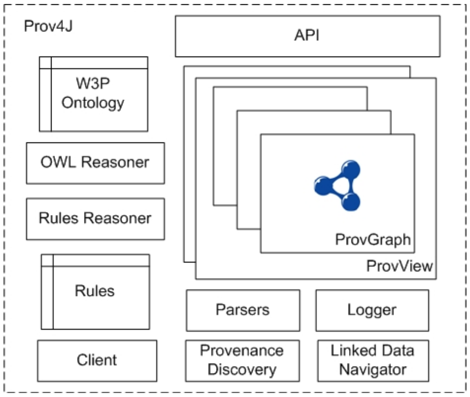


Figure 2: Prov4J key components.

After all provenance graphs are merged into provenance views, elements from different vocabularies are mapped into W3P entities using rules reasoning. Rules provide an expressive mechanism which allows complex mappings between different vocabularies which cannot be addressed by *owl:equivalentClass* or *owl:equivalentProperty*. Examples of more complex mappings across different provenance representations can be found in Miles [8], which defines OPM Profiles for Dublin Core vocabulary elements. The use of rules for vocabulary mappings also allows the representation of the mappings in standardized representations such as SWRL[9].

Once the vocabularies are mapped into W3P elements, the framework applies RDFS/OWL reasoning over the provenance model. *owl:TransitiveProperty* and *owl:InverseProperty* are used in W3P to improve the number of provenance queries answered by the framework (*subsection B* in this section). The consumption process is dependent on the type of provenance data source: for provenance stores and linked provenance data, the reasoning is done at query time, while for descriptors the *discovery-parsing-reasoning* is done during the definition of data sources on the interface. Prov4J uses the Jena framework[10] in its core and Pellet [9] is used for both OWL and Rules reasoning. Users can disable both types of reasoning from the API.

B. *Provenance Queries*

The provenance consumption API (ProvClient) provides the core operations over the provenance views. The ProvClient API contains key interface methods for a set of provenance queries, minimizing the interaction from users with SPARQL. A SPARQL query interface is also exposed to allow non-predefined types of queries over the model. The framework supports five query categories:

SPARQL based queries: Provenance queries supported by the elements of the SPARQL specification[11] are accessible by using the direct query over the provenance model or by using API methods for common queries. Prov4J SPARQL also includes syntactic extensions: GROUP BY, HAVING and aggregation. ARQ with syntactic extensions[12] is the core query engine behind Prov4J.

Queries supported by reasoning: Some key provenance queries over W3P can be addressed by applying OWL reasoning over provenance data. Examples of queries of this type involve the determination of indirect relationships/dependencies in a workflow chain, such as "*list all artifacts which were used directly or indirectly in artifact X*". Similarly, rules can be applied to improve query expressivity.

Path queries: One important feature for provenance queries is the ability to query paths over provenance trails. A typical path query is "*show all the processes between artifacts A and B*" or more specifically "*list all the trails containing a process which uses artifact C between artifacts A and B*". Regular expressions queries over RDF elements can be used for expressing provenance path patterns. Prov4J uses the Gleen SPARQL extension described in [10] for path queries. Prov4J users can launch their own path queries or can access some of the functionalities provided by Gleen through API methods.

[8] http://www.w3.org/DesignIssues/LinkedData.html

[9] http://www.w3.org/Submission/SWRL/

[10] http://jena.sourceforge.net

[11] http://www.w3.org/TR/rdf-sparql-query/

[12] http://jena.sourceforge.net/ARQ/extension.html

Navigational queries: In some scenarios the primary way to consume provenance information is through RDF published as Linked Data. In this case Prov4J provides two interfaces: one for users browsing provenance data and the other for navigational queries. In the first case, the first level provenance descriptor of an artifact is available as a de-referentiable URI. The RDF provenance data can be consumed by the application and further de-referentiations are directed by user input (in this case Prov4J provides a simple interface for node de-referentiations). A second type of navigation provided by the framework is through the provision of iterators to provenance nodes where provenance properties are used to determine which provenance nodes to de-reference. For example, the iterator defined by the property *w3p:used* can be used to navigate through a chain of artifact dependencies. The third functionality is defined by the idea of navigational queries, which are mechanisms to query Linked Data by launching a SPARQL query over a collection of RDF graphs collected from a de-referentiable URI entry point [7]. In the case of Prov4J, a simple de-referentiation algorithm follows the provenance links until it reaches a pre-configured limit.

Similarity queries: One type of provenance query refers to the similarity analysis between two provenance graphs. This type of comparison can be used in the determination of similar workflow conditions and have potential applications in quality assessment scenarios. A user may trust a specific workflow and may want to query for similar or identical conditions. The matching process used in Prov4J is based on the approach described by Oldakowsky & Bizer [11] adapted to the W3P provenance model.

C. Provenance Discovery

Provenance Discovery consists in automatically discovering the provenance given an information resource and it is an important requirement for a generic provenance management framework for consuming provenance data on the Web. Information resources can be HTML pages, elements inside the page, SPARQL endpoints, RDF files or de-referentiable URIs. A provenance discovery mechanism should not rely on centralized crawled provenance repositories: it should always be possible to navigate from the artifact to its provenance descriptor. Prov4J supports four mechanisms to discover provenance on the Web:

Semantic Sitemaps + robots.txt: Used to discover the provenance descriptor of a dataset having as a starting point a domain name. As covered in [12], the mechanism used by voiD [13], using *robots.txt* and the *semantic sitemaps extension*, can be used to discover dataset provenance descriptors.

Linked Provenance Data: Provenance descriptors can be published as Linked Data in two ways: (1) the URI represents an artifact and links directly to other provenance properties, (2) a provenance property such as *w3p:provenance* links the URI to the starting point of a provenance descriptor (a mirror to the provenance layer representation of the artifact).

Embedded RDFa: Provenance data can be embedded as RDFa in HTML pages.

POWDER: POWDER (Protocol for Web Description Resources) [13] is a W3C recommendation which provides a standard for describing general Web resources. Provenance descriptors can be embedded as RDF payloads in POWDER files.

VI. Provenance Capture

One key challenge in the process of building a generic provenance capture framework is the process of providing a simple yet expressive provenance interface. The ability to express provenance accurately and with the mimimum amount of intervention in the application is a fundamental feature in the process of introducing the provenance functionality in existing applications. In order to achieve this objective, the provenance capture engine was built using the following principles:

Pushback capture: Provenance capture or logging can be implemented as pushback operation, where, from the capture interface perspective, new provenance information is inserted but never deleted or updated. This assumption is consistent with the fact that provenance maps to the actual temporal execution flow of the application. Instead of allowing a full interaction with the provenance store, the ProvLogger interface is primarily designed for pushing back fragmented provenance logs, which are reconstructed in the provenance store (concept present in [4] and [14]).

Minimization of adaptations: Prov4J capture interface can be used to implement *adaptations*, a concept defined by Munroe [14], in a software engineering methodology designed for the development provenance-aware applications (PrIMe). Adaptations allow actors to record process documentation, adding the provenance functionality to the application. Relations among *entities* in the provenance model can be determined based on the *execution scope* of these elements. Temporal relations, order relations, relationships between agents, processes and artifacts in the same execution scope are examples of provenance data which can be determined without explicit *adaptations*. The ProvLogger component minimizes the user input in the construction of the provenance model, *'filling the gaps'* in the provenance model. Figure 3 shows examples of adaptations.

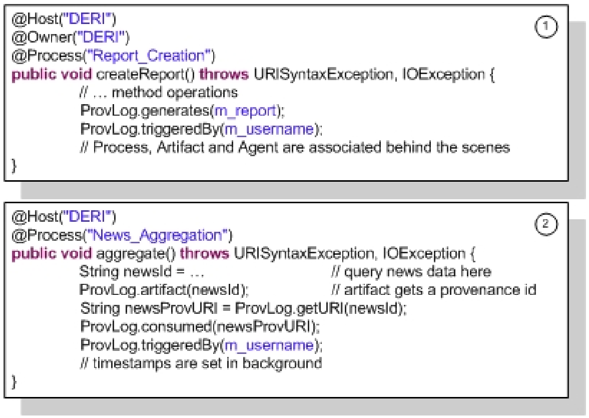
Provenance URIs: In some cases, provenance entities can be interconnected with elements in different parts of the workflow (e.g. a process consuming an artifact that was generated by another process at a different time). The logger interface provides a mechanism to interconnect provenance entities in different execution scopes. Users can associate different provenance entities by using internally the concept of ApplicationId-URI mapping, which associates Ids inside the application to provenance URIs. These associations can also be done directly by referencing directly provenance URIs. To

[13] http://www.w3.org/TR/powder-dr/

minimize the performance impact, this mechanism relies in the construction of a provenance URI cache in the capture mechanism.

Annotations: Java Annotations provide a mechanism to map the structure of an application to provenance elements. Annotations also allow users to provide provenance relationships valid in a specific scope. Provenance entities inside the scope of a method may be directly associated with an entity represented in the annotation, depending upon their relationship (figure 3). The design of ProvLogger allows users to express provenance information by maximizing the mapping between the application object structure and the provenance elements. In this case, classes, methods and member variables can directly map to provenance artifacts, processes and agents by using annotations.

The ProvLogger interface uses the concepts of aspect oriented programming (AOP) and Java annotations to maximize the isolation between cross-cutting concerns, allowing users to separate distinct functionalities of a software. AOP combined with Java Annotations can provide a powerful mechanism to implement the separation of concerns in provenance capture.



```
@Host("DERI")
@Owner("DERI")
@Process("Report_Creation")
public void createReport() throws URISyntaxException, IOException {
            // … method operations
            ProvLog.generates(m_report);
            ProvLog.triggeredBy(m_username);
            // Process, Artifact and Agent are associated behind the scenes
}
```

```
@Host("DERI")
@Process("News_Aggregation")
public void aggregate() throws URISyntaxException, IOException {
            String newsId = …                       // query news data here
            ProvLog.artifact(newsId);               // artifact gets a provenance id
            String newsProvURI = ProvLog.getURI(newsId);
            ProvLog.consumed(newsProvURI);
            ProvLog.triggeredBy(m_username);
            // timestamps are set in background
}
```

Figure 3: Examples of adaptations using the capture interface.

After the information is collected in the capture interface, the provenance log information is sent to the *ProvServer* and translated into a SPARQL/Update query to the provenance store. Prov4J relies on Scribe [14], a high-performance logging mechanism for the communication and distribution of provenance logs.

VII. FRAMEWORK ANALYSIS

The framework was analyzed using the scenario described in section II. A provenance dataset was generated using real financial data aggregated from multiple data sources, which focused on news and opinions about businesses collected from the Web. These data elements defined the ground artifacts which were further aggregated, curated and analyzed in a financial analysis workflow simulator. The output of the workflow is a report for a specific company, which is a mash-up of business data[15]. The provenance of the final report and each of its artifacts is tracked down to their original sources. In the experiment, business reports were generated with data collected for 100/500/1000 companies with up to 100/500/1000 news respectively for each company. Details about the experiment are outlined in table I. The experiment sought to determine the performance of the framework in a realistic scenario.

Data set	*Reasoning level*	*# triples*	*min query (ms)*	*max query (ms)*	*Reasoning (ms)*
1000	voc	674.786	1,2	680,6	2.717,9
	voc+owl+rules	686.829	2,4	> 90.000	314.846,2
500	voc	231.217	1,2	246,1	959,8
	voc+owl+rules	234.572	1,1	22.445,4	44.536,9
100	voc	84.520	1,2	217,9	602,9
	voc+owl+rules	87.204	1,4	5.180,7	16.246,9

Table I: Prov4J performance metrics.

The experiment was run in a 2.53 GHz Intel Core 2 Duo computer with 4 GB of memory. The minimum level of reasoning enabled was vocabulary rules mapping (*voc*) and the maximum added OWL features and 5 additional rules (*voc+owl+rules*). The input provenance data was consumed from 2 RDF files which were merged inside the framework. In the experiment path-based queries showed the highest execution time (*max query*), being highly sensitive to reasoning. SPARQL queries over basic elements of the ontology accounted for the lowest execution time (*min query*). Most of the queries (aggregate included) showed low execution time increase after the reasoning was enabled. Navigational and similarity queries were not tested. The 1000/1000 dataset counted 53.844 processes, 20.179 artifacts and 30 agents. The framework was able to do reasoning and answer the majority of provenance queries present in the API with acceptable runtime latencies. The scalability of reasoning could, however, represent a problem for larger provenance datasets.

VIII. RELATED WORK

Different approaches for provenance management have been described in the literature. Pegasus [15] is a workflow-based system that uses both OWL and relational databases to represent provenance. ES3 [16] is an OS-based provenance system that represents provenance data in XML and provides query support through XQuery. PReServ [17] is a process-based provenance recorder that allows the integration of provenance into third-party applications. PReServ uses XML to persist provenance data; queries are provided through a Java query API and XQuery. In [18] Bochner et al. describe a python client library for provenance recording and querying in a PReServ store. Taverna [19] is a workflow-based system which represents provenance in both Scufl Model and prospective provenance in RDF. SPARQL is used as a query language. In [20], Sahoo et al. present PrOM, a Semantic Web provenance management framework focused on scalable querying for eScience. The reader is referred to [1, 2] for comprehensive surveys and analysis of existing provenance management systems.

[14] http://sourceforge.net/projects/scribeserver/

[15] http://prov4j.org/w3p/scenario/financial.html

Compared to existing works, Prov4J stands out as most heavily leveraging Semantic Web standards and tools, using RDF as its core provenance representation and both OWL and rules reasoning over provenance data. In addition, Prov4J extends existing SPARQL query capabilities: a Java API, SPARQL aggregate functions, regular expression path queries and similarity queries together with reasoning provide additional query expressivity. Prov4J also incorporates important requirements for the Web: provenance discovery and Linked Data navigation. Compared to PrOM, which is targeted towards the provision of a scalable query mechanism for an eScience scenario, Prov4J focuses on generic provenance management for the Web including both provenance capture and discovery. Similarly to the combination PreServ + Python Provenance Client Library, Prov4J is designed as an independent provenance layer, being designed for the provision provenance-awareness to generic applications.

In [12], Hartig and Zhao covers the main aspects of publishing and consuming provenance in the Web of Data using the Provenance Vocabulary. The provenance discovery mechanism behind Prov4J shares common aspects with the publication methodology described in their work. Additionally, the mapping mechanism behind Prov4J allows the framework to consume and query Provenance Vocabulary descriptors.

IX. Conclusion & Future Work

This work described Prov4J, a generic provenance management framework. The design of the framework focused on the following features: (1) the provision of expressive provenance queries; (2) the maximization of the use of Semantic Web standards to address the challenges of managing provenance data; (3) software engineering aspects for provenance capture; (4) discovery mechanisms for provenance descriptors on the Web. The use of Semantic Web tools and standards to address these challenges played a fundamental role in the construction of the framework. Prov4J benefited largely from: the use SWRL-like rules to map and align different provenance vocabularies; OWL reasoning to address a subset of provenance queries; use of different publishing protocols (POWDER, semantic sitemaps, etc) for provenance discovery on the Web; SWRL-like rules applied to the enrichment of the provenance structure; RDF to represent the bulk of provenance data, SPARQL as a query mechanism and Linked Data as a publication mechanism for the Web. The use of non-standardized extensions over existing standards such as aggregate SPARQL queries, SPARQL/Update and path queries provided important features for the framework. The ensemble of these technologies proved to achieve a good performance under a realistic provenance scenario.

From the software engineering perspective, Prov4J orchestrates different strategies to maximize the separation between provenance aspects and core concerns and to reduce the number of application adaptations for provenance capture.

Future work will include a detailed analysis of the query expressivity and query performance of the framework. A mapping mechanism from W3P to OPM profiles using rules is planned. One current limitation of the framework is related to the deployment of security and integrity mechanisms. In addition, an in-depth comparative study across existing provenance management systems is planned. Improvements over the framework to transform Prov4J from an experimental to a robust provenance solution are set as a priority.

Acknowledgment

The work presented in this article has been funded substantially by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2). The authors want to express their gratitude to Tomas Knap and Micah Herstand for their support in the elaboration of this work.

References

[1] Y. Simmhan, B. Plale, and D. Gannon, “A survey of data provenance in e-science,” *SIGMOD Record,* vol. 34, 2005, pp. 31-36.

[2] J. Freire, D. Koop, E. Santos, and C.T. Silva, “Provenance for Computational Tasks: A Survey,” *Computing in Science Engineering,* vol. 10, no. 3, 2008, pp. 11 -21.

