LA-UR- 07-0665

Approved for public release; distribution is unlimited.

Title:	A Practical Ontology for the Large-Scale Modeling of Scholarly Artifacts and their Usage
Author(s):	Marko A. Rodriguez Johan Bollen Herbert Van de Sompel
Intended for:	IEEE/ACM Joint Conference on Digital Libraries June, 2007 Vancouver, Canada

A Practical Ontology for the Large-Scale Modeling of Scholarly Artifacts and their Usage

Marko A. Rodriguez
Digital Library Research & Prototyping Team
Los Alamos National Laboratory
Los Alamos, NM 87545
marko@lanl.gov

Johan Bollen
Digital Library Research & Prototyping Team
Los Alamos National Laboratory
Los Alamos, NM 87545
jbollen@lanl.gov

Herbert Van de Sompel
Digital Library Research & Prototyping Team
Los Alamos National Laboratory
Los Alamos, NM 87545
herbertv@lanl.gov

ABSTRACT

The large-scale analysis of scholarly artifact usage is constrained primarily by current practices in usage data archiving, privacy issues concerned with the dissemination of usage data, and the lack of a practical ontology for modeling the usage domain. As a remedy to the third constraint, this article presents a scholarly ontology that was engineered to represent those classes for which large-scale bibliographic and usage data exists, supports usage research, and whose instantiation is scalable to the order of 50 million articles along with their associated artifacts (e.g. authors and journals) and an accompanying 1 billion usage events. The real world instantiation of the presented abstract ontology is a semantic network model of the scholarly community which lends the scholarly process to statistical analysis and computational support. We present the ontology, discuss its instantiation, and provide some example inference rules for calculating various scholarly artifact metrics.

Categories and Subject Descriptors

I.2.4 [**Knowledge Representation Formalisms and Methods**]: Semantic Networks; H.3.7 [**Digital Libraries**]: Standards—*ontologies*

General Terms

Ontologies, Scholarly Communication

Keywords

Resource Description Framework and Schema, Web Ontology Language, Semantic Networks

1. INTRODUCTION

New publications are added to the scholarly record at an accelerating pace. This point is realized by observing the evolution of the amount of publications indexed in Thomson Scientific's citation database over the last fifteen years: 875,310 in 1990; 1,067,292 in 1995; 1,164,015 in 2000, and 1,511,067 in 2005. However, the extent of the scholarly record reaches far beyond what is indexed by Thompson Scientific. While Thompson Scientific focuses primarily on quality-driven journals (roughly 8,700 in 2005), they do not index more novel scholarly artifacts such as preprints deposited in institutional or discipline-oriented repositories, datasets, software, and simulations that are increasingly being considered scholarly communication units in their own right.

While the size (and growth) of the scholarly record is impressive, the extent of its use is even more staggering. For instance, in November 2006, Elsevier's Science Direct, which provides access to articles from approximately 2,000 journals, celebrated its 1 billionth full-text download since counting started in April of 1999[1]. And, again, the extent of scholarly usage clearly reaches far beyond Elsevier's repository. Furthermore, usage events include not only full-text downloads, but also events such as requesting services from linking servers, downloading bibliographic citations, emailing abstracts, etc.

To a large extent, the effect of usage behavior on the scholarly process is a horizon that is only beginning to be understood and, if properly studied, will offer clues to the evolutionary trends of science [12, 8, 4], quantitative models of the value of scholarly artifacts [7, 5], and services to support scholars [6]. The Andrew W. Mellon funded MESUR[2] project at the Research Library of the Los Alamos National Laboratory aims at developing metrics for assessing scholarly communication artifacts (e.g. articles, journals, conference proceedings, etc.) and agents (e.g. authors, institutions, publishers, repositories, etc.) on the basis of scholarly usage. In order to do this, the MESUR project makes use of a representative collection of bibliographic, citation and usage data. This data is collected from a wide variety of sources including academic publishers, secondary publishers, institutional linking servers, etc. Expectations are that the collected data will eventually encompass tens of millions of bibliographic records, hundreds of millions of citations,

[1] Elsevier's 1 billion downloads article available at: http://www.info.sciencedirect.com/news/archive/2006/news_billionth.asp

[2] MEtrics from Scholarly Usage of Resources available at: http://www.mesur.org/

and billions of usage events. Mining such a vast data set in an efficient, performing, and flexible manner presents significant challenges regarding data representation and data access. This article presents, the OWL ontology [17] used by MESUR to represent bibliographic, citation and usage data in an integrated manner. The proposed MESUR ontology is practical, as opposed to all encompassing, in that it represents those artifacts and properties that, as previously shown in [6], are realistically available from modern scholarly information systems. This includes bibliographic data such as author, title, identifier, publication date and usage data such as the IP address of the accessing agent, the date and time of access, type of usage, etc. Finally, another novel contribution of this work is the hybrid storage and access architecture in which relational database and triple store technology are combined. This is achieved by storing core data and relationships in the triple store and auxiliary data in a relational database. This design choice is driven by the need to keep the size of the triple store to a level that can realistically be handled by current technologies. The combination of the data architecture and scholarly ontology presented in this article provide the foundation for the large-scale modeling and analysis of scholarly artifacts and their usage.

