Submitted to the 22nd International Conference
on Scientific and Statistical Database Management (SSDBM 2010).
draft material -- please do not cite

Provenance Context Entity (PaCE): Scalable Provenance Tracking for Scientific RDF Data

Satya S. Sahoo[1], Olivier Bodenreider[2], Pascal Hitzler[1], Amit Sheth[1], Krishnaprasad Thirunarayan[1],

[1] Kno.e.sis Center, Computer Science and Engineering Department, Wright State University, Dayton, OH, USA
[2] Lister Hill National Center for Biomedical Communications, National Library of Medicine,NIH, Bethesda, MD, USA
{sahoo.2, pascal.hitzler, t.k.prasad, amit.sheth}@wright.edu, obodenreider@mail.nih.gov

Abstract. The Semantic Web Resource Description Framework (RDF) format is being used by a large number of scientific applications to store and disseminate their datasets. The provenance information, describing the source or lineage of the datasets, is playing an increasingly significant role in ensuring data quality, computing trust value of the datasets, and ranking query results. Current Semantic Web provenance tracking approaches using the RDF reification vocabulary suffer from a number of known issues, including lack of formal semantics, use of blank nodes, and application-dependent interpretation of reified RDF triples that hinders data sharing. In this paper, we introduce a new approach called Provenance Context Entity (PaCE) that uses the notion of *provenance context* to create provenance-aware RDF triples without the use of RDF reification or blank nodes. We also define the formal semantics of PaCE through a simple extension of the existing RDF(S) semantics that ensures compatibility of PaCE with existing Semantic Web tools and implementations. We have implemented the PaCE approach in the Biomedical Knowledge Repository (BKR) project at the US National Library of Medicine to support provenance tracking on RDF data extracted from multiple sources, including biomedical literature and the UMLS Metathesaurus. The evaluations demonstrate a minimum of 49% reduction in total number of provenance-specific RDF triples generated using the PaCE approach as compared to RDF reification. In addition, using the PACE approach improves the performance of complex provenance queries by three orders of magnitude and remains comparable to the RDF reification approach for simpler provenance queries.

Keywords: Provenance context entity, Biomedical knowledge repository, Context theory, RDF reification, Provenir ontology, Provenance Management Framework.

1 Introduction

An increasing number of scientific applications are storing and disseminating their datasets using the Semantic Web Resource Description Framework (RDF) format [1] [2] [3]. RDF is also being used as an information integration platform in multiple scientific domains. The Biomedical Knowledge Repository (BKR) project at the U.S.

National Library of Medicine is creating a comprehensive repository of integrated biomedical data from a variety of sources such as biomedical literature (textbooks and journal articles), structured data bases (for example the NCBI Entrez system [4]), and terminological knowledge sources (for example, the Unified Medical Language System (UMLS) [5]) [6]. BKR represents the integrated information in RDF, for example, the RDF statement "`lipoprotein→affects→inflammatory_cells`"[1] was extracted by a text mining tool from a journal article (with PubMed identifier PMID: 17209178) and states that `lipoprotein` (denoted as "subject" of the RDF triple[2]) `affects` (denoted as "property" of the triple) `inflammatory_cells` (denoted as the "object" of the triple). In addition to the advantages of more expressive modeling [7], storing information as RDF statements enables BKR to be compatible with the rapidly growing Linked Open Data (LOD) initiative that currently has more than 4.2 billion RDF statements representing a large number of domains including biomedicine, census data, chemistry, and geography [8].

In addition to the biomedical data, BKR also records and uses provenance metadata describing the history or lineage of the RDF statements. The provenance information identifies the source of an extracted RDF triple, temporal information (for example, the date of publication of a source article), version information for a database, and the confidence value associated with a triple (indicated by a text mining tool). The provenance information is essential in the BKR project to ensure the quality of data and associate trust value with the RDF triple. It has specific applications in the four services offered by the BKR namely, enhanced information retrieval (search based on the named relationship linking two entities), multi document summarization, question answering, and knowledge discovery. We describe example scenarios that highlight the use of provenance information in the four services offered by BKR:

1. *Enhanced information retrieval service*: Locate all documents that mention the RDF statement `lipoprotein→affects→inflammatory_cells`. A similar query uses the provenance metadata to identify all RDF triples extracted from a particular document.
2. *Multi-document summarization*: Rank RDF statements from multiple documents using the confidence associated with each statement (indicated by the text mining tool).
3. *Question answering service*: Specify that the answers should be sourced only from reputable entities (for example, curated databases) or extracted from a journal article published recently (e.g., during the past year).
4. *Knowledge discovery service*: Using reasoning rules, implicit knowledge can be inferred from existing RDF triples in the BKR project. Often, provenance of the original triples is required to accurately interpret new triples. The application of reasoning rules can also be restricted to a specific set of RDF triples based on their provenance.

To address the above requirements, BKR collects the provenance information associated with an RDF triple at two levels. At the first level, provenance information directly associated with a RDF triple is collected, including the source of the triple

[1] We use the `courier new` font to represent RDF and OWL statements.

[2] We use the notions *RDF statement*, *RDF triple*, and *triple* interchangeably in the rest of this paper.

draft

(journal article, database) or some confidence value associated with it. At the second level, BKR records additional provenance information collectively associated with a set of triples. For example, all triples extracted from a given journal article inherit the date of publication, author names, and set of index terms of this particular journal article (for example, in Medline [9]).

The RDF reification vocabulary [10] has been traditionally used by Semantic Web applications to track provenance in RDF documents. The RDF reification vocabulary consists of the four terms `rdf:Statement`,[3] `rdf:subject`, `rdf:predicate`, and `rdf:object`. Figure 1 illustrates the two levels of provenance recorded for the triple "`lipoprotein→affects→inflammatory_cells`" using RDF reification. A variety of problems have been identified in the use of RDF reification vocabulary for provenance tracking in Semantic Web applications and we discuss these issues in the next section.