[3] A. Harth, A. Polleres, and S. Decker, “Towards A Social Provenance Model for the Web,”, 2007.

[4] P. Groth, S. Jiang, S. Miles, S. Munroe, V. Tan, S. Tsasakou, and L. Moreau, “ An Architecture for Provenance Systems,”, ECS, University of Southampton., 2006.

[5] P. Buneman, S. Khanna, and W.C. Tan, “Why and Where: A Characterization of Data Provenance,” *ICDT*, pp. 316-330.

[6] P. Buneman, and W.C. Tan, “Provenance in databases,” *SIGMOD Conference*, pp. 1171-1173.

[7] P. Bouquet, C. Ghidini, and L. Serafini, “A Formal Model of Queries on Interlinked RDF Graphs,” *AAAI Spring Symposium Series*.

[8] S. Miles, L. Moreau, and J. Futrelle, “OPM Profile for Dublin Core Terms,” *Book OPM Profile for Dublin Core Terms*, Series OPM Profile for Dublin Core Terms, 2009.

[9] E. Sirin, B. Parsia, B. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A practical OWL-DL reasoner,” *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 5, no. 2, 2007, pp. 51-53.

[10] L.T. Detwiler, D. Suciu, and J.F. Brinkley, “Regular paths in SparQL: querying the NCI Thesaurus,” *AMIA Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2008, pp. 161-165.

[11] R. Oldakowski, and C. Bizer, “SemMF: A Framework for Calculating Semantic Similarity of Objects Represented as RDF Graphs (Poster).”, 2005.

[12] O. Hartig, and J. Zhao, “Publishing and Consuming Provenance Metadata on the Web of Linked Data,” *Book Publishing and Consuming Provenance Metadata on the Web of Linked Data*, Series Publishing and Consuming Provenance Metadata on the Web of Linked Data, 2010.

[13] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao, “Describing Linked Datasets,” *Proceedings of the Linked Data on the Web Workshop LDOW 2009*.

[14] S. Munroe, S. Miles, L. Moreau, and J. Vazquez-Salceda, “PrIMe:A software engineering methodology for developing provenance-aware applications,” *Foundations of Software Engineering*, 2006, pp. 39-39.

[15] J. Kim, E. Deelman, A. Gil, G. Mehta, and V. Ratnakar, “Provenance Trails in the Wings/Pegasus System”.

[16] J. Freire, D. Koop, L. Moreau, J. Frew, and P. Slaughter, “ES3: A Demonstration of Transparent Provenance for Scientific Computation,” *Provenance and Annotation of Data and Processes*, Lecture Notes in Computer Science 5272, Springer Berlin / Heidelberg, 2008, pp. 200-207.

[17] P. Groth, S. Miles, and L. Moreau, “PReServ: Provenance Recording for Service,”, 2005.

[18] J. Freire, D. Koop, L. Moreau, C. Bochner, R. Gude, and A. Schreiber, “A Python Library for Provenance Recording and Querying,” *Provenance and Annotation of Data and Processes*, Lecture Notes in Computer Science 5272, Springer Berlin / Heidelberg, 2008, pp. 229-240.

[19] T. Oinn et al., “Taverna: lessons in creating a workflow environment for the life sciences:Â Research Articles,” *Concurrency and Computation: Practice & Experience,* vol. 18, no. 10, 2006, pp. 1067-1067.

[20] S.S. Sahoo, R. Barga, A. Sheth, K. Thirunarayan, and P. Hitzler, “PrOM: A Semantic Web Framework for Provenance Management in Science,”, 2009.

Semantic Provenance Registration and Discovery using Geospatial Catalogue Service

Peng Yue[1], Jianya Gong[1], Liping Di[2], Lianlian He[3], Yaxing Wei[4]
[1] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University,
129 Luoyu Road, Wuhan, China, 430079
[2] Center for Spatial Information Science and Systems (CSISS), George Mason University,
4400 University Drive, MS 6E1, Fairfax, VA 22030, USA
[3] Department of Mathematics, Hubei University of Education, Nanhuan Road 1, Wuhan, Hubei, China, 430205
[4] Environmental Sciences Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6407, USA

***Abstract* – A geospatial catalogue service allows geospatial users to discover appropriate geospatial data and services in a Web-based distributed environment. Metadata for geospatial data and services is organized structurally in catalogue services. Provenance for geospatial data products, as a kind of metadata describing the derivation history of data products, can be managed in a same way as other kinds of metadata using metadata catalogue services, thus keeping consistency and interoperability with existing metadata catalogue services. Meanwhile, Semantic Web technologies have shown considerable promises for more effective connection, discovery, and integration of provenance information. This paper addresses how geospatial catalogue services can be enriched with semantic provenance. Semantic relationships defined in provenance ontologies are registered in an OGC standard-compliant CSW service by extending ebRIM elements. The work illustrates that such a semantically-enriched CSW can assist in the discovery of data, service, and knowledge level of geospatial provenance.**

Keywords: Data Provenance, Lineage, GIS, CSW, ebRIM, Geospatial Web Service

I. INTRODUCTION

The advancement of Earth observing technologies has significantly increased the capability for collecting geospatial data. The National Aeronautics and Space Administration (NASA)'s Earth Observing System (EOS) alone is generating 1000 terabytes annually [1]. Significant efforts have been devoted to make full use of the data and derive useful information from the raw data. The Open Geospatial Consortium (OGC)'s Web Service technologies such as the Web Feature Service (WFS), Web Map Service (WMS), and Web Processing Service (WPS) [2] have been widely used in geospatial domain to facilitate the open discovery of, access to, and processing of distributed geospatial data. A geospatial catalogue service allows geospatial users to discover appropriate geospatial data and services in a Web-based distributed environment. Metadata for geospatial data and services is organized structurally in catalogue services. The OGC's Catalogue Services for the Web (CSW) is a domain consensus regarding an open, standard interface for geospatial catalogue service [3].

Provenance for geospatial data products records the derivation history of the data products. In a service-oriented information infrastructure, geoprocessing steps in deriving a data product are usually implemented by chaining multiple geoprocessing services together. To derive useful data products from large volumes of raw data, the integration of geoprocessing services become more and more frequent. Provenance provides important context information to help end users make decisions about the quality of the derived data products. Semantic Web technologies provide ways to connect Web resources together and allow semantics of Web resources to be machine-understandable, thus enabling more effective discovery, automation, integration, and reuse of resources. Semantic provenance, provenance information represented using Semantic Web technologies, therefore, can provide more informed understanding and effective usage of provenance information.

In the geospatial domain, provenance information has been regarded as part of metadata describing data quality information in the International Organization for Standardization (ISO) 19115 geospatial information—metadata standard. Similar to other kinds of geospatial metadata managed using metadata catalogue services, provenance information can be registered and discovered in the metadata catalogue services to keep consistency and interoperability with legacy geographic information system (GIS) applications. The registration of provenance information in the catalogue services requires the specification of the registration information model. OGC has recommended the ebXML Registry Information Model (ebRIM) for registration of geospatial information, the so-called ebRIM profile of CSW [4]. However, the existing standard does not address the registration of provenance information.

This paper explores the use of OGC CSW for registration and query of semantic provenance. To make use of semantics for provenance discovery in CSW, semantic relationships defined in provenance ontologies are registered in an OGC standard-compliant CSW service by extending ebRIM elements. The work illustrates that such semantically-enriched CSW can assist in the discovery of data, service, and knowledge level of geospatial provenance. The rest of the paper is organized as follows. Section 2 introduces the semantic representation of provenance for geospatial data products. Section 3 describes the ebRIM-based information model in CSW, and Section 4 presents the registration of semantic provenance. Section 5 describes the provenance discovery using semantically-enriched CSW. The work is compared with related work in Section 6, and conclusions and pointers to future work are given in Section 7.

II. SEMANTIC PROVENANCE FOR GEOSPATIAL DATA PRODUCTS

In the context of this paper, we focus on the provenance in a service-oriented environment in which geospatial data products are generated by executing geoprocessing service chains. In the general information domain, service chaining is a hot research topic in the Web Service area and can be called service composition. Approaches for service composition generally follow a three-phase procedure [5-7]: (1) process modeling, which generates an abstract process model consisting of the control flow and data flow among process nodes; (2) process model instantiation, where the abstract process model is instantiated into an executable service chain; and (3) workflow execution, where the chaining result is executed in the workflow engine to generate the required data product. The information involved in the three phases, therefore, can contribute to the provenance of the data products.

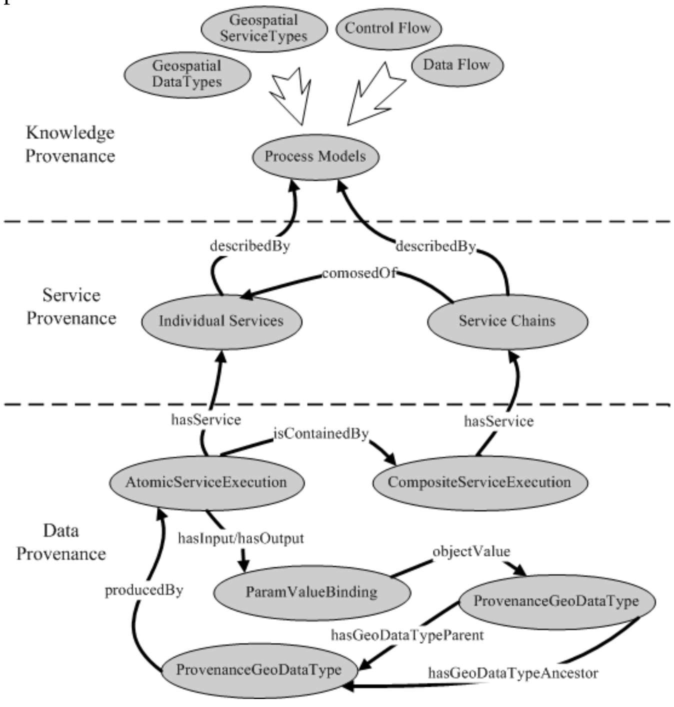


Figure 1. Semantic provenance for geospatial data products.

A three-level view of semantic provenance is adopted for the geospatial data products generated based on the three-phase procedure of service composition (Fig. 1). The first level is the knowledge level provenance, which contains process model ontologies as a knowledge base to support generation of complex process models. The process model ontologies are formulated by linking geospatial domain DataType, ServiceType, and workflow ontologies together. Examples of process model ontologies are atomic and composite process models for geoprocessing services described using the process model ontologies in the Web Ontology Language (OWL) Service Ontology (OWL-S). The second level is the service level provenance, which includes the individual services and service chains. Both can be represented using the service ontologies in OWL-S. And the final level is the data level provenance, which contains the provenance information generated during the execution. Examples of provenance in this level include source, intermediate, and final data products, atomic service executions, and service chain executions.

The ontologies for the knowledge level provenance and service level provenance use the geospatial domain ontologies and OWL-S ontologies. The data level provenance includes classes and relationships for data products required or generated by execution (ProvenanceGeoDataType class), value bindings between parameters and their values (ParamValueBinding class), specific executions of services (AtomicServiceExecution class) and service chains (CompositeServiceExecution class). Example ontologies in OWL can be viewed online at http://www.laits.gmu.edu/geo/nga/landslideprovenance.html .

The three-level view of geospatial provenance corresponds to the three phases of automatic service composition. The knowledge level provenance records the process model knowledge used to derive geospatial data products in the process modeling phase. Using provenance at this level, users can check the correctness of the process model and try a different model when necessary. The service level provenance describes concrete service chains that can be executed to generate the geospatial data products. Using this information, it is possible for users to re-select services based on the performance of services. The data level provenance helps users to find dependencies among physically-existed data products and supports analysis applications such as error source identification and propagation.

III. CSW-EBRIM PROFILE

CSW specification provides a framework for the implementation of application profiles. The core elements in an OGC catalogue service are the information model, the query language, and the interface [3]. The information model describes information structures and semantics of information resources. Therefore, the information model of catalogue services should address the content, syntax, and semantics of geospatial resources. The ebRIM standard has been defined by the Organization for the Advancement of Structured Information Standards (OASIS) and selected by OGC as the information model for specifying how catalogue content is structured and interrelated.

Fig. 2 shows the ebRIM-based catalogue information model. The core metadata class is the RegistryObject. Most other metadata classes in the information model are derived from this class. An instance of RegistryObject may have a set of zero or more Slot instances that serve as extensible attributes for this RegistryObject instance. An Association instance represents an association between a source RegistryObject and a target RegistryObject. Each association has an associationType attribute that identifies the type of that association. A Classification instance classifies a RegistryObject instance by referring to a node defined within a ClassificationScheme instance. A ClassificationScheme instance in the ebRIM model defines a tree structure made up of nodes that can be used to describe a taxonomy.

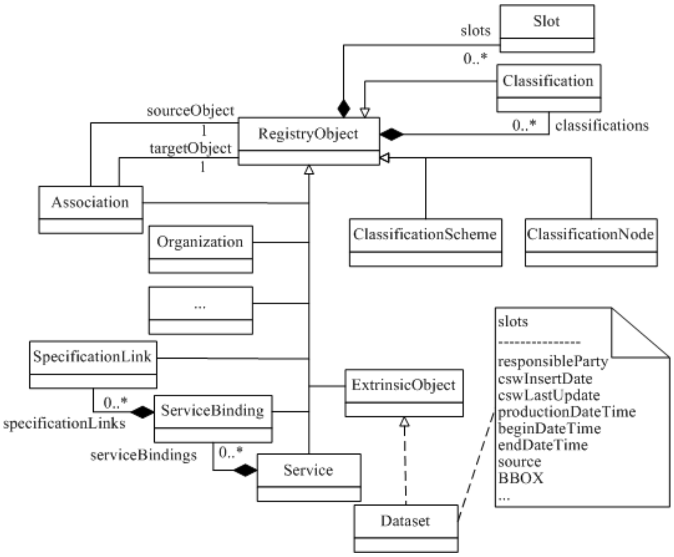


Figure 2. The ebRIM-based catalogue information model.

The ebRIM provides a general and standard metadata registration information model. However, it needs to be extended with some extension elements to meet common requirements in the geospatial domain. Under the guidelines of the ebRIM profile for CSW, the CSW implementation[1], developed and maintained by Laboratory for Advanced Information Technology and Standards (LAITS) from George Mason University [8], has extended ebRIM using international geographic standards: ISO 19115 Geographic Information — Metadata (including part 2: Extensions for imagery and gridded data) and ISO 19119 Geographic Information — Services.

The ebRIM is extended with ISO 19115 and ISO 19119 in two ways. The first is by importing new classes into the ebRIM class tree, deriving new metadata classes from existing ebRIM classes. The new Dataset class is used to describe geographic datasets. Many new attributes are added to the Dataset class based on ISO 19115 and its part 2. The second way to extend ebRIM is to use Slots to extend an existing class. The Service class included in ebRIM can be used to describe geographic services, but the available attributes in the class Service are not sufficient to describe geospatial Web services. New attributes derived from ISO 19119 are added to the Service class through Slots.

IV. SEMANTIC PROVENANCE REGISTRATION

The registration of semantic provenance in the CSW takes advantages of extensibility points in ebRIM. Such extensibility points include new kinds of classes, associations, classifications, and additional slots to record OWL classes, properties and related axioms. Some efforts have already addressed the registration of OWL-based ontologies in ebRIM [9-12]. In this study, we focus on the application and extension of ebRIM in the provenance registration. In particular, the paper explores how to register the OWL-based semantic provenance in the ebRIM-based catalogue information model to support the provenance discovery.

For the knowledge level and service level provenance, we adopt the previous approach on registration of OWL/OWL-S [13]. A new type of ExtrinsicObject, named ProcessModel, is created in the ebRIM model to describe process models. Geospatial DataType and ServiceType ontologies are recorded using two new ClassificationScheme instances, which can be used to classify the ProcessModel and Dataset instances. The Service class in the ebRIM model can be used to describe both services and service chains, since a service chain as a whole can act as a service. The semantics for inputs, outputs, preconditions and effects (i.e. IOPE semantics) are recorded by using slots.