2. SEMANTIC NETWORK ONTOLOGIES

A semantic network (sometimes called a multi-relational network or multi-graph) is composed of a set of nodes (representing heterogeneous artifacts) connected to one another by a set of qualified, or labeled, edges [25]. In a graph theoretic sense, a semantic network is a directed labeled graph. Because an edge is labeled, two nodes can be connected to one another by an infinite number of edges. However, in most cases, the possible interconnections between node types is constrained to a predetermined set. This predetermined set is made explicit in the semantic network's associated ontology. An ontology is generally defined as a set of abstract classes, their relationship to one another, and a collection of inference rules for deriving implicit relationships [1]. An ontology makes no explicit reference to the actual instances of the defined abstract classes; this is the role of the semantic network.

An ontology is related to the developer's API in object oriented programming languages such as C++ and Java (minus the explicit representation of class methods/functions). For example, the set of relationships of an ontological class are known as the class' properties and, in the object oriented lexicon, can be understood as class fields. Also, a taxonomy is usually expressed in a semantic network ontology. A taxonomy of sub- and super-classes support the inheritance of class properties. For instance, if all mammals are warm blooded, then all humans are warm blooded because all humans are mammals. In an inheritance hierarchy, the warm blooded property of mammals is inherited by all sub-classes of mammal (e.g. human).

Figure 1 diagrams the relationship between an ontology and its semantic network instantiation. The circles represents objects that are instances of the dash-dot pointed to abstract classes (the squares). The three lower squares are subclasses of a more general top-level class (denoted by the dashed edges). The horizontal edges in the ontology denote permissible property types in the instantiation and thus, corresponding horizontal labeled edges in the semantic network may exist. Figure 1 does not expose the range of conceptual nuances that can be expressed by modern ontology languages and thus, only provides a rudimentary representation of the relationship between an ontology and its semantic network instantiation.

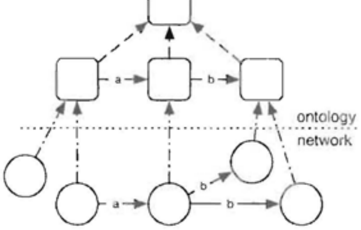

Figure 1: The relationship between an ontology and its semantic network instantiation

2.1 Semantic Network Technology

The most popular semantic network representational framework is the Resource Description Framework and Schema, or RDF(S) [16]. RDF(S) represents all nodes and edges by Universal Resource Identifiers (URI) [9]. The URI approach supports the use of namespacing such that the URI `http://www.science.org#Article` has a different meaning, or connotation, than what may be understood by the URI `http://www.newspaper.net#Article`.

The Web Ontology Language (OWL) is an extension of RDF(S) that supports a richer vocabulary (e.g. promotes many set theoretical concepts) [17]. Protégé[3] is perhaps the most popular application for designing OWL ontologies [18]. While OWL is primarily a machine readable language, an OWL ontology can be diagrammed using the Unified Modeling Language's (UML) class diagrams (i.e. entity relationship diagrams).

Modern semantic network data stores represent the relationship between two nodes by a *triple*. For instance, the triple

$\langle \mathrm{URI}_a$, `http://xmlns.com/foaf/0.1/#knows`, $\mathrm{URI}_b \rangle$

states that the resource identified by URI_a knows the resource identified by URI_b, where URI_a and URI_b are nodes and `http://xmlns.com/foaf/0.1/#knows` is a directed labeled edge (see Figure 2). The meaning of `knows` is fully defined by the URI `http://xmlns.com/foaf/0.1/`. The union of instantiated FOAF triples is a FOAF semantic network. Current platforms for storing and querying such semantic networks are called *triple stores*. Many open source and proprietary triple stores currently exist. Various querying languages exist as well [15]. The role of the query language is to provide the interface to access the data contained in the triple store. This is analogous to the relationships

URI_a
http://xmlns.com/foaf/0.1/#knows
URI_b

Figure 2: A diagrammed triple

[3] Protégé available at: http://protege.stanford.edu/

between SQL and relational databases. Perhaps the most popular triple store query language is SPARQL [20]. An example SPARQL query is

```
PREFIX   foaf:  <http://xmlns.com/foaf/0.1/#>
PREFIX   vub:  <http://homepages.vub.ac.be/#>
SELECT   ?x
WHERE    ( ?x foaf:knows vub:cgershen ).
```

In the above query, the ?x variable is bound to any node that is the domain of a triple with an associated predicate of http://xmlns.com/foaf/0.1/#knows and a range of vub:cgershen. Thus, the above query returns all people who know vub:cgershen (i.e. Carlos Gershenson).

The ontology plays a significant role in many aspects of a semantic network. Figure 3 demonstrates the role of the ontology in determining which real world data is harvested, how that data is represented inside of the triple store (semantic network), and finally, what queries and inferences are possible to execute.