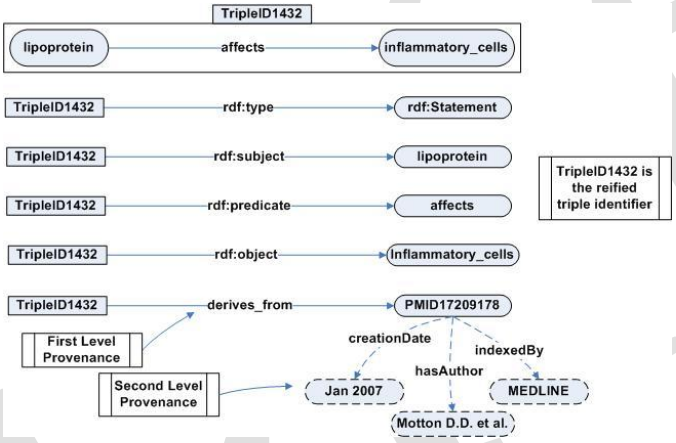

Figure 1: Schematic representation of using RDF reification to track provenance of a triple

1.1 Limitations of the RDF Reification and Related Approaches

The limitations of the RDF reification vocabulary are discussed along two dimensions, namely, (a) formal semantics, and (b) implementation issues for real world applications. The RDF specification [11] states that the RDF formal semantics does not extend to the reification vocabulary, and the intended interpretation of an RDF document using reification is application dependent (i.e., it may vary across applications) [10]. In addition to limited formal semantics, the RDF syntax does not provide a mechanism to link the reified triple to the RDF statement itself [10]. For example, there is no support in the RDF syntax to link "TripleID1432" in Figure 1 to the triple `lipoprotein→affects→inflammatory_cells`. The lack of support for consistent interpretation of RDF documents using reification is a significant

[3] The `rdf` namespace represents the http://www.w3.org/1999/02/22-rdf-syntax-ns Internationalized Resource Identifier (IRI).

challenge for scientific projects such as BKR that aim to serve a large community of researchers and to support multiple applications. The RDF specification describes a "conventional use" of the reification vocabulary [10] where "the subject of a reification triple" is a specific RDF triple in a particular RDF document and not any RDF triple (that may also have the same subject (S), predicate (P), and object (O)). Specifically, the assertion `TripleID1432→derives_from→PMID17209178` in Figure 1 is not applicable to all triples sharing the same S, P, O.

Inference rules are an important component of Semantic Web applications, especially for knowledge discovery tasks in projects such as the BKR. But, the RDF specification states that entailment rules do not hold between an RDF triple and its reification [11]. Further, the use of blank nodes, which have no "global meaning" outside a particular RDF graph [11] and have no corresponding real world entities in scientific domains, is a significant challenge to Semantic Web applications relying on reification. The use of blank nodes makes it difficult to use reasoning [12] and increases the complexity of query patterns since the queries have to explicitly take into account an extra entity (that cannot be "typed" as instance of domain ontology class) in the query pattern.

We now describe the implementation specific limitation of RDF reification. Though incorporating additional metadata descriptions in form of provenance information necessarily increases the total size of an RDF document, the RDF reification approach leads to a disproportionate increase in the total size of the RDF document without corresponding enhancement in information content of the RDF document. For example, as illustrated in Figure 1, reification of a single RDF triple leads to the creation of four extra RDF triples that do not model any provenance-related information but are merely artifacts of the RDF syntax. This would adversely affect the scalability of large projects, such as BKR, that track provenance of hundreds of millions of RDF triples.

We now briefly describe two approaches, namely RDF named graph [13] and RDF molecule [14], that enable RDF provenance tracking at different levels of granularity. The named graph approach, part of the RDF specification, associates an identifier to an RDF graph that allows applications to make assertions about a set of RDF triples contained in the graph [13]. The named graph approach also defined the syntax, semantics and its relationship to RDF triples. The limitations of the named graph include its coarse granularity (that makes it impractical for use in real world applications) and the use of blank nodes. The RDF molecule is a similar approach to track provenance information at a finer level of granularity through lossless decomposition of a RDF graph to identify sub-graphs but not for a triple, using blank nodes [14]. In [15], a generalization of the RDF named graphs is proposed called "colored RDF triples" that uses a semigroup structure to reason over provenance information, but this work does not address the primary disadvantage of the named graph approach, that is, use of blank nodes.

In this paper, we introduce a new approach called Provenance Context Entity (PaCE) to enable provenance tracking in Semantic Web applications using neither RDF reification vocabulary nor blank nodes. The PaCE approach creates the S, P, O RDF entities that reflect the provenance requirements of a Semantic Web application. PaCE is part of a broader framework for provenance management in scientific applications called PrOM [16]. PrOM consists of a foundational upper-level ontology

called provenir (to facilitate provenance interoperability), a set of dedicated provenance query operators and a query engine implemented for RDF data stores [16].

1.2 Contributions and Overview

The contributions of this paper are four-fold:

1. Define the PaCE approach to track provenance in RDF-based Semantic Web applications without use of reification vocabulary and blank nodes (Section 2),
2. Define the formal semantics of PaCE, using model theory, by extending the existing RDF and RDFS formal semantics to ensure compatibility with existing RDF tools and implementations (Section 2 and 3),
3. Demonstrate the practical feasibility of PaCE through implementation in the BKR project (Section 4), and
4. Evaluate the advantages of PaCE in terms of storage and query performance as compared to the RDF reification approach (Section 5).

We conclude in Section 6.

2 Foundations of Provenance Context Entity

The intuition for the PaCE approach is that the provenance associated with RDF statements provides the necessary contextual information for applications to interpret two RDF statements to be equivalent or distinct. Contexts as formal objects have long been used in Artificial Intelligence (AI) applications, such as Cyc [17] and also to a limited extent in the Semantic Web, to facilitate processing of information that do not have a global frame of reference [18]. The next section reviews the existing work on context theory in AI.

2.1 Context Theory in AI

Contexts were introduced as formal objects in AI systems in the 1990s [19] [17] to allow applications to process statements only in specific frames of reference. Using the construct *ist* (*c*, *p*), which asserts that a statement *p* is true in a context *c*, context theory also defined mechanisms called "lifting rules" to process statements in different contexts [19] [17]. Various advantages of using contexts include (a) ability to make domain specific assumptions, (b) selection of a manageable subset of the knowledge base, and (c) maintaining consistency within a context without the need for maintaining global consistency [17]. There has been a lot of work in context research including appropriate extensions to the model theory and description of the associated computational complexity [20-22].

Contexts have been introduced for use in the Semantic Web to address challenges faced by data aggregation applications such as the TAP project [23]. For example, two apparently contradictory statements, "John Kennedy is president of USA" and "Barack Obama is president of USA", can be reconciled using contextual metadata describing the temporal information associated with the statements. The context mechanism in the TAP project associates a context with each Web data source and all information extracted from the source is assumed to be true (in the given context)

draft

[23]. The context for Semantic Web uses the *ist* (*c*, *p*) notation with appropriate extensions to the RDF model theory [23]. As discussed earlier in [13], these extensions to the RDF model theory require significant changes to existing implementations of Semantic Web inference systems. In contrast, existing implementations can process RDF documents that use the PaCE approach, which is defined in the next section, to track provenance information.