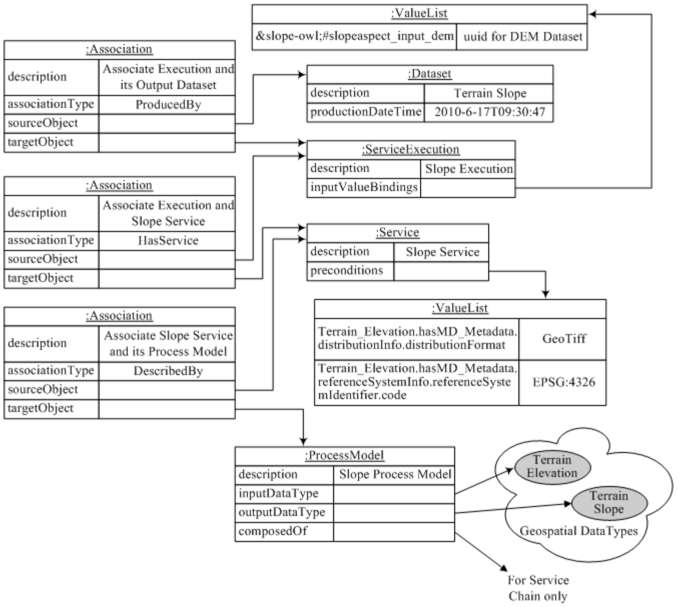


Figure 3. Associations among Dataset, Service, ServiceExecution, and Process Model.

For the data level provenance, a new type of ExtrinsicObject, ServiceExecution, which can support the registration of both atomic and service chain execution, is created. ProvenanceGeoDataType in OWL is mapped to the existing class Dataset. Individuals of ParamValueBinding in

[1] Online services are available at http://geobrain.laits.gmu.edu/ .

provenance ontologies are recorded using the slots of the ServiceExecution. The relationships among AtomicServiceExecution, CompositeServiceExecution, and ProvenanceGeoDataType in provenance ontologies are registered using associations in the ebRIM.

Fig. 3 shows an execution of slope computation service, which generates terrain slope data from the digital elevation model (DEM) data. The knowledge level provenance is recorded by using instances of ProcessModel whose slots specifies the input Geospatial DataType (Terrain Elevation) and output Geospatial DataType (Terrain Slope). The service level provenance is recorded using instances of Service. DescribedBy association connects a service with its process model. Some individual geospatial services have their own metadata constraints on the input data and this can be recorded using slots. For example, the slope computation service in Fig. 3 specifies that the input terrain elevation data should be in the GeoTIFF data format with the EPSG:4326 geographic coordinate reference system. Data level provenance includes the registration of ServiceExecution and Dataset. A ServiceExecution is linked to the service executed using the HasService association. The Terrain slope dataset generated by the specific ServiceExecution is described using the ProducedBy association. More kinds of associations can be registered such as the HasGeoDataTypeAncestor relationship between datasets.

V. PROVENANCE DISCOVERY

Based on the semantic content registered in the CSW, three types of provenance discoveries are achieved using CSW queries:

A. Discovery for data level provenance

The discovery is based on provenance associations at the data level. Examples of CSW queries includes: collecting descendant or ancestor datasets to a specific dataset; finding service executions to generate a specific dataset; retrieving parameters and values involved when conducing a specific service execution.

B. Discovery for service level provenance

One discovery is to locate services or service chains used to generate a specific geospatial data product. The query is based on the HasService association between service executions and services. Additional discovery includes query on the preconditions of a specific service. The results from this query can help check preconditions of the service to find whether input data is semantically valid. For example, does the input DEM data have a valid spatial projection?

C. Discovery for knowledge level provenance

This is to discover process model knowledge used to derive geospatial data products. The CSW query uses DescribedBy association as a search condition. The process model, when obtained, can be rechecked and compared with alternative process models. Another query strategy is to add semantically-matched ServiceTypes in the search condition to find alternate process models for decision support. The semantic match is performed based on the subsumption reasoning in description logic.

```
<?xml version="1.0" encoding="UTF-8"?>
<csw:GetRecords …>
  <csw:Query typeNames="ServiceExecution Association
Dataset ClassificationNode">
    <csw:ElementSetName>full</csw:ElementSetName>
    <csw:ElementName>/ServiceExecution/</csw:ElementName
>
    <csw:Constraint version="1.0.0"><ogc:Filter><ogc:And>
      <!--temporal condition-->…
      <!--spatial condition-->…
      <!—ontological concept-->
        <ogc:PropertyIsEqualTo><ogc:PropertyName>/Dataset/@i
d</ogc:PropertyName>
          <ogc:PropertyName>/Classification/@classifiedObject</
ogc:PropertyName></ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo><ogc:PropertyName>/Classificati
on/@classificationScheme</ogc:PropertyName>
          <ogc:PropertyName>/ClassificationScheme/@id</ogc:Pr
opertyName></ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>/ClassificationScheme/Description/
LocalizedString/@value</ogc:PropertyName>
          <ogc:Literal>geospatial data type ontology</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo><ogc:PropertyName>/Classificati
on/@classificationNode</ogc:PropertyName>
          <ogc:PropertyName>/ClassificationNode/@id</ogc:Prop
ertyName></ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo><ogc:PropertyName>/Classificati
onNode/@code</ogc:PropertyName>
      <ogc:Literal>ETM_NDVI</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <!--provenance association-->
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>/Dataset/@id</ogc:PropertyName>
        <ogc:PropertyName>/Association/@sourceObject</ogc:Pr
        opertyName></ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>/ServiceExecution/@id</ogc:Propert
yName>
        <ogc:PropertyName>/Association/@targetObject</ogc:Pro
pertyName></ogc:PropertyIsEqualTo>...
  </ogc:And></ogc:Filter></csw:Constraint></csw:Query>
  </csw:GetRecords>
```

Figure 4. Provenance query using CSW operation.

All queries are realized through CSW standard query operations. The query language is implemented using the OGC Filter specification. It supports comparison operators and spatial operators. An example provenance query is shown in Fig. 4. A Web client, e.g. HTML form, can submit queries using the GetRecords operation based on the request-response model of the HTTP protocol.

VI. RELATED WORK

A substantial research on provenance issue has been conducted in the general information domain. Traditional data provenance issue focuses on the database systems [14-16]. With the advancement of service-oriented infrastructure in recent years, provenance for scientific workflows or service chains becomes an active research field [17, 18]. The international workshop on data derivation and provenance and its follow-up workshops, namely International Provenance and Annotation Workshop (IPAW), have been held five times and resulted in the "provenance challenge" activities. Within GIS domain, how to incorporate provenance support in geospatial services is still a challenge. The use of OGC CSW for serving geospatial provenance is compliant with existing service standards in geospatial domain can allows easy integration with legacy GIS applications.

Some efforts have been devoted to the use of Semantic Web technologies for representing and querying data provenance information [19-22]. Our approach differs from their approaches in that we use existing registry services for management of provenance. The registration of ontologies in ebRIM can support semantics-enhanced discovery of information resources in registries [9-12]. The work here extends this approach in the provenance research area and proposes the registration of semantic provenance in the ebRIM model.

Provenance investigation in GIS can be traced back to Lanter's [23] work on data lineage metadata. Frew et al. [24] provide lineage support for remote sensing data processing in a script-based environment. Wang et al. [25] proposed a provenance-aware architecture to record the lineage of spatial data. Tilmes and Fleig [26] discuss some general concerns of provenance tracking for Earth science data processing systems. Plale et al. [27] described architectural considerations to support provenance collection and management in geosciences. Yue et al. [28] propose provenance capture in geospatial service composition when instantiating a geoprocessing model into an executable service chain. How provenance can be integrated into existing service-oriented GIS applications has not been addressed in the literature. In addition, the arrangement of provenance in the CSW-ebRIM profile facilitates the query of data, service, and knowledge level of provenance by exploring the associations among provenance, data, services, and chains.

VII. CONCLUSION AND FUTURE WORK

The ontology approach for provenance representation provides a common vocabulary for provenance information and defines explicitly the meaning of the terms and the relations between them. Registration of provenance ontologies in CSW allows users to take advantage of that benefit in registries. This paper describes how semantic provenance can be registered into the ebRIM-based CSW. Such a semantically-enriched CSW provides support in discovery of data, service, and knowledge level of geospatial provenance. Future work includes developing user-friendly tools to facilitate provenance registration and visualization of query results, exploring the lifetime management of provenance information, and developing provenance-aware applications to demonstration advantages and usage of provenance.

ACKNOWLEDGEMENT

We are grateful to the anonymous reviewers for their comments. This work was funded jointly by Project 40801153 supported by NSFC, 863 Program of China (2007AA120501), LIESMARS and SKLSE (Wuhan University) Special Research Fundings.

REFERENCE

[1] D. Clery and D. Voss, "All for one and one for all," Science, 308 (5723), p. 809, 2005.

[2] OGC, Open Geospatial Consortium, www.opengeospatial.org, [Accessed 16 May, 2010].

[3] D. Nebert, A. Whiteside, and P. Vretanos (eds), OpenGIS® Catalog Services Specification, Version 2.0.2, OGC 07-006r1, Open GIS Consortium Inc. 218 pp, 2007.

[4] R. Martell (ed), CSW-ebRIM Registry Service—Part 1: ebRIM profile of CSW, Version 1.0.0, OGC 07-110r2, Open Geospatial Consortium, Inc., 57 pp, 2008.

[5] B. Srivastava and J. Koehler, "Web service composition - current solutions and open problems," in: Proceedings of International Conference on Automated Planning and Scheduling (ICAPS) 2003 Workshop on Planning for Web Services, Trento, Italy, pp. 28-35.

[6] J. Rao and X. Su, "A survey of automated web service composition methods," in: Proceedings of First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, California, USA, pp. 43-54.

[7] J. Peer, Web service composition as AI planning - a survey, Technical report, University of St.Gallen, Switzerland, 63 pp, 2005.

[8] Y. Wei, L. Di, B. Zhao, G. Liao, A. Chen, Y. Bai, and Y. Liu, 2005. "The Design and Implementation of a Grid-enabled Catalogue Service," 25th Anniversary IGARSS 2005, July 25-29, COEX, Seoul, Korea. pp. 4224-4227.

[9] A. Dogac (ed.), ebXML Registry Profile for Web Ontology Language (OWL), Version 1.5, regrep-owl-profile-v1.5-cd01, Organization for the Advancement of Structured Information Standards (OASIS). 76 pp, 2006.

[10] A. Dogac, Y. Kabak, G.B. Laleci, C. Mattocks, F. Najmi, and J. Pollock, "Enhancing ebXML registries to make them OWL aware," Distributed and Parallel Databases Journal, Springer-Verlag, 18(1), pp. 9-36, July 2005.

[11] W. Liu, K. He, and W. Liu, "Design and realization of ebXML registry classification model based on ontology." In: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), 2005, pp. 809-814.

[12] A. Bechini, A. Tomasi, and J. Viotto, "Enabling ontology-based document classification and management in ebXML registries," In: Proceedings of the 2008 ACM symposium on Applied computing, Fortaleza, Ceara, Brazil, 2008, pp. 1145-1150.

[13] P. Yue, J. Gong, L. Di, L. He, and Y. Wei, "Integrating semantic web technologies and geospatial catalog services for geospatial information discovery and processing in cyberinfrastructure," GeoInformatica. 2009. DOI: 10.1007/s10707-009-0096-1.

[14] A. Woodruff and M. Stonebraker, "Supporting fine-grained data lineage in a database visualization environment," In: Proceedings of International Conference on Data Engineering (ICDE), 1997, pp. 91-102.

[15] Y. Cui, J. Widom, and J. L. Wiener, "Tracing the lineage of view data in a warehousing environment," ACM Transactions on Database Systems, 25(2), pp. 179-227, 2000.

[16] P. Buneman, S. Khanna, and W. C. Tan, "Why and where: a characterization of data provenance," In: Proceedings of International Conference on Database Theory (ICDT), 2001, pp. 316-330.

[17] Y. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," SIGMOD Record, vol. 34, pp. 31-36, 2005

[18] S. Miles, P. Groth, M. Branco, and L. Moreau, "The requirements of using provenance in e-Science experiments," Journal of Grid Computing, 5(1), pp. 1-25, 2007.

[19] J. Zhao, C. Goble, M. Greenwood, C. Wroe, and R. Stevens, "Annotating, linking and browsing provenance logs for e-Science," In: Proceedings of the Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, FL., USA, 6 pp, 2003.

[20] J. Golbeck and J. Hendler, "A semantic web approach to the provenance challenge," The First Provenance Challenge (this issue), Concurrency and Computation: Practice and Experience, 20 (5), pp. 431-439, 2007.

[21] S.S. Sahoo, A. Sheth, and C. Henson, "Semantic provenance for eScience: managing the deluge of scientific data," IEEE Internet Computing, 12 (4), pp. 46-54, 2008.

[22] S. Zednik, P. Fox, D. L. McGuinness, P. P. da Silva, and C. Chang, "Semantic provenance for science data products: application to image data processing," in: Proceedings of the First International Workshop on the Role of Semantic Web in Provenance Management (SWPM 2009), Washington DC, USA, CEUR-WS, vol. 526, October 25 2009, 7 pp.

[23] D. P. Lanter, "Design of a lineage-based meta-data base for GIS," Cartography and Geographic Information Systems, vol. 18, No. 4, pp. 255-261, 1991.

[24] J. Frew, D. Metzger, and P. Slaughter, "Automatic capture and reconstruction of computational provenance," Concurrency and Computation: Practice and Experience. 20(5). John Wiley & Sons, Ltd. pp. 485-496, 2007.

[25] S. Wang, A. Padmanabhan, D. J. Myers, W. Tang, and Y. Liu, "Towards provenance-aware geographic information systems," In: Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems (ACM GIS 2008). 4pp, 2008.

[26] C. Tilmes and J. A. Fleig, "Provenance tracking in an earth science data processing system," In: Proceedings of Second International Provenance and Annotation Workshop (IPAW), LNCS 5272, pp. 221-228, 2008.

[27] B. Plale, B. Cao, C. Herath, and Y. Sun, "Data provenance for preservation of digital geoscience data," Geological Society of America (GSA), Memoir Volume 12, 14pp, 2010.

[28] P. Yue, J. Gong, and L. Di, "Augmenting Geospatial Data Provenance through Metadata Tracking in Geospatial Service Chaining," Computers & Geosciences, vol. 36, no. 3, pp. 270-281, 2010

Towards Interoperable Metadata Provenance

Kai Eckert, Magnus Pfeffer
University Library
University of Mannheim
Mannheim, Germany
Email: eckert/pfeffer@bib.uni-mannheim.de

Johanna Völker
KR&KM Research Group
University of Mannheim
Mannheim, Germany
Email: voelker@informatik.uni-mannheim.de

***Abstract*—Linked data has finally arrived. But with the availability and actual usage of linked data, data from different sources gets quickly mixed and merged. While there is a lot of fundamental work about the provenance of metadata and the commonly recognized demand for expressing provenance information, there still is no standard or at least best-practice recommendation. In this paper, we summarize our own requirements based on experiences at the Mannheim University Library for metadata provenance, examine the feasibility to implement these requirements with currently available (de-facto) standards, and propose a way to bridge the missing gaps. By this paper, we hope to obtain additional feedback, which we will feed back into ongoing discussions within the recently founded DCMI task-group on metadata provenance.**

I. INTRODUCTION

At the Mannheim University Library (MUL), we recently announced a Linked Data Service[1] (LDS). Our complete catalog with about 1.4 million is made available as RDF, with proper dereferenceable URIs and a human-readable presentation of the data as HTML pages. The title records are linked to classification systems, subject headings and to other title records. The Cologne University Library made its catalog data available under a creative commons CC-0 license, so we converted it to RDF and made it available along our own catalog.

The HTML view[2] provides browsable pages for all resources described in the RDF data. It fetches additional statements when users click on the URIs, provided that they are available by URI dereferencing. The resulting statements are presented to the user within the LDS layout and cannot be easily distinguished from the data that is made available by the Mannheim University Library itself. There is only a note about the "data space", basically indicating the domain where the dereferenced URI resides.

A good thing is that the service is totally *source-agnostic* and fetches and presents everything that is available. With two clicks, the user gets subject data from the library of congress (LoC), just because we use the German subject headings and the German National Library (Deutsche Nationalbibliothek, DNB) provides skos:match statements to LoC subject headings (LCSH).

A bad thing is that the service is totally *source-agnostic* (apart from the data-space notion). For example, the DNB states on its website that the data is provided only as a prototype, should only be used after a consultation and not for commercial applications. The LCSH data is public domain and freely available. But also within our triple store, there are different datasets. The MUL catalog is currently provided without a specific license, as questions about the proper licensing still are discussed. The data from the Cologne University Library has been processed by us and the processed data is provided by a the creative commons CC-0 license, too.

A. Motivation

Our predicament: We do want the LDS to be source-agnostic. But at the same time we want to know about the license of the data that is displayed to the user, and we want to present him with this information. Moreover, besides license and source information, we also have other information that we would like to make available to the user or other applications in a reusable way. But the current state of the art is that this information is either not made available within the RDF datasets yet – the case for DNB, LoC and our own data – or not in a consistent way. For example, the data from the OCLC service dewey.info[3] contains licensing statements as part of the RDF statements about a given resource (Ex. 1).

```
<http://dewey.info/class/641/2009/08/about.en>
 a <http://www.w3.org/2004/02/skos/core#Concept>;
 xhv:license
   <http://creativecommons.org/licenses/by-nc-nd/3.0/>;
 cc:attributionName
   "OCLC Online Computer Library Center, Inc.";
 cc:attributionURL <http://www.oclc.org/dewey/>;
 ...
 skos:prefLabel "Food & drink";
 skos:broader
   <http://dewey.info/class/64/2009/08/about.en>;
 cc:morePermissions
   <http://www.oclc.org/dewey/about/licensing/>.
```

Example 1: Provenance in dewey.info dataset

[1] http://data.bib.uni-mannheim.de

[2] currently implemented with Virtuoso RDF-Mapper

[3] http://dewey.info

As another example, the New York Times expresses provenance outside the actual data record, more precisely by means of statements about the data record (Ex. 2).

```
<http://data.nytimes.com/46234942819259373803.rdf>
  foaf:primaryTopic
    <http://data.nytimes.com/46234942819259373803>
  dcterms:rightsHolder
    "The New York Times Company"
  ...
  cc:license
    <http://creativecommons.org/licenses/by/3.0/us/>
  cc:attributionURL
    <http://data.nytimes.com/46234942819259373803>
  cc:attributionName
    "The New York Times Company"

<http://data.nytimes.com/46234942819259373803>
  ...
  a <http://www.w3.org/2004/02/skos/core#Concept>
  ...
  skos:prefLabel "Faircloth, Lauch"
  ...
```

Example 2: Provenance in New York Times dataset

Our goal is to make this kind of information available to the user in a *consistent* way. We respect all the different licenses and do not want to make users believe that all this data is provided by ourselves, without any licensing information.

Besides provenance information, we also need to provide other information that further qualifies single statements of the datasets. For example, in a past project we automatically created classifications and subject headings for bibliographic resources. We provide this data also via the LDS which is very convenient and greatly facilitates the reuse of the data. But automatically created results often lack the desired quality, moreover the processes usually provide further information, like a weight, rank or other measures of confidence [1]. All this information should also be provided to the user in a well-defined way.

B. Data, Metadata, Metametadata, ...

Data provided as RDF is not necessarily metadata in a strict sense; in general it is data about resources. But in many cases – and especially in the context of this paper – the resources are data themselves, like books, articles, websites or databases. In the library domain, the term **"metadata"** is thus established for all the data about the resources a librarian is concerned with – including, but not restricted to bibliographic resources, persons and subjects. This is the reason, why one cannot distinguish easily between data and metadata in the context of RDF. We therefore regard them as synonyms.

Metadata is itself data and there are a lot of use-cases where one wants to make further statements about metadata, just as well as metadata provides statements about data: who created the metadata, how was the metadata created, ... – in general additional statements to further qualify and describe the metadata. Thus we will refer to this kind of additional information unambiguously as **"metametadata"**.

C. Metametadata Principles

To achieve interoperability for accessing metametadata, choosing a representation of the metametadata is only the first, merely technical step. In our opinion, the following principles and requirements have to be met to achieve this type of interoperability:

1) Arbitrary metametadata statements about a set of statements.
2) Arbitrary metametadata statements about single statements.
3) Metametadata on different levels for each statement or sets of statements.
4) Applications to retrieve, maintain and republish the metametadata without data loss or corruption.
5) Data processing applications to store the metametadata about the original RDF data.

Requirements 1 - 3 address the technical requirements that have to be met by the metadata format(s) in use. They are met by RDF, but in RDF there are two distinct approaches that can be used to represent metametadata:

Reification: RDF provides a means for the formulation of statements about statements, called *reification*. In the RDF model, this means that a complete statement consisting of subject, predicate and object becomes the subject of a new statement that adds the desired information.[4]

Named Graphs: Another technique that can be used to provide statements about statements are the "Named Graphs", introduced by Carroll et al. [2]. The Named Graphs are not yet officially standardized and part of RDF. They have to be considered work in progress, but are already widely used by the community and can already be considered as a kind of de-facto standard that is likely to have a big impact on future developments in

[4] As a statement cannot be identified uniquely in RDF beside the notion of S, P and O, a reification statement refers to *all* triples with the given S, P and O. In our context, this ambiguity has no substantial effects, as identical triples are semantically equivalent to duplicated metadata that can be safely discarded as redundant information.

the RDF community.[5] Named Graphs are an extension of RDF, both on the model and syntax level. They allow the grouping of RDF statements into a graph. The graph is a resource on its own and can thus be further described by RDF statements, just like any other resource. There are extensions for SPARQL and N3 to represent and query Named Graphs, but they are for example not representable in RDF-XML.[6]

To meet requirements 4 and 5, further conventions among interoperable applications are needed that have to be negotiated on a higher level and are (currently) beyond the scope of RDF. By virtue of the following use-cases, we demonstrate that the technical requirements are already met and that we only need some conventions to represent such information in an consistent way – at least as long as the official RDF standard does not address the metametadata issue.

II. Example Use-Cases

The following use cases[7] are meant to be illustrating examples, especially to emphasize the need for the representation of arbitrary information – not only provenance – about data on various levels, from whole datasets over records to single statements or arbitrary groups of statements.

In this section, we develop a scenario where such metametadata can be used to prevent information loss while merging subject annotations from different sources. We show that this is the key to make transparent use of different annotation sources without compromises regarding the quality of your metadata. In line with our argumentation in this paper, we propose the storage of metametadata to mitigate any information loss and allow the usage of this information to achieve a better retrieval experience for the users. With various queries, we show that we can access and use the additional pieces of information to regain a specific set of annotations that fulfills our specific needs.

This scenario focuses on the merging of manually assigned subject headings with automatically assigned ones. Example 3 shows a DC metadata record with subject annotations from different sources and additional information about the assignments via RDF reification. Note that we present the triples in a table and give them numbers that are then used to reference them.

	Subject	*Predicate*	*Object*
1	ex:docbase/doc1	dc:subject	ex:thes/sub20
2	#1	ex:source	ex:sources/autoindex1
3	#1	ex:rank	0.55
4	ex:docbase/doc1	dc:subject	ex:thes/sub30
5	#4	ex:source	ex:sources/autoindex1
6	#4	ex:rank	0.8
7	ex:docbase/doc1	dc:subject	ex:thes/sub30
8	#7	ex:source	ex:sources/pfeffer
9	#7	ex:rank	1.0
10	ex:docbase/doc1	dc:subject	ex:thes/sub40
11	#10	ex:source	ex:sources/pfeffer
12	#10	ex:rank	1.0
13	ex:sources/autoindex1	ex:type	ex:types/auto
14	ex:sources/pfeffer	ex:type	ex:types/manual

Example 3: Subject assignments by different sources

There is one document (*ex:docbase/doc1*) with assigned subject headings from two different sources. For each subject assignment, we see that a source is specified via a URI. Additionally, a rank for every assignment is provided, as automatic indexers usually provide such a rank. For example, a document retrieval system can make direct use of it for the ranking of retrieval results. For manual assignments, where usual no rank is given, this could be used to distinguish between high quality subject assignments from a library and, for example, assignments from a user community via tagging.

The statements #13 and #14 are used to further qualify the source, more precisely, to indicate, if the assignments were performed manually (*ex:types/manual*) or automatically (*ex:types/auto*).

A. Use-case 1: Merging annotation sets

Usually, the statements from Example 3 are available from different sources (as indicated) and might also belong to different shells in the shell model. The integration requires to merge them in a single store. An interesting side-effect of the use of RDF and reification is that the merged data is still accessible from every application that is able to use RDF data, even if it is not possible to make reasonable use of our metametadata. This is demonstrated by the first query in Example 4, which retrieves all subject headings that are assigned to a document. As in RDF all statements are considered identical that have the same subject, predicate and object, every subject heading is returned that is assigned by at least one source. In most cases, these completely merged statements are not wanted. As promised, we show with the second query in Example 4 that we are able to regain all annotations that were assigned by a specific source (here *ex:sources/pfeffer*).

[5] See http://www.w3.org/2004/03/trix/ for a summary. There are already further extensions or generalizations of Named Graphs, like Networked Graphs [3] that allow the expression of views in RDF graphs in a declarative way. Flouris et al. propose a generalization to maintain the information associated with graphs, when different graphs are mixed [4]: Here, colors are used to identify the origin of a triple, instead of names. A notion of “Color1+Color2” is possible and the paper demonstrates, how reasoning can be used together with these colored triples. Gandon and Corby published a position paper [5] about the need for a mechanism like Named Graphs and a proper standardization as part of RDF.

[6] You can see the grouping of statements in a single RDF-XML file as the notion of an implicit graph and use the URI of the RDF-XML file to specify further statements about this graph, just like Ex. 2

[7] First published at the DC 2009 conference [6].

```
SELECT ?document ?value WHERE {
   ?t rdf:subject ?document .
 ?t rdf:predicate dc:subject .
 ?t rdf:object ?value .
}
```

document	*subject*
ex:docbase/doc1	ex:thes/sub40
ex:docbase/doc1	ex:thes/sub30
ex:docbase/doc1	ex:thes/sub20

```
SELECT ?document ?value WHERE {
   ?t rdf:subject ?document .
 ?t rdf:predicate dc:subject .
 ?t rdf:object ?value .
 ?t ex:source <ex:sources/pfeffer> .
}
```

document	*subject*	*source*
ex:docbase/doc1	ex:thes/sub40	ex:sources/pfeffer
ex:docbase/doc1	ex:thes/sub30	ex:sources/pfeffer

Example 4: Querying the merged statements

B. Use-case 2: Extended queries on the merged annotations

In the following we show two extended queries that make use of the metametadata provided in our data store. Usually, one does not simply want to separate annotation sets that have been merged, but instead wants to make further use of these merged annotations. For example, we can provide data for different retrieval needs.

The first query in Example 5 restricts the subject headings to manually assigned ones, but they still can originate from different sources. This would be useful if we are interested in a high retrieval precision and assume that the results of the automatic indexers decrease the precision too much.

The second query, on the other hand, takes automatic assignments into account, but makes use of the rank that is provided with every subject heading. This way, we can decide to which degree the retrieval result should be extended by lower ranked subject headings, be they assigned by untrained people (tagging) or some automatic indexer.

III. Related Work

Early initiatives to define a vocabulary and usage-guidelines for the provenance of metadata was the A-Core [7] and based on it the proposal [8] for the DCMI Administrative Metadata Working Group (http://dublincore.org/groups/admin/). The working group finished its work in 2003 and presented the Administrative Components (AC) in [9], that addressed metadata for the entire record, for update and change and for batch interchange of records. Both initiatives focused more on the definition of specific vocabularies to describe the provenance of metadata. There was not yet a concise model to relate the metametadata with the metadata. For example, there was only an example given, hot to use the AC in an XML representation. This is not enough to enable applications the automatic integration of these information without proper knowledge, how the information is actually represented from a data model perspective.

```
SELECT DISTINCT ?document ?subject WHERE {
 ?t rdf:subject ?document .
 ?t rdf:predicate dc:subject .
 ?t rdf:object ?subject .
 ?t ex:source ?source .
 ?source ex:type ?type .
 FILTER ( ?type = <ex:types/manual>  )
}
```

document	*subject*	*type*
ex:docbase/doc1	ex:thes/sub40	http://example.org/types/manual
ex:docbase/doc1	ex:thes/sub30	http://example.org/types/manual

```
SELECT DISTINCT ?document ?subject WHERE {
 ?t rdf:subject ?document .
 ?t rdf:predicate dc:subject .
 ?t rdf:object ?subject .
 ?t ex:source ?source .
 ?source ex:type ?type .
 ?t ex:rank ?rank .
 FILTER ( ?type = <ex:types/manual> || ?rank > 0.7 )
}
```

document	*subject*	*rank*
ex:docbase/doc1	ex:thes/sub40	1.0
ex:docbase/doc1	ex:thes/sub30	1.0
ex:docbase/doc1	ex:thes/sub30	0.8

Example 5: Ranked assignments and additional source information

An implementation with a clear semantic of metadata provenance statements is included in the protocol for metadata harvesting by the The authors (Rephrase with cite) in [10] (OAI-PMH). But the provenance information can only be provided for a whole set of metadata and there is no easy way to extend it with other additional information. The Open Archives Initiative provides with Object Reuse and Exchange (ORE) another, more abstract approach that addresses the requirement of provenance information for aggregations of metadata [11]. ORE particularly introduces and motivates the idea to give metadata aggregations specific URIs to identify them as independent resources. Essentially, ORE postulates the clear distinction between URIs identifying resources and URIs identifying the description of the resources. This is in line with the general postulation of “Cool URIs”[12] and the proposed solution to the so called httpRange-14 issue[8].

Hillmann et al. [13] considered the problem of metadata quality in the context of metadata aggregation. While mainly focused on the practical problems of

[8] httpRange-14 (http://www.w3.org/2001/tag/issues.html#httpRange-14) was one (the 14th) of several issues that the Technical Architecture Group (TAG) of the W3C had to deal with: “What is the range of the HTTP dereference function?”Basically, the problem is that if a URI identifies a resource other than a webpage (non-information resource), then under this URI, no information about the resource can be provided, because in this case, the URI would also be the identifier for this information. The solution is to use HTTP redirects in this case, as described in this mail: http://lists.w3.org/Archives/Public/www-tag/2005Jun/0039.html

aggregation, the paper addresses the aspect of subsequent augmentation with subject headings and changes the emphasis from the record to the individual statement. Noting provenance and means of creation on this level of detail is considered necessary by the authors. They proposed an extension of OAI-PMH to implement their solution. [14] further expands on quality issues and note inconsistent use of metadata fields and the lack of bibliographic control among the major problems. Preserving provenance information at the repository, record or statement level is one of the proposed methods to ensure consistent metadata quality.

Currently, the W3C Provenance Incubator Group (Prov-XG, http://www.w3.org/2005/Incubator/prov/) addresses the general issue of provenance on the web. The requirements abstracted from various use-cases are summarized and further explained in by Zhao et al. [15]. The conclusion of this paper is basically ours: We need further standardization for the representation of provenance information for interoperable provenance-aware applications. They recommend that a possible next RDF standard should address the provenance issue.

Lopes et al. [16] emphasize the need for additional information as well, they refer to them as annotations and examine the need for annotations without consideration of the actual implementation - be it reification or named graphs. They come up with five types of annotations – time, spatial, provenance, fuzzy and trust – that can be seen as the most obvious use-cases for additional information.

A general model for the representation of provenance information as well as a review of provenance-related vocabularies is provided by The authors (Rephrase with cite) in [17]. The model aims to represent the whole process of data creation and access, as well as the publishing and obtaining of the associated provenance information.

With the Open Provenance Model (OPM, http://openprovenance.org/) exists a specification for a provenance model that meets the following requirements [18]: Exchange of provenance information, building of applications on top of OPM, definition of provenance independent from a technology, general applicability, multiple levels of descriptions. Additionally, a core set of rules is defined that allow to identify valid inferences that can be made on the provenance representation.

Finally, a comprehensive survey about publications on provenance on the web was created by The authors (Rephrase with cite) in [19], who also mentions approaches to modeling provenance in OWL ontologies.

The most powerful means to dealing with metametadata in OWL is the use of higher-order logics, which is supported, e.g., by OWL Full. However, as this type of metamodeling comes at the expense of decidability [20], weaker forms of metamodeling such as *punning*, a restricted way of using identical names for different types of entities (e.g. classes and individuals), have been proposed by the OWL community. In OWL 2, annotation properties can be used to make statements about entities, axioms and even annotations, but as annotation properties do not have a defined semantics in OWL, integrated reasoning over the various layers of metadata requires additional implementation effort [21]. Vrandecic et al. [22] discuss different metamodeling options by virtue of several use cases, including the representation of uncertainty in ontology learning [1], as well as ontology evaluation based on OntoClean (see also [23]). In addition to these application scenarios, weak forms of metamodeling in OWL are used, e.g., for including linguistic information in ontologies [24], but only few of these approaches are able to leverage the full power of logical inference over both metadata and metametadata [25].