2.2 Provenance Context and RDF Generation

The contextual information in the BKR project consists of the provenance information about the source of an RDF statement, that is, the journal identifier or the UMLS identifier or the Entrez Gene identifier. In other words, this *provenance context* is a formal object instantiated in form of set of concepts and relationships that capture the necessary contextual provenance information to enable application to correctly interpret RDF statements. Similar to the provenance context defined in the BKR project, other Semantic Web applications can also define a relevant provenance context for interpreting their RDF dataset. For example, an application in the sensor domain can define its provenance context to consist of sensor used to collect data readings, the geographical location of the sensor, and the timestamp value associated with a data reading. To formalize the notion of provenance context we define it in terms of the foundational model of provenance called provenir ontology (Figure 2) [24]. The provenir ontology is a upper-level provenance ontology representing a minimum set of provenance concepts common across domains and is modeled using the description logic profile of the W3C Web Ontology Language (OWL-DL) [25].

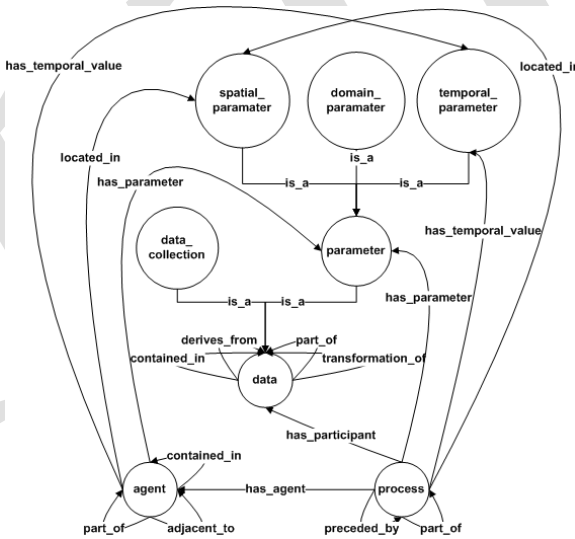

Figure 2: Upper-level provenir provenance ontology schema

The role of domain-independent upper ontologies to facilitate interoperability, consistent modeling of concepts, and uniform use of terms has led to creation of a number of ontologies such as BFO [26], DOLCE [27], and SUMO [28]. In addition to

these upper ontologies, a set of "*domain* upper ontologies" have also been proposed, such as BioTop [29] and Simple Bio Upper ontology [30], that serve as intermediate level ontologies between the highly specific domain ontologies (for example, Gene Ontology [31]) and abstract upper ontologies. The provenir ontology is one such upper domain ontology for provenance modeling that facilitates consistent modeling of interoperable provenance information. The provenir ontology has been successfully extended to create domain-specific provenance ontologies in multiple projects [24] [32], including the Parasite Experiment ontology that has been listed in the BioPortal, the ontology repository of the National Center for Biomedical Ontologies (NCBO).

The provenir ontology consists of three primary concepts of "`data`", "`agent`" and "`process`" linked by ten relationships adapted from the upper-level Relation Ontology [33] (Figure 2). An application can define its provenance context either in terms of the provenir ontology or in terms of a domain-specific provenance ontology, which extends provenir ontology. For example, the BKR project uses the UMLS Semantic Network (SN) [5] as the domain ontology and hence defines its provenance context in terms of the SN (in section 4 we describe the mapping of relevant SN terms to the provenir ontology).

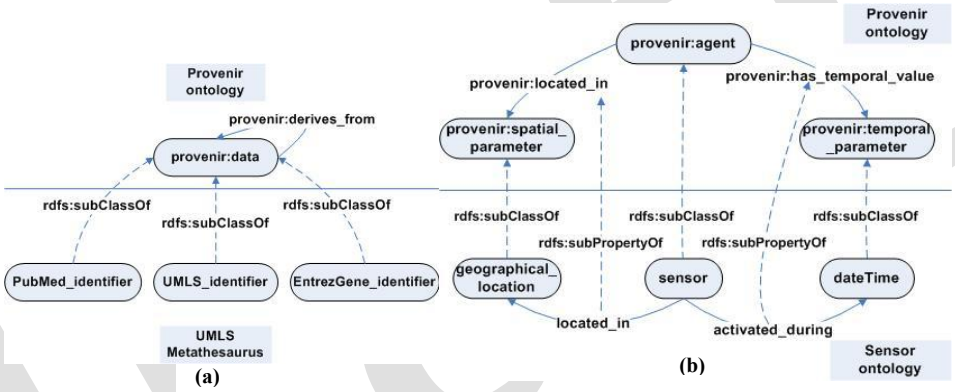

Figure 3: Schematic representation of provenance context for the BKR project and a sensor application

Figure 3 (a) illustrates the BKR provenance context for the BKR project and Figure 3 (b) illustrates the provenance context for the sensor domain (consisting of the sensor identifier, the geographical location of a sensor, and date-time value associated with a sensor reading). It is important to note that the classes and properties used to define the provenance contexts for the BKR and sensor domain application can be linked to the provenir ontology using either "subclass" or "sub-property" relations. The use of the provenir ontology to define a provenance context has several advantages including the flexibility to model domain-specific provenance at a fine level of granularity, while ensuring consistent modeling and the support for RDF and OWL inferencing [11]. Further, applications can leverage a set of dedicated provenance query operators, defined in terms of the provenir ontology, as part of the broader provenance management framework called PrOM [16]. Though provenance context as a notion is derived from the context theory used in AI systems, it is distinct

in terms of both formal semantics and implementation. These differences are listed below:

1. A provenance context is used only for generating the S, P, O of an RDF triple and this approach of generating "provenance-aware" RDF triples is called the Provenance Context Entity (PaCE) approach. In contrast, traditional AI systems use context primarily during processing or interpreting data.
2. The PaCE approach involves *apriori* use of the context object during RDF triple generation; hence it does not use the *ist* (*c*, *p*) construct to interpret RDF statements. In addition, unlike the context mechanism introduced in [23], the PaCE approach does not require extensive modifications to the RDF model theory (described in Section 3).
3. The PaCE approach defines a single application-wide provenance context and unlike traditional AI systems does not include multiple context objects. Hence, the PaCE approach does not require use of lifting rules to process RDF statements in different contexts [17].

The PaCE approach allows an application to decide the level of granularity in modeling provenance of an RDF triple. For example, Figure 4 illustrates the three possible implementations of the PaCE approach in the BKR project that create distinct RDF triples extracted from two separate journal articles (though they share the same S, P, and O). The first implementation (Figure 4 (a)) is an exhaustive approach and explicitly links the S, P, and O to the source journal article and the second implementation (Figure 4 (b)) is a minimalist approach that links only the S of a RDF triple to the source article. Though the first implementation creates three provenance-specific triples, in contrast to just one triple by second implementation, there is no ambiguity in correctly interpreting the provenance of the triples. The second implementation, on the other hand, requires the application to make additional assumption, while processing the RDF triples, that the whole triple is extracted from the same source as the source of S. Hence, the additional complexity associated with the second implementation may make it unsuitable for some applications.

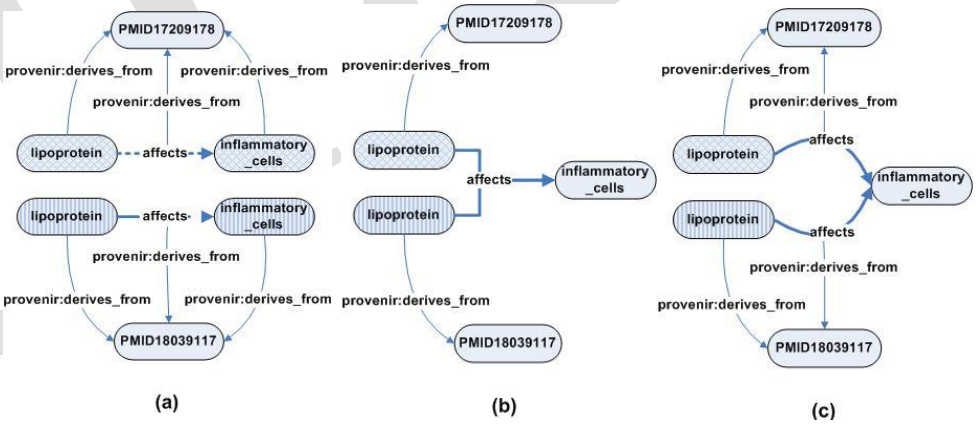

Figure 4: Implementation of the PaCE mechanism to track provenance of RDF triples extracted from two journal articles

The third implementation (Figure 4(c)) takes an intermediate approach that creates two additional provenance-specific triples but requires the application to assume that the source of the O is the same as the S, and P. Similar to the minimalist approach, this approach reduces the total number of provenance-specific RDF triples, but introduces additional complexity that may make this approach unsuitable for some applications. The choice to associate explicit "`derives_from`" property with one particular RDF component (S or P or O) in the minimalist (Figure 4 (b)) and the intermediate (Figure 4(c)) is arbitrary and has minimal impact on the provenance tracking functionality of the application.

It is important to note that, in contrast to the reification approach, none of the three variants of the PaCE approach requires the use of RDF reification vocabulary or the use of blank nodes. Further, the reification approach creates a total of six RDF triples (Figure 1) for each RDF triple, while the exhaustive implementation of the PaCE approach creates a total of four triples for one RDF triple. This difference in the total number of RDF triples generated by the reification approach versus the PaCE approach has significant impact on the scalability of applications incorporating provenance information in real world applications, such as the BKR project that includes millions of RDF triples. This significant difference in total number of provenance-related RDF triples generated using the PaCE approach as compared to the RDF reification approach is further discussed in Section 5 (Evaluation).

Overall, the PaCE approach is an incremental and simple mechanism that does not define additional vocabulary or require changes to existing RDF data stores. In addition, the PaCE approach does not require modifications to existing RDF or OWL inference systems used in many Semantic Web applications. We now introduce the formal specification of provenance context.

2.3 Formal Specification of Provenance Context

The provenance information represented by a provenance context is an RDF/XML document grounded in the provenir ontology. Specifically, a provenance context consists of:

a) Provenir ontology classes and properties
b) Classes and properties of domain-specific ontologies that extend the provenir ontology using `rdfs:subClassOf`[4] and/or `rdfs:subPropertyOf`, where the provenir ontology classes and properties are the "super class" or "super property".
c) Instances (created using the `rdf:type` statements)

A detailed description of the provenir ontology classes and relationships is presented in [16]. In the next section, we introduce the formal semantics of PaCE that allows the definition of valid inference rules for PaCE provenance information in Semantic Web applications.

[4] The `rdfs` namespace represents the http://www.w3.org/2000/01/rdf-schema IRI

draft

3 Model Theoretic Semantics of PaCE Inferencing

The primary motivating factor for defining the formal semantics of PaCE is to provide a way to determine the validity of the inferencing process for Semantic Web applications that use the PaCE approach to track provenance. For example, the BKR project can derive a ranking of RDF statements extracted from journal articles by inferring the confidence value of an RDF statement from the precision and recall values indicated by a text mining tool. To define the formal semantics of PaCE we use model theory. Specifically, we build on the approach used to define the formal semantics for RDF and RDF Schema (RDFS) [11] to define the model theoretic semantics of PaCE. So our definition is based on the notions of *interpretations* and *models*, which are structures that enable us to capture information about truth values (true or false) of assertions [11]. In other words, if a particular interpretation I satisfies a specific assertion $s \in V$ then we call I a model of s and write $I \models s$ in this case (where V is a vocabulary and $\models$ is the so-called entailment relation).

Following [11], a *simple* interpretation I of a vocabulary V consists of

- a non empty set of *resources* IR that constitutes the domain or universe of I,
- a set of *properties* of I called IP,
- a function I_{EXT}: $IP \rightarrow 2^{IR \times IR}$ that maps each property in IP to a pair of resources in IR,
- a function I_S: $V \rightarrow IR \cup IP$ which maps IRIs in V to the union of IR and IP,
- a function IL which maps typed literals from V to resources in IR, and
- a subset of IR called *set of literal values*, LV, containing all untyped literals from V,

Each interpretation I then gives rise to an interpretation function $\cdot^I$, which maps each triple (over IR, IP, V and LV) to a truth value (true or false) in a canonical way (see [11]) An interpretation I of a graph R is said to be a *model* of R if I maps each triple in R to the truth value true. We write $I \models R$ in this case. Simple interpretations allows us to define an entailment relation between graphs, that is, a graph R_1 (simply) entails a graph R_2 if every simple interpretation that is a model of R_1 is also a model of R_2. A simple interpretation of a vocabulary $V \cup V_{RDF} \cup V_{RDFS}$ is called an *RDFS interpretation* of V if it satisfies a number of additional constraints specified in [11]. We say that a graph R_1 RDFS-entails a graph R_2 if every RDFS interpretation that is a model of R_1 is also a model of R_2.