IV. Conclusion

This paper is meant as a discussion paper. We have proposed five principles for the proper representation of metametadata which, in our opinion, have to be met by all source-agnostic, yet provenance-aware, linked data applications.

We have demonstrated that the technical requirements can already been met, and that the remaining problem is concerned with the establishments of conventions which define best-practice recommendations. In particular, these conventions should clarify how the metametadata is actually represented – so that an application can become aware of this metametadata, retrieve, maintain and republish it in a proper way. Currently, there is no accepted best-practice that follows our principles. We are involved in the Metadata Provenance Taskgroup of the Dublin Core Metadata Initiative[9] which aims to develop such best-practice recommendations in an as-open-as-possible way. This is why we are seeking for feedback, ideas and contributions to the ongoing discussions and the outcomes of this task group – because we want metadata provenance. Now!

Acknowledgements. Johanna Völker is financed by a Margarete-von-Wrangell scholarship of the European Social Fund (ESF) and the Ministry of Science, Research and the Arts Baden-Württemberg.

[9] http://dublincore.org/groups/provenance/)

REFERENCES

[1] P. Haase and J. Völker, "Ontology learning and reasoning – dealing with uncertainty and inconsistency," in *Uncertainty Reasoning for the Semantic Web I*, ser. Lecture Notes in Computer Science. Springer, 2008, vol. 5327, pp. 366–384.

[2] 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) 2005, May 10-14, 2005, Chiba, Japan*, 2005, pp. 613–622.

[3] S. Schenk and S. Staab, "Networked Graphs: A Declarative Mechanism for SPARQL Rules, SPARQL Views and RDF Data Integration on the Web," in *Proceedings of the 17th International Conference on World Wide Web (WWW) 2008, April 21-25, 2008, Beijing, China*, 2008.

[4] G. Flouris, I. Fundulaki, P. Pediaditis, Y. Theoharis, and V. Christophides, "Coloring rdf triples to capture provenance," in *ISWC '09: Proceedings of the 8th International Semantic Web Conference*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 196–212.

[5] F. Gandon and O. Corby, "Name That Graph - or the need to provide a model and syntax extension to specify the provenance of RDF graphs," in *Proceedings of the W3C Workshop - RDF Next Steps, June 26-27 2010, hosted by the National Center for Biomedical Ontology (NCBO), Stanford, Palo Alto, CA, USA*, 2010. [Online]. Available: http://www.w3.org/2009/12/rdf-ws/papers/ws06/

[6] K. Eckert, M. Pfeffer, and H. Stuckenschmidt, "A Unified Approach For Representing Metametadata," in *DC-2009 International Conference on Dublin Core and Metadata Applications*, 2009.

[7] R. Iannella and D. Campbell, "The A-Core: Metadata about Content Metadata," 1999, internet-Draft Document. [Online]. Available: http://metadata.net/admin/draft-iannella-admin-01.txt

[8] J. Hansen and L. Andresen, "Administrative Dublin Core (A-Core) Element," 2001.

[9] ——, "AC - Administrative Components: Dublin Core DCMI Administrative Metadata," 2003, final release of the Dublin Core Metadata Initiative Administrative Metadata Working Group. [Online]. Available: http://www.bs.dk/standards/AdministrativeComponents.htm

[10] Open Archives Initiative, "The Open Archives Initiative Protocol for Metadata Harvesting," 2008, protocol Version 2.0 of 2002-06-14, Edited by Carl Lagoze, Herbert Van de Sompel, Michael Nelson and Simeon Warner. [Online]. Available: http://www.openarchives.org/OAI/openarchivesprotocol.html

[11] ——, "Open Archives Initiative - Object Reuse and Exchange: ORE User Guide - Primer," Open Archives Initiative, 2008, edited by: Carl Lagoze, Herbert van de Sompel, Pete Johnston, Michael Nelson, Robert Sanderson and Simeon Warner. [Online]. Available: http://www.openarchives.org/ore/1.0/primer

[12] W3C SWEO Interest Group, "Cool URIs for the Semantic Web: W3C Interest Group Note 03 December 2008," 2008, edited by: Leo Sauermann and Richard Cyganiak. [Online]. Available: http://www.w3.org/TR/cooluris/

[13] D. I. Hillmann, N. Dushay, and J. Phipps, "Improving Metadata Quality: Augmentation and Recombination," in *Proceedings of the International Conference on Dublin Core and Metadata Applications*. Dublin Core Metadata Initiative, 2004. [Online]. Available: http://hdl.handle.net/1813/7897

[14] D. I. Hillmann, "Metadata Quality: From Evaluation to Augmentation," *Cataloging & Classification Quarterly*, vol. 46, no. 1, 2008. [Online]. Available: http://ecommons.library.cornell.edu/bitstream/1813/7899/1/Metadata_Quality_rev.pdf

[15] J. Zhao, C. Bizer, Y. Gil, P. Missier, and S. Sahoo, "Provenance Requirements for the Next Version of RDF," in *Proceedings of the W3C Workshop - RDF Next Steps, June 26-27 2010, hosted by the National Center for Biomedical Ontology (NCBO), Stanford, Palo Alto, CA, USA*, 2010.

[16] N. Lopes, A. Zimmermann, A. Hogan, G. Lukacsy, A. Polleres, U. Straccia, and S. Decker, "RDF Needs Annotations," in *Proceedings of the W3C Workshop - RDF Next Steps, June 26-27 2010, hosted by the National Center for Biomedical Ontology (NCBO), Stanford, Palo Alto, CA, USA*, 2010.

[17] O. Hartig, "Provenance Information in the Web of Data," in *Proceedings of the Workshop on Linked Data on the Web (LDOW) 2009, April 20, 2009, Madrid, Spain*. CEUR-WS, 2009, pp. 1–9. [Online]. Available: http://www.dbis.informatik.hu-berlin.de/fileadmin/research/papers/conferences/2009-ldow-hartig.pdf

[18] L. Moreau, B. Clifford, J. Freire, Y. Gil, P. Groth, J. Futrelle, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, Y. Simmhan, E. Stephan, and J. V. den Bussche, "The Open Provenance Model Core Specification (v1.1)," 2009. [Online]. Available: http://openprovenance.org/

[19] L. Moreau, "The Foundations for Provenance on the Web," *Foundations and Trends in Web Science*, November 2009. [Online]. Available: http://eprints.ecs.soton.ac.uk/18176/

[20] B. Motik, "On the properties of metamodeling in owl," in *International Semantic Web Conference*, ser. Lecture Notes in Computer Science, Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, Eds., vol. 3729. Springer, 2005, pp. 548–562.

[21] D. T. Tran, P. Haase, B. Motik, B. C. Grau, and I. Horrocks, "Metalevel information in ontology-based applications," in *Proceedings of the 23th AAAI Conference on Artificial Intelligence (AAAI)*, Chicago, USA, July 2008.

[22] D. Vrandecic, J. Völker, P. Haase, D. T. Tran, and P. Cimiano, "A metamodel for annotations of ontology elements in owl dl," in *Proceedings of the 2nd Workshop on Ontologies and Meta-Modeling*. Karlsruhe, Germany: GI Gesellschaft für Informatik, Oktober 2006.

[23] C. Welty, "Ontowlclean: Cleaning owl ontologies with owl," in *Proceeding of the 2006 conference on Formal Ontology in Information Systems*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2006, pp. 347–359.

[24] P. Buitelaar, P. Cimiano, P. Haase, and M. Sintek, "Towards linguistically grounded ontologies," in *ESWC*, ser. Lecture Notes in Computer Science, L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou, and E. P. B. Simperl, Eds., vol. 5554. Springer, 2009, pp. 111–125.

[25] B. Glimm, S. Rudolph, and J. Völker, "Integrated metamodeling and diagnosis in owl 2," in *Proceedings of the International Semantic Web Conference (ISWC)*, November 2006, to appear.

Provenance of Microarray Experiments for a Better Understanding of Experiment Results

Helena F. Deus
Department of Bioinformatics and Computational Biology
The University of Texas M. D. Anderson Cancer Center
Houston, USA
Instituto de Tecnologia Química e Biológica, UNL
Lisboa, Portugal

Jun Zhao
Deparment of Zoology
University of Oxford
Oxford, UK

Satya Sahoo
Kno.e.sis Center
Department of Computer Science and Engineering
Wright State University
Dayton, USA

Mathias Samwald
Digital Enterprise Research Institute
National University of Ireland Galway
Galway, Ireland

Eric Prud'hommeaux
World Wide Web Consortium
MIT
Cambridge, USA

Michael Miller
Tantric Designs
Seattle, USA

* M.Scott Marshall
Department of Medical Statistics and Bioinformatics
Leiden University Medical Center
Leiden, The Netherlands
Informatics Institute
University of Amsterdam
Amsterdam, The Netherlands

* Kei-Hoi Cheung
Center for Medical Informatics
Yale University School of Medicine
New Haven, USA

Abstract—**This paper describes a Semantic Web (SW) model for gene lists and the metadata required for their practical interpretation. Our provenance information captures the context of experiments as well as the processing and analysis parameters involved in deriving the gene lists from DNA microarray experiments. We demonstrate a range of practical neuroscience queries which draw on the proposed model. Our provenance representation includes the origins of the gene list and basic information about the data set itself (e.g. last modification date and original data source), in order to facilitate the federation of gene lists with other types of Semantic Web-formatted data and include the integration of a broader molecular context through additional omics data.**

Keywords-data integration, query federation, semantic web

I. INTRODUCTION

In the genomics/post-genomics era, massive amounts of data generated by high throughput experiments, including those using microarray technologies, have presented both promises and challenges to clinical, and translational research. One goal of microarray experiments is to discover, out of tens of thousands of genes, a small subset of genes (usually on the order of hundreds) whose expression pattern is indicative of some biological response to a given experimental condition.

Many computational/statistical approaches have been developed to detect such biologically significant gene lists. According to [1], the workflow of a microarray experiment is divided into the following steps: i) **experimental design** that includes the type of biological questions the experiment is designed to address, how the experiment is implemented (e.g., experiment and control), sample preparation, microarray platform selection, hybridization process, and scanning; ii) **data extraction**, which includes image quantification, filtering, and normalization; and iii) **data analysis and modeling**, which include approaches such as clustering, t-tests, enrichment analysis and so on.

The gene lists produced in step iii are usually reported as part of the experimental results published in scientific papers, and the steps involved in obtaining the gene lists are described in the methods section. Sometimes, gene lists are made electronically available (e.g., spreadsheets) through journal web sites. However, to the best of our knowledge, there is no standard format for uniformly representing and broadly sharing such gene lists in a focused scientific context.

We believe it would be useful to the community if such gene lists were commonly represented in a standard SW vocabulary and accessible to SW applications. This approach makes it possible for researchers to work with the gene list without requiring a post hoc significance analysis to re-derive the list. If experimental factors are included with gene lists, researchers can account for context without requiring labor-intensive manual research into the experimental factors for

*These authors contributed equally to this work. KC is supported in part by NIH grant U24 NS051869. JZ is supported by EPSRC grant EP/G049327/1. HFD is supported by the portuguese FCT (Fundação para a Ciência e Tecnologia) scholarship SFRH/BD/45963/2008. The work of MS was funded by the Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2) and by a postdoctoral fellowship from the Konrad Lorenz Institute for Evolution and Cognition Research, Austria.

each microarray study. A standard representation can be used both for gene lists reported in individual papers (note that these published gene lists are not yet stored in most microarray databases) and those computed from datasets collected from multiple microarray experiments across different microarray databases (e.g., GEO profiles [2] and Gene Expression Atlas [3]).

Integrated analysis (meta-analysis) requires raw and processed datasets from independent microarray experiments to be selected, compared, combined, and correlated using a variety of computational/statistical methods. This is, of course, much easier with machine-readable provenance and experimental context. To this end, MIAME [4] was proposed by the Microarray Gene Expression Data (MGED (http://www.mged.org)) community (now called "Functional Genomics Data Society" or FGED) to describe the *Minimum Information About a Microarray Experiment* (MIAME) that is needed to enable the interpretation of the results of the experiment unambiguously and potentially to reproduce the experiment. MIAME represents a set of guidelines for microarray databases and data management software. The MAGE data model and MAGE-ML (a standard XML format for serializing the MAGE model) [5] have been developed based on the MIAME data content specifications. In addition, MAGE-TAB [6] was proposed as a (more user-friendly) alternative to MAGE-ML.

Along with the development of these standards, a significant number of microarray databases ranging from individual labs (e.g., Nomad at deRisi lab (http://ucsf-nomad.sourceforge.net/)), institutions (e.g., SMD 7], YMD [8], and RAD [9]) to the scientific community (e.g., GEO [2] and ArrayExpress [10]) have been created, making large collections of microarray datasets accessible to the public. There are also microarray databases that serve the needs of specific biomedical domains (e.g., the NIH Neuroscience Microarray Consortium (http://np2.ctrl.ucla.edu/np2/home.do)). Major journal publishers have promoted sharing of microarray data by requiring authors to submit their data to public microarray repositories. Some journal publishers make supplemental data available on their web sites.

While many microarray databases are MIAME-compliant, several challenges still remain for researchers wishing to locate datasets relevant to their interest:

- There is no central repository for all microarray datasets, and experiment/dataset are stored on multiple databases.
- Users must learn to use different search interfaces and analytic facilities at each database.
- Many databases lack experimental context, annotation, and provenance.
- There is a lack of use of standard vocabularies in many microarray databases.
- The lists of differentially expressed genes discussed by most articles associated with a microarray study are not disclosed in any standard format, nor are they programmatically accessible.

The Semantic Web [11] has been actively explored in the context of biomedicine. For example, the W3C Semantic Web Health Care and Life Sciences Interest Group (HCLS IG) (http://www.w3.org/2001/sw/hcls/) represents a major community effort involving both academia and industry. The HCLS IG and allied efforts provide a growing corpus of biomedical datasets expressed in the Resource Description Framework (RDF) and web ontology language (OWL). Wang et al [12] has described how the transition from the eXtended Markup Language (XML) to RDF could potentially enhance semantic representation and integration of omic data. In addition to data, biomedical ontologies are made available to the community through organizations such as NCBO (http://www.bioontology.org/) and OBO Foundry (http://www.obofoundry.org/).

In this paper we explore using SW to represent microarray experimental data and provenance information about the context under which the data were generated, including the goal of the experiment, experimental factors (such as the disease or the cell region), and the statistical analysis process which leads to the experiment results. We explore the role of provenance information in helping biologists understand microarray experiments in the context of other experiments as well as other existing biomedical knowledge. To facilitate a quality-aware federation of microarray experiment results, we also provide provenance information about the gene lists data published using SW standards. As a pilot study, we take a bottom-up approach focusing on the type of provenance information required to meet our motivation use cases and creating a representation model with the minimum set of terms to meet these use cases. Although these terms are currently defined in our own namespaces, they can largely be mapped to existing provenance vocabularies, which are generically defined and evolving, to achieve maximum interoperability, in the next stage of our pilot study.

II. MOTIVATION

One motivation of microarray experiments is to identify genes that are differentially expressed in biological samples under different conditions (e.g., disease vs. control). The samples may come from tissues extracted from different organs or parts of the same organ (e.g., different brain regions). In this case, we may be able to discover differentially expressed genes in each organ/organ part and how disease may affect each organ/organ part at the gene expression level. A common outcome of experiments is a list of candidate genes which may serve as diagnostic or therapeutic markers. These gene lists, abundant in biomedical literature, are provided in heterogeneous formats (e.g., Excel spreadsheets and printed tables embedded in papers) that hinder the reuse of the results. In order to reuse such gene lists in additional pathway or molecular analysis, it is important that they are represented in a standardized, distributable, and machine-readable format that is amenable to semantic queries.

After obtaining a representative list of differentially expressed genes, scientists may need to study these experiment results in a broader molecular context with additional data. In the case of neurological disease studies such as Alzheimer's Disease (AD), researchers may want to combine gene expression data from multiple AD microarray studies. For example, one characterization of AD is the formation of intracellular neurofibrillary tangles that affect neurons in brain regions involved in the memory function. It is important to have meta-data such as the cell type(s), cell histopathology, and brain region(s) for comparing/integrating the results across different AD microarray experiments. It is important also to consider the (raw) data source and the types of analysis performed on the data to arrive at meaningful interpretations. Finally, gene expression data may be combined with other types of data including genomic functions, pathways, and associated diseases to broaden the spectrum of integrative data analysis.