The definition of the model-theoretic semantics of PaCE is a straightforward modification of the existing RDFS semantics and allows us to infer additional provenance information for triples by virtue of having similar source. Let provenance context pc of an RDF triple α (= (S, P, O)) be the common object of the predicate `provenir:derives_from` associated with the triple. We define an *RDFS-PaCE-interpretation* I of a vocabulary V to be an RDFS-interpretation of the vocabulary $V \cup V_{PaCE}$ that satisfies the following additional condition (meta-rule):

- For RDF triples $\alpha = (S_1,P_1,O_1)$ and $\beta = (S_2,P_2,O_2)$, (provenance-determined) predicates *p* and entities *v*,
  if $pc(\alpha) = pc(\beta)$
  then $(S_1, p, v) = (S_2, p, v)$ and, $(P_1, p, v) = (P_2, p, v)$ and, $(O_1, p, v) = (O_2, p, v)$

Provenance-determined predicates and entities are specific to the application domain.

Furthermore, a graph R_1 PaCE-entails a graph R_2 if every *RDFS-PaCE-interpretation* that is a model of R_1 is also a model of R_2. To illustrate the PaCE inference process, we consider two RDF statements in the BKR project (Figure 5). Given that the two RDF statements have equal provenance contexts (PubMed identifier: PMID17209178) additional provenance information, such as the confidence score (formalized via provenance-related predicate `has_confidence_value` and value `confidence_value_2`), associated with one of the triples can be inferred for the other RDF triple (`flow_cytometry`→`measures`→`interleukin-1_beta`) denoted by dotted arrows in Figure 5.

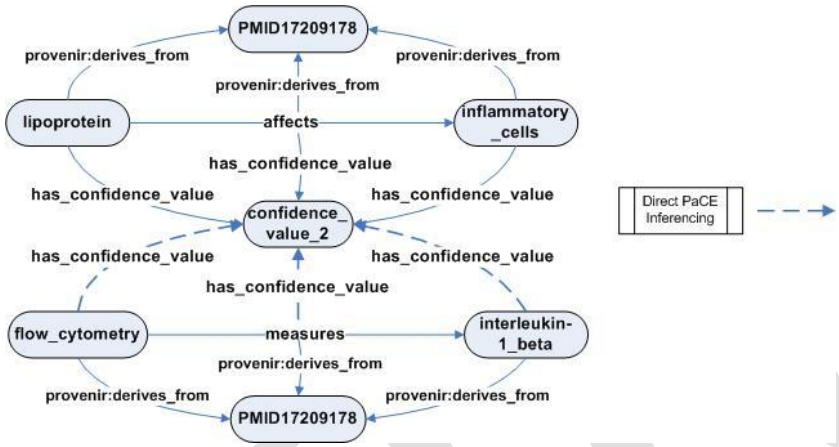

Figure 5: PaCE inferencing

We note that PaCE-entailment is strictly stronger than RDFS-entailment in the sense that all inferences which can be drawn using simple, RDF, or RDFS-entailment are also PaCE entailments. This is a deliberately conservative step on top of the existing Semantic Web recommendations that enables PaCE to be compatible with existing OWL and RDF tools and applications, and also allows implementing the PaCE-semantics by making reference to RDF reasoners as black boxes. In the next section, we describe the implementation of the BKR project using the PaCE approach.

4 Implementation: Using PaCE Approach in the BKR Project

Implementing the PaCE approach in BKR project involves two steps, namely, (a) Extending the provenir ontology with domain-specific concepts that serve as a reference for the definition of contextualized instances in the BKR, and (b) generating instance-level RDF triples leveraging the provenance model for data from several sources. We first describe the extension of the provenir ontology.

4.1 Extending the Provenir Ontology with Domain-specific Concepts

The provenir ontology provides a domain-independent model for provenance, which needs to be extended with domain-specific concepts in order to support the creation of contextualized instances. We use the Unified Medical Language System (UMLS) as

our main source of biomedical concepts [34]. More specifically, high-level categories (semantic types) from the UMLS Semantic Network can be integrated as subclasses (`rdfs:subClassOf`) of the provenir classes `provenir:data` or `provenir:event`. In turn, the two million concepts from the larger UMLS Metathesaurus are integrated as subclasses of the semantic types, using the categorization link provided by the UMLS. The provenir ontology is also extended with some 27,000 genes from Entrez Gene for better coverage of genomic entities.

Instances in the BKR are defined in reference to these domain-specific concepts. Analogously, instance-level predicates in the BKR are defined as subproperties (`rdfs:subPropertyOf`) of the 53 named relationships provided by the UMLS Semantic Network. A mapping between instance-level and Semantic Network predicates was created manually. In addition to links (`rdf:type`) between instances from the BKR and the corresponding classes, the definition of BKR provenance context in enabled through the `provenir:derives_from` relationship linking a triple to its source. The `provenir:derives_from` property is adapted from the `derives_from` property defined in the upper-level Relation Ontology and is used to model the "derivation history of data entities as a chain or pathway" [33].

4.2 Generating RDF Triples using the PaCE Approach

Contextualized RDF triples in the BKR represent knowledge extracted from the biomedical literature, as well as relations from the UMLS Metathesaurus. RDF triple entities (S, P, O) are identified using an unique identifier called the Uniform Resources Identifier (URI). A practical challenge for implementing the PaCE approach in the BKR is to formulate an appropriate provenance context-based URI (URI_p) scheme that also conforms to best practices of creating URIs for the Semantic Web, including support for use of HTTP protocol [35].

The design principle of URI_p is to incorporate a "provenance context string" as the identifying reference of an entity and is a variation of the "reference by description" approach that uses a set of description to identify an entity [35]. The syntax for URI_p consists of the `<base URI>`, the `<provenance context string>`, and the `<entity name>`. For example, the URI_p for the entity `lipoprotein` is `http://mor.nlm.nih.gov/bkr/PUBMED_17209178/lipoprotein` where the `PUBMED_17209178` provenance context string identifies the source of a specific instance of `lipoprotein`.

This approach to create URIs for RDF entities also enables BKR (and other Semantic Web applications using the PaCE approach) to group together entities with the same provenance context. For example,

- `http://mor.nlm.nih.gov/bkr/PUBMED_17209178/lipoprotein`
- `http://mor.nlm.nih.gov/bkr/PUBMED_17209178/affects`
- `http://mor.nlm.nih.gov/bkr/PUBMED_17209178/inflammatory_cells`

are entities extracted from the same journal article. Using this URI scheme, RDF statements were generated for the original triples (extracted from the biomedical literature by a text-mining application or found in the UMLS Metathesaurus).

In the next section, we evaluate the implementation of the PaCE approach to track provenance in the BKR project as compared to the RDF reification approach.

draft

5 Evaluation

The objective of our experiment is to evaluate the advantages of using the PaCE approach in place of the RDF reification approach to track provenance in the BKR project. Three specific aspects are investigated:

1. Measure the burden of representing provenance information, in number of triples required, compared to a "base dataset" (B) with no provenance information
2. Analyze the performance of four BKR provenance queries
3. Demonstrate the use of provenance information to support analytical queries in the BKR project and measure the associated cost in performance

The base dataset (B) comprises of 23,433,657 RDF triples extracted from two sources: the biomedical literature (PubMed) and the UMLS Metathesaurus.

The open source Virtuoso RDF store version 06.00.3123 was used for the experiments running on a Dell 2950 server (Dual Xeon processor) with 8GB of memory. A total of 500,000 9kB buffers were allocated to Virtuoso RDF store.

5.1 Number of Provenance-specific RDF Triples Generated

To evaluate the number of provenance-specific RDF triples generated using the two approaches, we augment the base dataset B with provenance information representing the source information of each triple. For the PaCE approach, we create three datasets representing the exhaustive (E_PaCE), minimalist (M_PaCE), and intermediate (I_PaCE) approaches illustrated in Figure 4 (a), 4 (b) and 4 (c), respectively. For the RDF reification dataset (R), we use the standard method (presented in Section 1). Figure 6 shows that the reification approach requires twice as many RDF triples (~152 million) for the representation of provenance information compared to the E_PaCE approach (~89 million). This 49% difference between E_PaCE and R represents a significant reduction in storage requirements (~85 million fewer triples) for the BKR project and, more generally, clearly demonstrates the practical benefits of using the PaCE approach over reification to track provenance in Semantic Web applications. Analogously, the M_PaCE and I_PaCE approaches create 72% and 59% fewer provenance-specific triples compared to the reification approach.

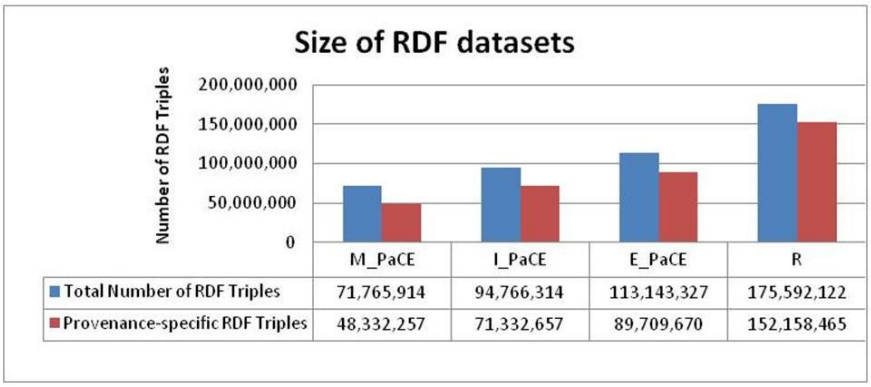

Figure 6: The relative number of provenance-specific triples created using PaCE and RDF reification

5.2 Performance of Provenance Queries

We use four representative categories of provenance queries in the BKR project to evaluate the query performance for the four datasets (E_PaCE, M_PaCE, I_PaCE and Reification). We describe the *pattern* of the four queries and their significance in the BKR project:

Query Pattern 1: List all the RDF triples extracted from a given journal article (e.g., journal article identified by PMID17209178). This query is used to retrieve all the triples from a given source.

Query Pattern 2: List all the journal articles from which a given RDF triple was extracted (e.g., `lipoprotein→affects→inflammatory_cells`). This query identifies the source(s) of a given triple.

Query Pattern 3: Count the number of triples in each source (biomedical literature and UMLS Metathesaurus) for the therapeutic use (predicate = treats) of a given drug (e.g., Thalidomide). This complex query illustrates the use of the BKR as a knowledge base for a query answering application (e.g., which diseases are treated by a particular drug?).

Query Pattern 4: Count the number of journal articles published between two dates (e.f., 2000-01-01 and 2000-12-31) for a given triple (e.g., `thalidomide → treats → multiple myeloma`). This typical information retrieval query leverages the provenance information associated with each triple. A more complex version of this query is used Section 5.2 for time series analysis.

We conducted the query performance evaluation in two phases. In the first phase the four queries are evaluated for fixed values, namely the value underlined in the query description above. In the second phase, queries are evaluated using a larger set of values. The queries are expressed in SPARQL syntax, the RDF query language [36], and primarily utilize the SPARQL basic graph patterns (BGP) with FILTER conditions. The queries are not listed in the paper due to space constraints and are available online along with the result set.[5] The numbers reported for the "fixed" value queries (first phase) are the average of last 5 of a total of 20 runs. The first phase of the evaluation starts with a "cold" cache for each query pattern.

The results in Figure 7 demonstrate that query performance for PaCE is generally better than or similar to reification. As expected, M_PaCE generally performs better than I_PaCE, and I_PaCE better than E_PaCE. However, reification performs better than I_PaCE for *Query 1* and better than both I_PaCE and E_PaCE for *Query 3*. *Query 4* is a complex query that uses the SPARQL FILTER to restrict publication dates to a particular range (January 1 to December 31, 2000). In this query, the query performance for E_PaCE is more than two orders of magnitude better than for R.

In the second phase of the evaluation, we aim to reflect the real-world requirements of the BKR project. Toward this end, each of the four query patterns is executed with different values, as if by different users. In practice, we use sets of 100 values for each query pattern. The resulting set of 100 queries is run 5 times (immediately following the first phase of evaluation for each dataset) and the average of the 100 queries for the last run is presented (Figure 8).