In our pilot study, we selected three microarray experiments from different journals ([13-15]) to explore how to represent gene list experiment results in a structured format and what types of metadata can better enable the computer to search for genes that may play a molecular role in the pathogenesis of AD. All the gene lists from the selected publications were derived from human brain samples that were prepared for AD studies. We wanted to be able to answer a variety of user questions regarding semantically related experiments and their experimental results. For example:

- Q0: What microarray experiments analyze samples taken from the Entorhinal cortex region of Alzheimer's patients?
- Q1: Was the same data normalization algorithm or statistical software package used in both studies that analyze gene expression in the entorhinal cortex region of AD patients?
- Q2: What genes are overexpressed in the Entorhinal cortex region in the context of Alzheimer's and what is their expression fold change and associated p-value?
- Q3: Are there any genes that are expressed differently in two different brain regions (such as in Hippocampus and Entorhinal cortex)?

The MIAME standard outlines the minimum set of information that is needed for describing microarray experiments in order to facilitate the reproduction of these experiments and a uniform interpretation of experiment results. Experiments recording and publishing MIAME-compliant experimental protocol should contain sufficient information to answer questions like Q0 and Q1. However, because MIAME does not specify a format, and MAGE-ML and MAGE-TAB do not specify a standard representation for experiment results (such as the set of genes showing particular expression patterns), there is no simple mechanism to find semantically related experimental results based on the patterns of differentially expressed genes.

In order to answer questions Q2 and Q3, it is necessary to model both experimental information (ex: Entorhinal cortex) and statistical data (e.g. the p-values associated with gene expression values).

Additionally, we want to be able to extend the knowledge about genes linked to AD such that scientists can access and extend their understandings about their gene expression data analysis results to answer questions like the following:

- Q4: What other diseases may be associated with the same genes found to be linked to AD?
- Q5: What drugs are known that affect the same overexpressed gene products and what are their target diseases?
- Q6: Select all the genes determined to be differentially expressed in the Entorhinal cortex in experiments performed by AD investigators at the Translational Genomics Research Institute

For these types of questions, the microarray experiment results need to be federated (Q4, Q5) or combined (Q6) with other datasets describing the data itself. We show how the structured representation of microarray experiment data and associated provenance metadata will enable us to query across different aspects of domain knowledge about these experiment results using several other datasets in the HCLS KB. We also show how we can provide additional provenance information about different datasets to support some quality-aware federation queries over distributed data sources.

III. Methods

To address questions Q0-Q3 we need both a precise representation of the gene lists reported in the three selected publications and a representation of the provenance of these gene lists, such as the methods and procedures involved in their generation. As mentioned in Section I, several standards exist for describing microarray experiment protocols, however, none is comprehensive enough to fully capture the complex process of reporting the results of a microarray experiment. To answer questions Q4-Q5 we need to query across the exemplar datasets, using provenance information of different levels of granularity, from the basic information about the context of each experiment to details about the analysis processes generating the gene expression results. Although a number of provenance vocabularies, such as the open provenance model (OPM, http://openprovenance.org/) and Provenir (http://wiki.knoesis.org/index.php/Provenir_Ontology) are available, we choose a bottom-up approach in this pilot study. On the one hand, at the time of the writing, little was known about how to choose between these existing vocabularies to best suit our purpose; on the other hand, our pilot study aims to focus on capturing the minimum information to answer our case study questions. This approach has the added advantage of shielding our model from having to keep pace with rapidly evolving ontologies while still enabling mapping to upper level ontologies in the future. For these reasons, our data model includes the minimum set of terms necessary to describe the three examples selected, and is made available under our own local namespace:

```
@prefix biordf:http://purl.org/net/biordfmicroarray/ns#
```

Compared with provenance vocabularies, many domain specific ontologies are much more established and stable, such as NIF (http://www.neuinfo.org/), disease ontology (DO, http://do-wiki.nubic.northwestern.edu/index.php/Main_Page), or the voiD vocabulary [16]. Therefore, we reuse terms from these ontologies that are already widely used to annotate (biological) datasets in our data model in order to enable maximum interoperability with other approaches.

A. *The Data Model*

Our data model captures the minimum information necessary to describe the gene lists and the microarray experiment context in which they were generated. To answer each of the individual case study questions, different aspects of each dataset had to be considered. For example, to answer questions like **Q0** and **Q3** a good overview of each microarray experiment is necessary, including the samples used, the disease of interest, microarray platform, etc. For questions like **Q1** and **Q2**, however, a different set of assertions concerned specifically with comparing gene expression quantification methods in different settings is required. Finally, the ability to answer questions like **Q4** and **Q5** involve the more complex component of performing simultaneous queries on more than one data source. As such, information describing the metadata associated with each data source is also necessary. To accommodate these different data types in our model, we have defined four provenance levels, with each level entailing different subsets of information:

Institutional level: Includes assertions about the laboratory where the experiments were performed and the reference where the results were published to help determine the trustworthiness of the data. This information is useful to constrain the list of significant genes to only those that are published in peer-reviewed articles and/or were performed at certain institutions that have the track record of generating high quality microarray data published in respected journals.

Experiment protocol level: Includes assertions about the brain regions from which the samples were gathered and the histology of the cells. Such information has been partially mapped to MGED, DO and NIF terms.

Data analysis and significance level: Includes assertions about the statistical analysis methodology for selecting the relevant genes. Terms defined for this level are also provided as a separate statistic module (http://purl.org/net/biordfmicroarray/stat#) to describe software tools and statistical terms.

Dataset description level: Includes assertions about when the dataset is published, based on which version of a source dataset, and who published the dataset. Some existing vocabularies for describing RDF datasets on the Web were reused to enhance their trustworthiness such as the Vocabulary of Interlinked Dataset (voiD) [16] that provide basic information about who published the data as well as a summary of the content of the dataset, such as the number of genes described by the dataset or the SPARQL endpoint through which the dataset can be accessed. The Provenance Vocabulary [17] was also used to provide a richer set of provenance information, such as when the dataset is published, using which tool, or by accessing which data server.

B. *Formulation of SPARQL queries*

The queries described here are formulated at our demo site (http://purl.org/net/biordfmicroarray/demo), where they can be directly executed or copied and performed locally using software such as SWObjects (https://sourceforge.net/projects/swobjects/files/). The demo site also includes a diagram explaining the four provenance levels and the types of data entailed in each level.

To answer **Q0**, experiments performed in samples collected from patients with Alzheimer's disease in a specific area of the brain, the Entorhinal cortex, must be selected from the RDF representation. The data necessary to answer to this question is completely entailed in the experimental provenance level and can be formulated in terms of the entities used to represent each step of the workflow involved in collecting a Sample. Making use of data from the statistical analysis provenance level, the same query **Q0** can be amended to filter the list of experiments retrieved based on the statistical normalization software thus enabling an answer to **Q1**. To answer questions **Q2** and **Q3** data pertaining to the experiment provenance level must also be combined with information about the gene lists, such as the expression level for each gene. A common requirement to measure statistical significance of differentially expressed genes is the p-value that is associated with gene expression fold change. In **Q2**, this information is used to trim the list of over-expressed genes by indicating that fold change > 0 but only in cases where the p-value is < 0.001.

One of the most significant advantages of representing gene lists in RDF is helping scientists enrich it with data from linked datasets such that questions like **Q4** and **Q5** may be answered. The dataset description provenance level enables the discovery of useful datasets for specific purposes, such as, e.g. using the HCLS Kb to discover diseases that may be associated with specific genes. **Q4**, detailed below, achieves that goal by first retrieving the same list of genes as in **Q2** and, secondly, by selecting the most recently updated SPARQL service which includes assertions about both genes and diseases. The final section queries this service to retrieve the correlated diseases.

```
    SELECT DISTINCT  ?diseaseName ?geneLabel ?geneName WHERE {
    #Retrieve a list of overexpressed genes in the entorhinal cortex of AD
patients
    {
      ?experimentSet dct:isPartOf ?microarray_experiment ;
                     biordf:has_input_value ?sampleList ;
                     biordf:differentially_expressed_gene ?gene ;
                     biordf:has_ouput_value ?foldChange .
      ?sampleList  biordf:derives_from_region ?brainRegion ;
                   biordf:patients_have_disease ?alzheimers .
      ?gene  rdfs:label ?geneLabel ;
             biordf:name   ?geneName .
      ?foldChange rdf:value ?foldChangeValue ;
                  stat:p_value ?pval .
      #Apply filters to constrain the amount of results
        FILTER (xsd:float(?foldChangeValue) > 0)
        FILTER (xsd:float(?pval) < 0.001 )
        FILTER (?brainRegion = neurolex:Entorhinal_cortex )
        FILTER (?alzheimers = doid:DOID_10652 )
    }
    #Find most recently updated SPARQL endpoint that contains information
about genes and diseases.
    {
        ?source rdf:type void:Dataset ;
        void:sparqlEndpoint ?srvc ;
        dct:issued ?issued  ;
        dct:subject diseasome:diseases ;
        dct:subject diseasome:genes .
    OPTIONAL {
        ?source1 rdf:type void:Dataset ;
        void:sparqlEndpoint ?srvc2 ;
        dct:issued ?issued2 ;
        dct:subject diseasome:diseases ;
        dct:subject diseasome:genes .
        FILTER (?issued2 > ?issued)
    }
    FILTER (!BOUND(?srvc2))
    }
    #Get associated diseases from most recently updated Diseasome server.
      SERVICE ?srvc2 {
        ?diseasomeGene rdfs:label ?geneLabel .
        ?disease diseasome:associatedGene ?diseasomeGene.
        ?disease rdfs:label ?diseaseName .
    }
    }
```

Finally, to answer **Q6** data from the institutional provenance level we must limit the list of retrieved experiments to those that were performed at a specific institution. The queries presented here are executable through our demo at http://purl.org/net/biordfmicroarray/demo. Their time to execution ranges between 100 and 200 ms for local queries (Q1-Q3, Q6) and a few seconds (2-5s) for federated queries (Q4-Q5) executed using SWObjects.

C. *Availability*

The RDF representation was generated using JavaScript and the data was loaded into a public SPARQL endpoint (http://purl.org/net/biordfmicroarray/sparql). We elaborate and further expand the provenance queries in this paper at our demo site http://purl.org/net/biordfmicroarray/demo. A figure associating each of the four provenance levels with the data that they are concerned with is also made available at the demo site. The complete RDF/turtle representation can be downloaded from http://biordfmicroarray.googlecode.com/files/all3_genelists_provenance.ttl. The JavaScript code to convert Excel spreadsheets into RDF is available at http://code.google.com/p/biordfmicroarray/ .

IV. Discussion

A data model to explicitly make the content and context of gene lists (e.g., differentially expressed genes) available in RDF format was developed. In the process, four types of provenance were identified that were found necessary to characterize, discover, reproduce, compare and integrate gene lists with other data. Expressing provenance in RDF enables describing the data itself (i.e. its origin, version and URL location) in the same language as the elements represented therein. The power of this uniform access to data and metadata should not be underestimated. In practice, this means that SPARQL queries can express constraints both about the origins of the data and contents (or attributes) of the data as demonstrated by query Q4. In the case of Linked Open Data, the set of best practices for exposing data as RDF through a SPARQL endpoint, researchers often need to distinguish between multiple RDF renderings (i.e. representations) of the same data set or different versions of it. Different endpoints can be discovered by issuing queries that target the data sources themselves: When was the last RDF rendering created and by whom (or which project)? Which ontologies/vocabularies were used? The same standardized SW mechanisms of reasoning and pattern matching can be applied to select a specific data source as the ones used to discover related facts across the data sources.

The provenance data model developed for reporting microarray experiment results while capturing different types of provenance information was motivated by our user-defined queries. We have therefore applied a bottom-up approach that focused on describing the data first before mapping it to widely used ontologies. Although several provenance ontologies are available, some of them are upper level ontologies, such as Provenir, therefore lacking the specific terms required for describing how gene lists were derived. Other ontologies, such as the Provenance Vocabulary for Linked Data and proof markup language, were created for specific application domains, such as explaining reasoning results. Our bottom-up approach enabled us to identify and define the minimum set of provenance terms to answer a set of queries from different perspectives and shield the data model from depending on external vocabularies which are often subject to changes. For increased interoperability, mapping terms from our model to terms from a community provenance model, such as the OPM or others is straightforward. For example, our property *biordf:has_input_value* can be made a sub-property of the inverse of OPM property *used*, and *biordf:derives_from_region* can become a sub-property of OPM property *wasDerivedFrom*.

Further down the pipeline of microarray studies, bioinformaticians will often need to combine knowledge about the genes derived from their microarray experiments in order to achieve a deeper understanding at a systems biology level. Although the number of genes that has to be taken into consideration while studying Alzheimer's has been significantly reduced by many gene expression studies, a good number of genes (ranging from tens to hundreds) are yet to be processed. One approach becoming increasingly popular is the use of scientific workflow workbenches (such as Taverna and Kepler) to perform large scale data analysis. Many such workbenches [19-20] also record the workflow provenance information about, for example, what genes from which organism were processed and how the proteins encoded by the genes were discovered by querying various genomic databases. Combining this workflow provenance information and the set of microarray experiment-related provenance information by mapping both to a common community provenance model, such as OPM, the trustworthiness and reproducibility of experiment results would be increased throughout the whole experiment life cycle. McCusker et al. [21] has taken a first step towards by providing a tentative translation from MGED-TAB to the OPM.

While we endorse the use of SW technologies as the standard machine-readable format, we acknowledge that most biologists are not familiar with SW and prefer to use formats such as Excel spreadsheets to work with gene list results. To this end, it would be useful to use a standardized user-friendly format (e.g., MAGE-TAB) for encoding gene lists and their context that could be easily converted into the SW format.

V. Conclusion

We describe and illustrate with a case study the beneficial role of Semantic Web technologies in 'omic' data representation by providing and querying a data model to capture provenance information related to reporting microarray experiment results. We have tackled not only the engineering aspect of the data integration problem, but also the more fundamental issues of federating data that begin with seemingly homogeneous data sources (microarray databases) and extends to heterogeneous data domains at multiple levels. This is also driven by the growing collaboration between a wide spectrum of scientific disciplines and communities such as is required for translational research. We have used a bottom-up approach that facilitated the identification of four provenance levels necessary to report microarray experiment results and shielded our data model from becoming dependent on constantly evolving ontologies. We have, however, discussed how some of the terms and relationships from existing provenance ontologies can be mapped to our model. Some issues found to be necessary in the integration of microarray data sources could also be considered relevant for the federation of data sources in general. As more 'omics' data are generated, the complexity and requirements for discovery-based research increases. As a result, there is a growing demand for effective data provenance and integration at many levels that counts on the active involvement of scientists and informaticians. Our work represents a step in this direction.

Acknowledgment

The authors would like to express their sincere gratitude to the HCLS IG for helping to organize and coordinate with different task forces including the BioRDF task force within which the neuroscience microarray use case was explored. Also thanks to Helen Parkinson, James Malone, Misha Kapushesky, Jonas Almeida and three anonymous reviewers. MSM appreciated the support of Jelle Goeman (LUMC) during this work.

References

[1] Stears RL, Martinsky T, Schena M. (2003). Trends in microarray analysis. Nature Medicine. (9): 140 – 145.

[2] Barrett T, Troup DB, et al.. (2009). NCBI GEO: archive for high-throughput functional genomic data. Nucleic Acids Res. 2009 Jan;37(Database issue):D885-90.

[3] Lukk M, Kapushesky M, et al.. A global map of human gene expression. *Nat Biotechnology* **28**, 322-324 (2010)

[4] Brazma A, Hingamp P, et al.. (2001). Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. Nat Genet. 29(4):365-71.

[5] Spellman PT, Miller M, et al.. (2002). Design and implementation of microarray gene expression markup language (MAGE-ML). Genome Biol. 3(9):RESEARCH0046.

[6] Rayner TF, Rocca-Serra P, et al.. (2006). A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB. BMC Bioinformatics. 7:489.

[7] Gollub J, Ball CA, et al.. (2003). The Stanford Microarray Database: data access and quality assessment tools. Nucleic Acids Res. 31(1):94-6.

[8] Cheung KH, White K, et al.. (2002). YMD: a microarray database for large-scale gene expression analysis. Proc AMIA Symp. 2002:140-4.

[9] Manduchi E, Grant GR, et al.. (2004). RAD and the RAD Study-Annotator: an approach to collection, organization and exchange of all relevant information for high-throughput gene expression studies. Bioinformatics. 20(4):452-9.

[10] Parkinson H, Sarkans U, et al.. (2005). ArrayExpress--a public repository for microarray gene expression data at the EBI. Nucleic Acids Res. 33(Database issue):D553-5.