[5] Query and result set available at: http://wiki.knoesis.org/index.php/ProvenanceContextEntity

draft

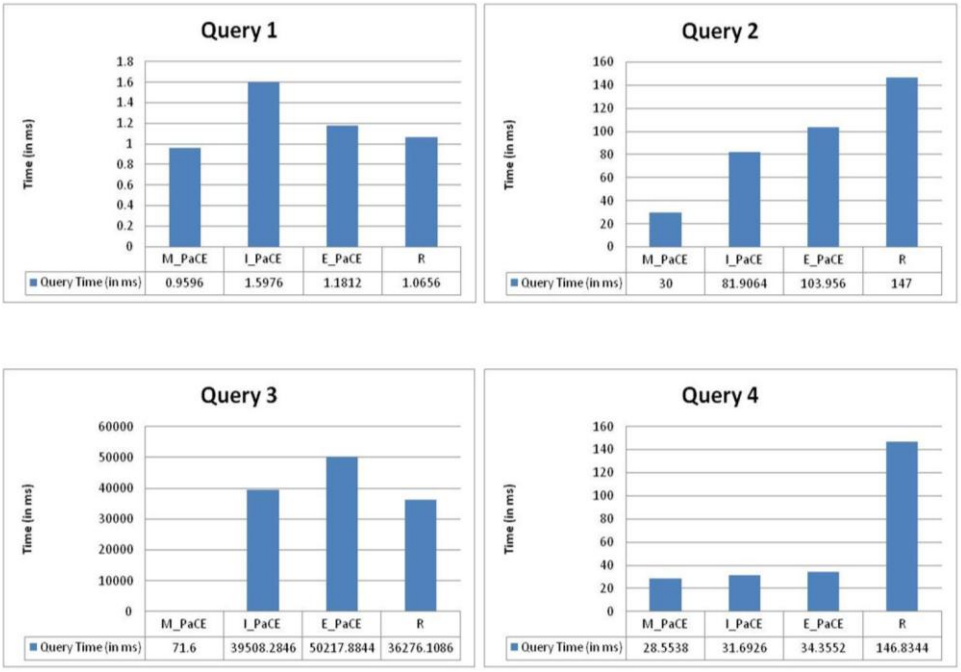

Figure 7: Query performance for fixed values

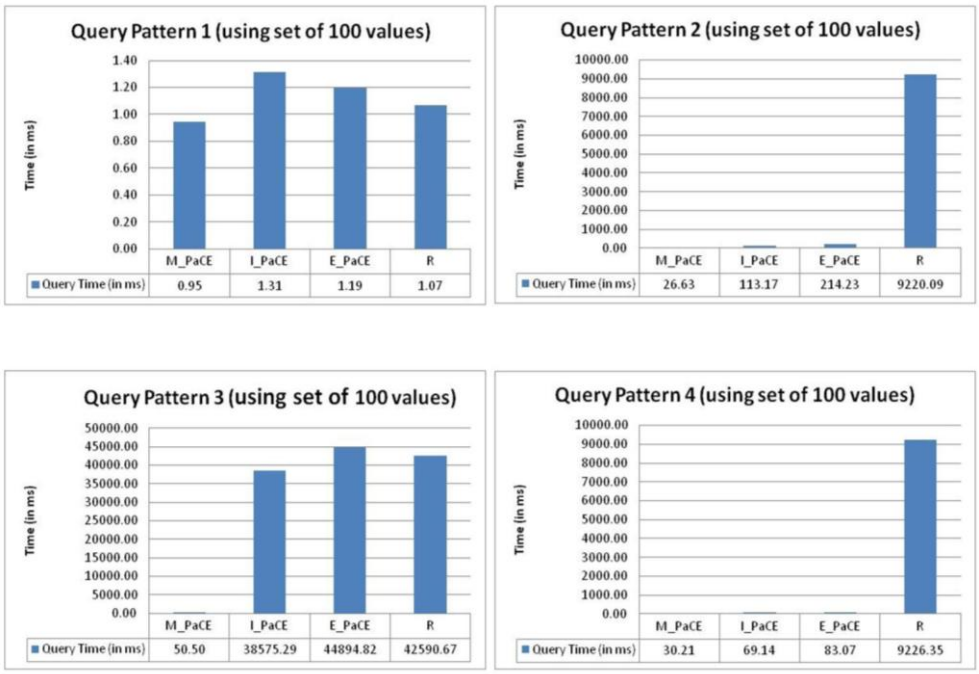

Figure 8: Query performance for query patterns using a set of 100 values

The results confirm the trend seen in the first phase of evaluation, with the added observation that for *Query Pattern 3* the difference between E_PaCE and R has decreased (R no longer outperforms E_PaCE significantly). In contrast, for the complex *Query Pattern 4*, the query performance for E_PaCE has further improved and is more than three orders of magnitude better than for R. The second phase of evaluation also confirms that in a real-world scenario the query performance of PaCE is comparable to reification for simple provenance queries and significantly better for complex provenance queries.

In the next section, we evaluate the query performance for an analytical query in the BKR project that uses provenance information for identify the publication pattern of journal articles on a specific topic of interest.

5.3 Application to Time Profiling of Scientific Results

An important objective for many applications and funding agencies is to understand the trend in research focused on a specific topic in biomedicine over a period of time. We extend the *Query Pattern 4* discussed in the previous section to define a query that collates the number of journal articles published over a period of 10 years (i.e., the span of the current BKR). As an example, we focus on mentions of the therapeutic use of the drug Thalidomide over time. This query translates to a complex SPARQL query that uses functions to aggregate number of publications per year. Figure 9 (a) shows a histogram created directly from the query results. The query performance is similar to what was observed for *Query Pattern 4*, that is, E_PaCE is three orders of magnitude faster than R (Figure 9(b)). This example demonstrates the feasibility of using RDF and SPARQL for representing and exploiting provenance information in large triple stores serving real-world applications.

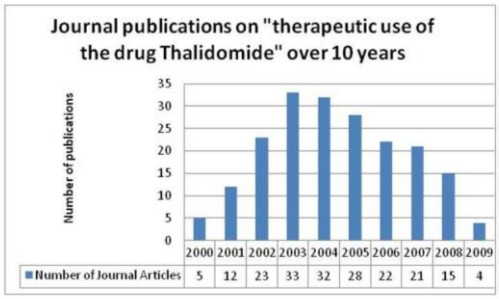

Figure 9 (a)

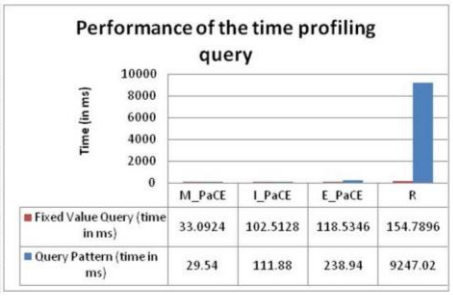

Figure 9 (b)

6 Conclusion

We show that that challenge of provenance tracking in RDF datasets can be effectively and efficiently addressed by using the PaCE approach in place of the RDF reification vocabulary. The PaCE approach addresses many of the issues associated with RDF reification, including lack of formal semantics, use of blank nodes, and

application-dependent interpretation of RDF documents. The PaCE approach uses the formal objects called provenance contexts that are defined in terms of the provenir upper-level provenance ontology to create provenance-aware RDF triple entities of S, P, and O. The model-theoretic semantics of PaCE is defined through a simple extension of the existing RDFS formal semantics. We implemented the PaCE approach in the BKR project. The evaluations demonstrate that using the PaCE approach to create provenance-specific RDF triples not only reduces the number of triples by at least 49% but also improves the performance of complex provenance queries by three orders of magnitude. We plan to extend BKR with data from additional sources and use the PaCE approach for provenance tracking.