[11] Berners-Lee T, Hendler J, Lassila O. (2001). The Semantic Web. Scientific American. 284(5):34-43

[12] Wang X, Gorlitsky R, Almeida JS. (2005) From XML to RDF: how semantic web technologies will change the design of 'omic' standards. Nat Biotechnol. 23(9):1099-103.

[13] Dunckley T, Beach TG, et al.. (2006). Gene expression correlates of neurofibrillary tangles in Alzheimer's disease. Neurobiol Aging;27: 1359-71.

[14] Liang WS, Dunckley T, et al.. (2007). Gene expression profiles in anatomically and functionally distinct regions of the normal aged human brain. Physiol Genomics 28: 311-22.

[15] Liang WS, Reiman EM, et al.. (2008). Alzheimer's disease is associated with reduced expression of energy metabolism genes in posterior cingulate neurons. Proc Natl Acad Sci U S A l2008;105: 4441-6.

[16] Alexander K, Cyganiak R, Hausenblas M, and Zhao J. Describing linked datasets. In *Linked Data on the Web Workshop in the International World Wide Web Conference*, Madrid, Spain, 2009 .

[17] Hartig O, Zhao J. Publishing and consuming provenance metadata on the web of linked data. In *Proceedings of The third International Provenance and Annotation Workshop*, Troy, NY, U.S.A, 2010. In press

[18] Kotecha N, Bruck K, Lu W, Shah N. Pathway knowledge base: An integrated pathway resource using BioPAX. Applied Ontology. 3(4); 235-245. 2008

[19] Missier P, Sahoo S, Zhao J, Goble C and Sheth A. Janus: Semantic Provenance Infrastructure for Taverna. In *Proceedings of The third International Provenance and Annotation Workshop*, Troy, NY, U.S.A, 2010. In press

[20] Altintas I, Anand M, et al.. Understanding Collaborative Studies Through Interoperable Workflow Provenance. In *Proceedings of The third International Provenance and Annotation Workshop*, Troy, NY, U.S.A, 2010. In press

[21] McCusker J. and McGuinness D. Explorations into the Provenance of High Throughput Biomedical Experiments. In *Proceedings of The third International Provenance and Annotation Workshop*, Troy, NY, U.S.A, 2010. In press

Semantic Representation of Provenance in Wikipedia

Fabrizio Orlandi
Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
fabrizio.orlandi@deri.org

Pierre-Antoine Champin
LIRIS, Université de Lyon, CNRS, UMR5205
Université Claude Bernard Lyon 1, F-69622
Villeurbanne, France
pchampin@liris.cnrs.fr

Alexandre Passant
Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
alexandre.passant@deri.org

***Abstract*—Wikis are often considered as being a wide source of information. However, identifying provenance information about their content is crucial, whether it is for computing trust in public wiki pages or to identify experts in corporate wikis. In this paper, we address this issue by providing a lightweight ontology for provenance management in wikis, based on the W7 model. Furthermore, we showcase the use of our model in a framework that computes provenance information in Wikipedia, also using DBpedia to compute provenance and contribution information per category, and not only per page.**

I. Introduction

From public encyclopedia to corporate knowledge management tools, wikis are often considered as being a wide source of information. Yet, since wikis generally offer an open publishing process where everyone can contribute, identifying provenance information in their pages is an important requirement. In particular this information can be used to identify trust values for pages or pages fragments [2] as well as for identifying experts based on the number of contributions [9] and other criteria such as the users' social graphs [10] etc. By providing this information as RDF [6], provenance metadata becomes more transparent and offers new opportunities for the previous use-cases, as well as letting people link to provenance information from other sources, and personalizing trust metrics based on the trust they have to a person regarding a particular topic [5].

This paper describes three of our contributions to address this issue and make provenance information in MediaWiki-powered wikis [1] available on the Semantic Web:

1) a lightweight ontology to represent provenance information in wikis, based on the W7 theory [13] and using SIOC and its extensions;
2) a software architecture to extract and model provenance information about Wikipedia pages and categories, using the aforementioned ontology;
3) a user-interface to make this information openly available on the Web, both to human and software agents and directly within Wikipedia pages.

In the next section, we discuss some related work in the realm of provenance management on the Semantic Web. Then, we give some background information regarding SIOC and various extensions used in our work. In Section IV, we present the W7 theory and the lightweight ontology we have built to represent it in RDFS. We then describe our software architecture and how we compute provenance information in Wikipedia and finally present the user-interface to access this information, before concluding the paper.

II. Related Work

The representation and extraction of provenance information is not a recent research topic. Many studies have been conducted for representing provenance of data [15], but few of them have been focused on integrating provenance information into the Web of data [6]. Providing this information as RDF would make provenance meta-data more transparent and interlinked with other sources, and it would also offer new scenarios on evaluating trust and data quality on the top of it. In this regard a W3C Provenance Incubator Group[2] has been recently established. The mission of the group is to "provide a state-of-the art understanding and develop a roadmap in the area of provenance for Semantic Web technologies, development, and possible standardization". Requirements for provenance on the Web[3], as well as several use cases and technical requirements have been provided by the working group. A comprehensive analysis of approaches and methodologies for publishing and consuming provenance metadata on the Web is exposed in [7].

Another research topic relevant to our work is the evaluation of trust and data quality in wikis. Recent studies proposed several different algorithms for wikis that would automatically calculate users' contributions and evaluate their quantity and quality in order to study the authors' behavior, produce trust measures of the articles and find experts. WikiTrust [2] is a project aimed at measuring the quality of author contributions on Wikipedia. They developed a tool that computes the origin and author of every word on a wiki page, as well as "a measure of text trust that indicates the extent with which text has been revised"[4]. On the same topic other researchers tried

This work is funded by the Science Foundation Ireland under grant number SFI/08/CE/I1380 (Líon 2) and by an IRCSET scholarship.

[1] MediaWiki is the wiki engine that powers Wikipedia – www.mediawiki.org

[2] established in September 2009. http://www.w3.org/2005/Incubator/prov/

[3] http://www.w3.org/2005/Incubator/prov/wiki/User_Requirements

[4] WikiTrust: http://wikitrust.soe.ucsc.edu/

to solve the problem of evaluating articles' quality, not only examining quantitatively the users' history [9], but also using social network analysis techniques [10].

From our perspective, there is a need of publishing provenance information as Linked Data from websites hosting a wide source of information (such as Wikipedia). Yet, most of the work on provenance of data is, either not focused on integrating the information generated on the Web of data, or mainly based on provenance for resource descriptions or already structured data. On the other hand, the interesting work done so far on analyzing trust and quality on wikis does not take into account the importance of making the information extracted available on the Web of data.

III. Background

A. *Using SIOC for wiki modelling*

The SIOC Ontology — Semantically-Interlinked Online Communities [1] — provides a model for representing online communities and their contributions[5]. It is mainly centered around the concepts of *users*, *items* and *containers*, so it can be used to model content created by a particular user on several platforms, enabling a distributed perspective to the management of User-Generated Content on the Web. In particular, the atomic elements of the Web applications described by SIOC are called `Items`. They are grouped in `Containers`, that can themselves be contained in other `Containers`. Finally, every `Container` belongs to a `Space`. As an example, a `Site` (subclass of `Space`) may contain a number of `Wikis` (subclass of `Container`) and every `Wiki` contains a set of `WikiArticles` (subclass of `Item`) generated by `UserAccounts`. For more details about SIOC, we invite the reader to consult the W3C Member Submission [1] and its online specification[6].

While the SIOC Types module provides several subclasses of `Container` and `Item`, including `Wiki` and `WikiArticle`, some characteristics of wikis required further modelling. Hence, in our previous work [11] we extended the SIOC Ontology to take into account such characteristics (*e.g.* multi-authoring, versioning, etc.). Then, some tools to generate and consume data from wikis using our model have also been developed [12].

B. *The SIOC Actions module*

While SIOC represents the state of a community at a given time, SIOC-actions [4] can be used to represent their dynamics, *i.e.* how they evolve. Hence, SIOC provides a *document-centric* view of online communities and SIOC-actions focuses on an *action-centric* view. More precisely, the evolution of an online community is represented as a set of *actions*, performed by a user (`sioc:UserAccount`), at some time, and impacting a number of objects (`sioc:Item`). SIOC-actions provides an extensible hierarchy of properties for representing the effect of an *action* on its *items*, such as `creates`, `modifies`, `uses`, etc. Besides the SIOC ontology, SIOC-actions relies on the vocabulary for Linking Open Descriptions of Events (LODE)[7]. The core of the module is the `Action` class (subclass of `event:Event` from the Event Ontology) which is a timestamped event involving an agent (*e.g.* a `UserAccount`) and a number of digital artifacts (*e.g.* `Items`). For more details about SIOC Actions and its implementation see the following Sec. IV.

IV. Representing the W7 model using RDFS/OWL

The W7 model is an ontological model created to describe the semantics of data provenance [13]. It is a conceptual model and to the best of our knowledge a RDFS/OWL representation of this model has not been implemented yet. Hence we will focus on an implementation of this model for the specific context of wikis. As a comparison, in [14] the authors use the example of Wikipedia to illustrate theoretically how their proposed W7 model can capture domain or application specific provenance.

The W7 model is based on the Bunge's Ontology [3], furthermore it is built on the concept of tracking the history of the events affecting the status of things during their life cycle. In this particular case we consider the data life cycle. The Bunge's ontology, developed in 1977, is considered as one of the main sources of constructs to model real systems and information systems. Since the Bunge's work is a theoretical work, there has been some effort from the scientific community to translate his work into machine readable ontologies[8].

The W7 model represents data provenance using seven fundamental elements or interrogative words: *what*, *when*, *where*, *how*, *who*, *which*, and *why*. It has been purposely built with general and extensible principles, hence it is possible to capture provenance semantics for data in different domains. We refer to [13] for a detailed description of the mappings between W7 and Bunge's models, and in Table I we provide a summary of the W7 elements (as in [14]).
Looking at the structure of the W7 model it is clear the motivation why we chose the SIOC Actions module as core of our model. Most of the concepts in the Actions module are the same as in the W7 model. Furthermore wikis are community sites and the Actions module has been implemented to represent dynamic, action-centric views of online communities.

In the following sections we give a detailed description of how we answered each of these seven questions.

A. *What*

The *What* element represents an event that affected data during its life cycle. It is a change of state and the core of the model. In this regard, there are three main events affecting data: *creation*, *modification* and *deletion*. In the context of wikis, each of them can appear: users can (1) add new sentences (or characters), (2) remove sequences of characters, or (3) modify characters by removing and then adding content

[5] http://sioc-project.org

[6] http://rdfs.org/sioc/spec/

[7] LODE Ontology specification — http://linkedevents.org/ontology/

[8] Evermann J. provides an OWL description of the Bunge's ontology at: http://homepages.mcs.vuw.ac.nz/~jevermann/Bunge/v5/index.html

Provenance element	Construct in Bunge's ontology	Definition
What	Event	An event (i.e. change of state) that happens to data during its life time
How	Action	An action leading to the events. An event may occur, when it is acted upon by another thing, which is often a human or a software agent
When	Time	Time or more accurately the duration of an event
Where	Space	Locations associated with an event
Who	Agent	Agents including persons or organizations involved in an event
Which	Agent	Instruments or software programs used in the event
Why	-	Reasons that explain why an event occurred

TABLE I
DEFINITION OF THE 7 WS BY RAM S. AND LIU J.

in the same position of the article. In addition, in systems like Wikipedia, some other specific events can affect the data on the wiki, for example "quality assessment" or "change in access rights" of an article [14]; however, they can be expressed with the three broader types defined above.

Since (1) wikis commonly provide a versioning mechanism for their content and (2) every action on a wiki article leads to the generation of a new article revision, the core event describing our *What* element is the creation of an article version. In particular we model this creation, and the related modification of the latest version (*i.e.* the permalink), using the SIOC-Actions model as shown in Listing 1.

```
<http://example.com/action?title=Dublin_Core#380106133>
    sioca:creates <http://en.wikipedia.org/w/index.php?
        title=Dublin_Core&oldid=380106133>;
    sioca:modifies <http://en.wikipedia.org/wiki/
        Dublin_Core>;
    a sioca:Action.
```

Listing 1. Representing the "What" element

As we can see from the example above expressed in Turtle syntax, we have a `sioca:Action` identified by the URI ⟨http://example.com/action?title=Dublin_Core#380106133⟩ that leads to the creation of a revision of the main wiki article about "Dublin Core". The creation of a new revision was originated by a modification (`sioca:modifies`) of the main Wikipedia article ⟨http://en.wikipedia.org/wiki/Dublin_Core⟩. Details about the type of event are exposed in the next section about the *How* element, where we identify the type of action involved in the event creation.

B. *How*

The *How* element in W7 is an equivalent to the *Action* element from Bunge's ontology, and describes the action leading to an event. In wikis, the possible actions leading to an event (*i.e.* the creation of a new revision) are all the edits applied to a specific article revision. By analyzing the *diff* between two subsequent revisions of a page, we can identify the type of action involved in the creation of the newer revision. In particular we focus on modelling the following types of edits: *Insertion*, *Update* and *Deletion* of both *Sentence*s and *Reference*s. With the term *Sentence* here we refer to every sequence of characters that does not include a reference or a link to another source, and with *Reference* we refer to every action that involves a link or a so-called Wikipedia *reference*. As discussed in [14], another type of edit would be a *Revert*, or an undo of the effects of one or more edits previously happening. However, in Wikipedia, a revert does not restore a previous version of the article, but creates a new version with content similar to the one from an earlier selected version. In this regard, we decided to model a revert as all the other edits, and not as a particular pattern. The distinction between a revert and other types of action can be yet identified, with an acceptable level of precision, by looking at the user comment entered when doing the revert, since most users add a related revert comment [9].

Going further, and to represent provenance data for the action involved in each wiki edit, we modelled the *diffs* appearing between pages. To model the differences calculated between subsequent revisions we created a lightweight Diff ontology, inspired by the Changeset vocabulary[10]. Yet, instead of describing changes to RDF statements, our model aims at describing changes to plain text documents. It provides a main class, the `diff:Diff` class, and six subclasses: `SentenceUpdate`, `SentenceInsertion`, `SentenceDeletion` and `ReferenceUpdate`, `ReferenceInsertion`, `ReferenceDeletion`, based on the previous *How* patterns.

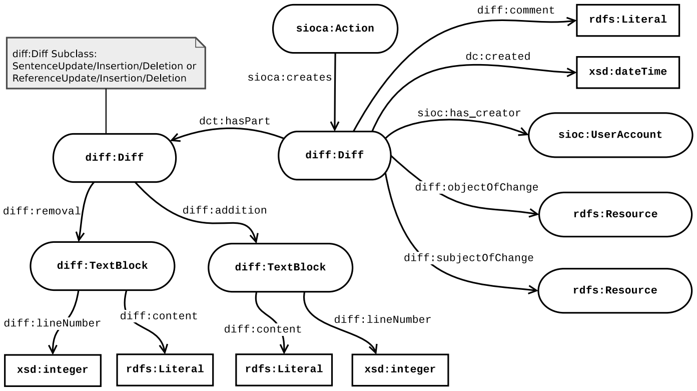


Fig. 1. Modeling differences in plain text documents with the *Diff* vocabulary

The main `Diff` class represents all information about the change between two versions of a wiki page (see Fig. 1). The `Diff`'s properties `subjectOfChange` and `objectOfChange` point respectively to the version changed by this *diff* and to the newly created version. Details about the time and the creator of the change are provided respectively by `dc:created` and `sioc:has_creator`. Moreover, the comment about the change is provided by the `diff:comment` property with range `rdfs:Literal`. In

[9] Note that we could also compare the n-1 and n+1 version of each page to identify if a change is a revert

[10] The Changeset schema: http://purl.org/vocab/changeset/schema#

Figure 1 we also display a Diff class linking to another Diff class. The latter represents one of the six Diff subclasses described earlier in this section. Since a single *diff* between two versions can be composed by several atomic changes (or *"sub-diffs"*), a Diff class can then point to several subclasses using the dc:hasPart property. Each Diff subclass can have maximum one TextBlock removed and one added: if it has both, then the type of change is an *Update*, otherwise the type would be an *Insertion* or a *Deletion*.

The TextBlock class is part of the Diff ontology and represents a sequence of characters added or removed in a specific position of a plain text document. It exposes the content itself of this sequence of characters (content) and a pointer to its position inside the document (lineNumber). It is important to precise that usually the document content is organized in sets of lines, as in wiki articles, but this class is generic enough to be reusable with other types of text organization. To note also that each of the six subclasses of the Diff class inherit the properties defined for the parent class, but unfortunately this is not displayed in Figure 1 for space reasons.

With the model presented it is possible to address an important requirement for provenance: the reproducibility of a process. Starting from an older revision of a wiki article, just following the *diffs* between the newer revisions and the TextBlocks added or removed, it is possible to reconstruct the latest version of the article. This approach goes a step further than just storing the different data versions: it provides details of the entire process involved in the data life cycle.

C. When

The *When* element in W7 is equivalent to the *Time* element from Bunge's ontology, and obviously refers to the time an event occurs, which is recorded in every wiki platform for page edits. As depicted in Figure 1, each Diff class is linked to the timestamp of the event using the dc:created property. The same timestamp is also linked to each Diff subclass using the same property (not shown in Fig. 1 for space reasons). The time of the event is modelled with more detail in the Action element as shown in the following Listing 2 [11].

```
<http://example.com/action?title=Dublin_Core#380106133>
   dc:created "2010-08-21T06:36:17Z"^^<http://www.w3.org
       /2001/XMLSchema#dateTime>;
   lode:atTime [
     a time:Instant;
     time:inXSDDateTime "2010-08-21T06:36:17Z"^^<http://
          www.w3.org/2001/XMLSchema#dateTime>.
   ];
   a sioca:Action.
```

Listing 2. Representing the "When" element in Turtle syntax

In this context we consider actions to be instantaneous. As in [4] we track the instant that an action is taking effect on a wiki (*i.e.* when a wiki page is saved). Usually, this creation time is represented using dc:created. Another option, provided by the LODE ontology, uses the lode:atTime property to link to a class representing a time interval or an instant.

D. Where

The *Where* element represents the online "Space" or the location associated with an event. In wikis, and in particular in Wikipedia, this is one of the most controversial elements of the W7 model. If the location of an article update might be considered as the location of the user when updating the content, then this information on Wikipedia is not completely provided or accurate. Indeed we can extract this information only from the IP address of the anonymous users but not from all the Wikipedia users. To note that is possible to link a sioc:UserAccount (*e.g.* ⟨http://en.wikipedia.org/wiki/User:96.245.230.136⟩) to the related IP address using the SIOC ip_address property.

E. Who

The *Who* element describes an agent involved in an event, therefore it includes a person or an organization. On a wiki it represents the editor of a page, and it can be either a registered user or an anonymous user. A registered user might also have different roles in the Wikipedia site and, on this basis, different permissions are granted to its account. With this work we are only interested in keeping track of the user account involved in each event, and not also in the role on the wiki. Users are modelled with the sioc:UserAccount class and linked to each sioca:Action, sioct:WikiArticle and diff:Diff with the property sioc:has_creator. A sioc:UserAccount represents a user account, in an online community site, owned by a physical person or a group or an organization (*i.e.* a foaf:Agent). Hence a physical person, represented by a foaf:Person subclass of foaf:Agent, can be linked to several sioc:UserAccount.

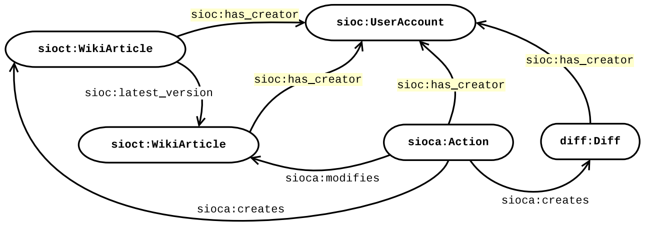


Fig. 2. Modeling the *Who* element with sioc:UserAccount

F. Which

The *Which* element represents the programs or the instruments used in the event. In our particular case it is the software used in editing the event, which might be a bot or the wiki software used by the editor. Since there is not a direct and precise way to identify whether the edit has been made by a human or a bot, our model does not make this distinction. A naive method could be to look at the username and check if it contains the "bot" string.

G. Why

The *Why* element represents the reasons behind the event occurrence. On Wikipedia it is defined by the justifications for a change inserted by a user in the "comment" field. This is

[11] For all the namespaces see: http://prefix.cc

not a mandatory field for the user when editing a wiki page but the Wikipedia guidelines recommend to fill-in this text field. We model the comment left by the user with a property `diff:comment` linking the `diff:Diff` class to the related `rdfs:Literal`.

V. Application using provenance data from Wikipedia

A. Collecting the data from the Web

In order to validate and test our modelling solution for provenance on wikis and in particular from the Wikipedia website, we collected data from the English Wikipedia and the DBpedia service. The DBpedia project[12] since it extracts and publishes structured information from the English Wikipedia, is considered as its RDF export. Collecting data not only from Wikipedia but also from the DBpedia source has an important advantage: it directly provides us structured data modelled with popular standard lightweight ontologies in RDF. We use the DBpedia data especially for the categories that hierarchically structure the articles on Wikipedia. We ran our experiment collecting a portion of the Wikipedia articles, and in particular the articles belonging to the whole hierarchy under a given category. By doing this we could limit our dataset only to articles strongly related with each other, and collect a user community with the same interest in common.

A PHP script has been developed to extract all the articles belonging to a category and all its subcategories, and for each article all its revision history. More in detail, this program:

- Executes a SPARQL[13] query over the DBpedia endpoint to get the categories hierarchy;
- Stores the categories hierarchy (modelled with the SKOS[14] vocabulary) in a local triplestore;
- Queries again the DBpedia endpoint to get all the articles belonging to the categories collected;
- For all the articles collected it generates (and stores locally) RDF data using the SIOC-MediaWiki exporter[15];
- Using the `sioc:previous_version` property it exports RDF for all the previous revisions of each article.

It is clear the advantage of using DBpedia in this process since we collected structured data just executing two lightweight SPARQL queries.

A second PHP script has been developed to extract detailed provenance information from the articles collected with the previous step. This script calculates the *diff* function between consecutive versions of the articles, and retrieves more related information from the Wikipedia API. The data retrieved from the API is composed by all the information needed for the creation of the model described in the previous section. Therefore information about the editor, the timestamp, the comment and the ID of the versions are identified. Moreover the algorithm is not only capable of extracting the *diff* function, but also to compute the type of change for each of the differences identified. This allows us to mark each change with one of the *Sentence* or *Reference Insertion/Update/Deletion* subclasses of the `diff:Diff` class. Finally the script generates RDF data with the model described before and inserts it in the local triplestore. In order to test our application we ran the data extraction algorithm starting from the category "*Semantic Web*" on the English Wikipedia, and we generated data for all the 166 wiki articles belonging to this category and its subcategories recursively. As we can see, using Semantic Web technologies, we have the advantage of having a single and standard language to query wiki and provenance data together, while developers that need to query original systems have to learn a new API for each new system we want to query.

B. A Firefox plug-in for provenance from Wikipedia

In order to show the potential of the data collected and the data model created, we built an application to show some interesting statistics extracted from provenance information of the analyzed articles. The application displays a table directly on the top of each Wikipedia article exposing some information about the most active users on the article and their edits. In particular this has been developed using a Greasemonkey[16] script: a Mozilla Firefox extension that allows users to install scripts that make on-the-fly changes to HTML web page content. This script is developed in JavaScript language and is now compatible with other popular Web browsers. The structure of the application is then composed by the following elements: 1) **The triplestore** containing the data collected and exposing a SPARQL endpoint for querying the data; 2) **A PHP script**, used as an interface between the Greasemonkey script and the triplestore; 3) **A Greasemonkey script**, which retrieves the URL of the Wikipedia loaded page, sends the request to the PHP script and then displays the returned HTML data on the Wikipedia page. The PHP script in this application is important because it is responsible for executing the SPARQL queries on the triplestore. Furthermore it retrieves the results and creates the HTML code to embed on the Wikipedia page. A screenshot of the result of the process is displayed in Figure 3.

The tables displayed in Figure 3 appear only on the top of the Wikipedia articles and categories that we analyzed with the method described in Section V-A. A different type of table is showed when the page visited is a category page. In Figure 3 on the top table, we can see the top six users who did the biggest number of edits on the article. For each of these users we then compute: (1) their total number of edits on the page; (2) their percentage of "ownership" on the page (or better, the percentage of their edits compared to all the edits done on the article); (3) their number of lines added on the article; (4) their number of lines removed on the article; (5) their total number of lines added and removed on all the articles belonging to the category *"Semantic Web"*. With the other use-case, when the user visits a Wikipedia category page, we display different

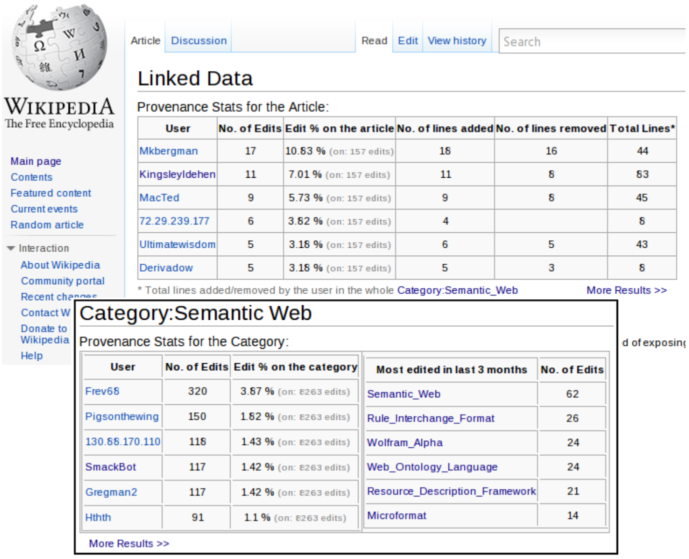
[12] http://dbpedia.org

[13] Query Language for RDF: http://www.w3.org/TR/rdf-sparql-query/

[14] SKOS Reference: http://www.w3.org/TR/skos-reference/

[15] http://ws.sioc-project.org/mediawiki/

[16] http://www.greasespot.net/



Linked Data

Provenance Stats for the Article:

User	No. of Edits	Edit % on the article	No. of lines added	No. of lines removed	Total Lines*
Mkbergman	17	10.83 % (on: 157 edits)	18	16	44
KingsleyIdehen	11	7.01 % (on: 157 edits)	11	8	83
MacTed	9	5.73 % (on: 157 edits)	9	8	45
72.29.239.177	6	3.82 % (on: 157 edits)	4		8
Ultimatewisdom	5	3.18 % (on: 157 edits)	6	5	43
Derivadow	5	3.18 % (on: 157 edits)	5	3	8

* Total lines added/removed by the user in the whole Category:Semantic_Web   More Results >>

Category:Semantic Web

Provenance Stats for the Category:

User	No. of Edits	Edit % on the category
Frev68	320	3.87 % (on: 8263 edits)
Pigsonthewing	150	1.82 % (on: 8263 edits)
130.88.170.110	118	1.43 % (on: 8263 edits)
SmackBot	117	1.42 % (on: 8263 edits)
Gregman2	117	1.42 % (on: 8263 edits)
Hthth	91	1.1 % (on: 8263 edits)

Most edited in last 3 months	No. of Edits
Semantic_Web	62
Rule_Interchange_Format	26
Wolfram_Alpha	24
Web_Ontology_Language	24
Resource_Description_Framework	21
Microformat	14

More Results >>

Fig. 3. A screenshot of the application on the "Linked_Data" page and the table from the Category "Semantic_Web" page

types of information but using the same method. See the table on the bottom in Figure 3. Browsing a wiki category page, the application shows a list of the users with the biggest number of edits on the articles of the whole category (and related subcategories). It also shows the related percentages of their edits compared to the total edits on the category. The second table on the right exposes a list of the most edited articles in the category during the last three months. To note also that at the bottom of each table there is a link pointing to a page where a longer list of results will be displayed.

At the moment the PHP script developed is available at http://vmuss06.deri.ie/WikiProvenance/index.php. Just using this script is possible to have the same information displayed using the Greasemonkey script and also to have the RDF descriptions of the page requested. In order to represent these statistical information in RDF, we use SCOVO, the Statistical Core Vocabulary [8]. It relies on the concept of *Item* and *dimensions* to represent statistical information. In our context, the item is one piece of statistical information (*e.g. user "X" edited 10 lines on page "Y"*), and various items are involved in the description: (1) the type of information that we want to represent (number of edits, percentage, lines added and removed etc.); (2) the page or the category impacted; (3) the user involved. Hence, we created four instances of `scv:Dimension` to represent the first dimension, and relied then simply on the `scv:dimension` property for the other ones. As an example, the following snippet represents that the user *KingsleyIdehen* made 11 edits on the *SIOC* page.

```
ex:123 a scovo:Item ;
    rdf:value 11 ;
    scv:dimension :Edits ;
    scv:dimension <http://wikipedia.org/wiki/SIOC>;
    scv:dimension <http://wikipedia.org/wiki/User:
          KingsleyIdehen>.
```

Listing 3. Representing the number of edits by a user with SCOVO

VI. Conclusion and Future Work

The goal of this paper was to provide a solution for representing and managing provenance of data from Wikipedia (and other wikis) using Semantic Web technologies. To solve this problem we provided: a specific lightweight ontology for provenance in wikis, based on the W7 model; a framework for the extraction of provenance data from Wikipedia; an application for accessing the generated data in a meaningful way and exposing it to the Web of data. We showed that the W7 model is a good choice for modelling provenance information in general and in wikis but, because of its high abstraction level, it has to be refined using for instance other specific lightweight ontologies. In our case this has been done using SIOC and the Actions module. Future developments will include a refinement of the proposed model and a subsequent alignment with other general-purpose ontologies for representing provenance as Linked Data (*e.g.* the *Open Provenance Model*). We also plan to improve and extend the potentialities of our application offering more features, and providing a wider range of data with an architecture that automatically updates the data as soon as it changes on Wikipedia.

References

[1] SIOC Core Ontology Specification. W3C Member Submission 12 June 2007, World Wide Web Consortium, 2007. http://www.w3.org/Submission/sioc-spec/.

[2] B.T. Adler, L. de Alfaro, I. Pye, and Vishwanath Raman. Measuring author contributions to the wikipedia. In *Proceedings of WikiSym '08*. ACM, 2008.

[3] Mario Bunge. *Treatise on Basic Philosophy: Ontology I: The Furniture of the World*. Riedel, Boston, 1977.

[4] P.A. Champin and A. Passant. SIOC in Action - Representing the Dynamics of Online Communities. In *Proceedings of the 6th International Conference on Semantic Systems (I-SEMANTICS 2010)*. ACM, 2010.

[5] J. Golbeck, B. Parsia, and J. Hendler. Trust networks on the semantic web. *Cooperative Information Agents VII*, pages 238–249, 2003.

[6] Olaf Hartig. Provenance information in the web of data. In *2nd Workshop on Linked Data on the Web (LDOW 2009) at WWW*, 2009.

[7] Olaf Hartig and Jun Zhao. Publishing and Consuming Provenance Metadata on the Web of Linked Data. In *Proceedings of 3rd Int. Provenance and Annotation Workshop*, 2010.

[8] M Hausenblas, W Halb, Y Raimond, L Feigenbaum, and D Ayers. SCOVO: Using statistics on the Web of data. In *Semantic Web in Use Track of the 6th European Semantic Web Conference (ESWC2009)*, 2009.

[9] B Hoisl, W Aigner, and S Miksch. Social Rewarding in Wiki Systems–Motivating the Community. In *Proceedings of the 2nd international conference on Online communities and social computing*, pages 362–371. Springer-Verlag, 2007.

[10] NT Korfiatis, M Poulos, and G Bokos. Evaluating authoritative sources using social networks: an insight from Wikipedia. *Online Information Review*, 2006.

[11] Fabrizio Orlandi and Alexandre Passant. Enabling cross-wikis integration by extending the SIOC ontology. In *4th Semantic Wiki Workshop (SemWiki 2009)*. CEUR-WS, 2009.

[12] Fabrizio Orlandi and Alexandre Passant. Semantic Search on Heterogeneous Wiki Systems. In *International Symposium on Wikis (WikiSym2010)*. ACM, 2010.

[13] Sudha Ram and Jun Liu. *Understanding the semantics of data provenance to support active conceptual modeling*, pages 17–29. Springer Berlin / Heidelberg, lncs edition, 2007.

[14] Sudha Ram and Jun Liu. A New Perspective on Semantics of Data Provenance. In *First International Workshop on the role of Semantic Web in Provenance Management (SWPM 2009)*, 2009.

[15] Y.L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance techniques. *Computer Science Department, Indiana University, Bloomington IN*, 47405, 2005.