Acknowledgments. This research was supported in part by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM) and the NIH RO1 Grant# 1R01HL087795-01A1. The authors would like to thank Tom Rindflesch and Marcelo Fiszman for providing a corpus of triples extracted from MEDLINE. Our thanks also go to Genaro Hernandez and Ramez Ghazzaoui for helping transform these triples into RDF. The open source version of the Virtuoso triple store is made available by OpenLink Software.

References

1. Protein knowledgebase: Uniprot. http://www.uniprot.org/, Retrieved Jan 10 2010
2. Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., and Hattori, M.: The KEGG resources for deciphering the genome. Nucleic Acids Res. **32** (2004) D277-D280
3. Vastrik, I., D'Eustachio, P., Schmidt, E., Joshi-Tope, G., Gopinath, G., Croft, D., de Bono, B., Gillespie, M., Jassal, B., Lewis, S., Matthews, L., Wu, G., Birney, E., Stein, L.: Reactome: a knowledge base of biologic pathways and processes. Genome Biology **8** (2007)
4. Entrez. http://www.ncbi.nlm.nih.gov/Database/, Retrieved Jan 10 2010
5. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res. **32** (2004) 267-270
6. Bodenreider, O., Rindflesch, T.C.: Advanced library services: Developing a biomedical knowledge repository to support advanced information management applications. Lister Hill National Center for Biomedical Communications, National Library of Medicine, Bethesda, Maryland (2006)
7. RDB2RDF XG Final Report. (2009)
8. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems **Special Issue on Linked Data** (2009)
9. MedlinePlus. Vol. 2010. National Library of Medicine (US), Bethesda (MD)
10. Manola, F., Miller, E.(Eds.): RDF Primer. W3C Recommendation (2004)
11. Hayes, P.: RDF Semantics. W3C Recommendation (2004)
12. ter Horst, H.J.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Journal of Web Semantics **3** (2005) 79-115
13. Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, provenance and trust. 14th International WWW Conference. ACM, New York, NY, Chiba, Japan (2005) 613-622
14. Ding, L., Finin, T., Joshi, A., Peng, J., Pinheiro da Silva, P., McGuinness, D.L.: Tracking RDF Graph Provenance using RDF Molecules. Proceedings of the 4th International Semantic Web Conference (2005)

15. Flouris, G., Fundulaki, I., Pediaditis, P., Theoharis, Y., Christophides, V.: Coloring RDF Triples to Capture Provenance. International Semantic Web Conference 2009, Washington D.C., USA (2009) 196-212
16. Sahoo, S.S., Barga, R.S., Goldstein, J., Sheth, A.P., Thirunarayan, K.: "Where did you come from...Where did you go?" An Algebra and RDF Query Engine for Provenance Kno.e.sis Center, Wright State University (2009)
17. Guha, R.V.: Contexts: A Formalization and Some Applications PhD Thesis, Stanford University (1991)
18. Guha, R.V., McCarthy, J.: Varieties of Contexts. CONTEXT 2003 (2003) 164–177
19. McCarthy, J.: Generality in artificial intelligence. Formalizing Common Sense: Papers by John McCarthy (1990) 226–236
20. Nayak, P.P.: Representing multiple theories. In: B. Hayes-Roth, R.K. (ed.): AAAI-94, Menlo Park, CA, USA (1994) 1154–1160
21. Buvaˇc, S., Mason, I.: Propositional logic of context. AAAI (1993) 412–419
22. Giunchiglia, F., Ghidini, C.: Local models semantics, or contextual reasoning = locality+compatibility. In: Anthony G. Cohn, L.S., Stuart C. Shapiro (ed.): KR'98: Principles of Knowledge Representation and Reasoning. San Francisco, CA, USA (1998) 282-289
23. Guha, R., McCool, R., Fikes, R.: Contexts for the Semantic Web In: Sheila A. McIlraith, D.P., Frank van Harmelen (ed.): International Semantic Web Conference, Vol. 3298. Springer, Hiroshima, Japan (2004) 32-46
24. Sahoo, S.S., Sheth, A. : Provenir ontology: Towards a Framework for eScience Provenance Management. Microsoft eScience Workshop. Pittsburgh, USA (2009)
25. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S.: OWL 2 Web Ontology Language Primer. W3C Recommendation (2009)
26. Grenon P, S.B., Goldberg L.: Biodynamic ontology: applying BFO in the biomedical domain. Stud Health Technol Inform. **102** (2004) 20-38
27. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening Ontologies with DOLCE. In: A. Gómez-Pérez, V.R.B. (ed.): 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. Springer Verlag, Siguenza, Spain (2002) 166-181
28. Niles, I., Pease, A. : Towards a Standard Upper Ontology. In: Welty, C., Smith, B. (ed.): 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Ogunquit, Maine (2001)
29. Beißwanger, E., Schulz, S., Stenzhorn, H., Hahn, U.: BioTop: An Upper Domain Ontology for the Life Sciences - A Description of its Current Structure, Contents, and Interfaces to OBO Ontologies. Applied Ontology **3** (2008) 205-212
30. Rector, A.L., Stevens, A., Rogers J.: Simple Bio Upper Ontology. (2006)
31. Ashburner, M., Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet. **25** (2000) 25-29
32. Sahoo, S.S., Weatherly, D.B., Muttharaju, R., Anantharam, P., Sheth, A., Tarleton, R.L.: Ontology-driven Provenance Management in eScience: An Application in Parasite Research. The 8th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 09). Springer Verlag, Portugal (2009) 992-1009
33. Smith, B., Ceusters, W., Klagges, B., Kohler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A.L., Rosse, C.: Relations in biomedical ontologies. Genome Biol **6** (2005) R46
34. Unified Medical Language System (UMLS).
35. Ayers, A., Völkel, M.: Cool URIs for the Semantic Web. Woking Draft. W3C (2008)
36. Prud'hommeaux, E., Seaborne, A. SPARQL Query Language for RDF. W3C Recommendation (2008)