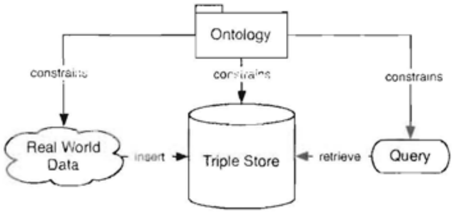

Figure 3: The many roles of an ontology

3. SCHOLARLY ONTOLOGIES

In general, an ontology's classes, their relationships, and inferences are determined according to what is being modeled, for what problems that model is trying to solve, and how that model's classes can be instantiated according to real world data. Thus, there were three primary requirements to the development of the MESUR ontology:

1. realistically available real world data
2. ability to study usage behavior
3. scalability of the triple store instantiation.

Without real-world data, an ontology serves only as a conceptual tool for understanding a particular domain and, in such cases, ontologies of this nature may be very detailed in what they represent. However, for ontologies that are designed to be instantiated by real world data, the ontology is ultimately constrained by data availability. Thus, the MESUR ontology is constrained to bibliographic and usage data since these are the primary sources of scholarly data. In the scholarly community, while articles, journals, conference proceedings, and the like are well documented and represented in formats that lend themselves to analysis, other information, such as usage data, tends to be less explicit due to the inherent privacy issues surrounding individual usage behavior. Therefore, a primary objective of the MESUR project is the acquisition of large-scale usage data sets from providers world-wide.

The purpose of the MESUR project is to study usage behavior in the scholarly process and therefore, usage modeling is a necessary component of the MESUR ontology. Given both usage and bibliographic data, it will be possible to generate and validate metrics for understanding the 'value' of all types of scholarly artifacts. Currently, the scholarly community has one primary means of understanding the value of a journal and thus its authors: the ISI Impact Factor [10]. With a semantic network data structure that includes not only article (and thus, journal) citation, but also authorship, usage, and institutional relationships, new metrics that not only rank journals, but also conferences, authors, and institutions will be created and validated.

Finally, the proposed ontology was engineered to handle an extremely large semantic network instantiation (on the order of 50 million articles with a corresponding 1 billion usage events). The MESUR ontology was engineered to make a distinction between required base-relationships and those, that if needed, can be inferred from the base-relations. Futhermore, due to the fact that the MESUR ontology was developed to support the large-scale analysis of usage, many of the metadata properties such as article title or author name are not explicitly represented in the ontology and thus, as will be demonstrated, such data can be accessed outside the triple store by reference to a relational database.

4. RELATED WORK

Other efforts have produced and exploited scholarly ontologies, but they do not cover the needs of the MESUR project for two primary reasons. First, they generally lack the integration of publication, citation and usage data, which MESUR requires in order to represent and analyze these crucial stages of the public scholarly communication process. Second, scalability appears to not have been a major concern when designing the ontologies and thus, instantiating them at the order of what MESUR will be representing is unfeasible. Sometimes, the ontology is too elaborate, adding complexity that rarely pays off for the simple reason that it is hard to realistically come by data to populate defined properties (e.g. detailed author or affiliation information). Other times, the ontology requires the storage of information that cannot realistically be represented for vast data collections using current triple store technologies.

Several scholarly ontologies are available in the DAML Ontology Library[4]. While they focus on bibliographic constructs, they do not model usage events. The same is true of the Semantic Community Web Portal ontology [26], which, in addition maintains many detailed classes whose instantiation is unrealistic given what is recorded by modern scholarly information systems.

The ScholOnto ontology was developed as part of an effort aimed at enabling researchers to describe and debate, via a semantic network, the contributions of a document, and its relationship to the literature [24]. While this ontology supports the concept of a scholarly document and a scholarly agent, it focuses on formally summarizing and interactively debating claims made in documents, not on expressing the actual use of documents. Moreover, support for bibliographic data is minimal whereas support for discourse constructs, not required for MESUR, is very detailed.

The ABC ontology [13] was primarily engineered as a com-

[4] DAML Ontology Library available at: http://www.daml.org/ontologies/

mon conceptual model for the interoperability of a variety of metadata ontologies from different domains. Although the ABC ontology is able to represent bibliographic and usage concepts by means of constructs such as artifact (e.g. article), agent (e.g. author), and action (e.g. use), it is designed at a level of generality that does not directly support the granularity required by the MESUR project.

An interesting ontology-based approach was developed by the Ingenta MetaStore project [19]. Unfortunately, again, the Ingenta ontology does not support expressing usage of scholarly documents, which is a primary concern in MESUR. Nevertheless, the approach is inspiring because Ingenta faces significant challenges regarding scalability of the ontology-based representation, storage and access of their bibliographic metadata collection, which covers approximately 17 million journal articles. However, the scale of the MESUR data set is several orders of magnitude larger, calling for optimizations wherever possible. For example, given the MESUR project's focus on usage, storing bibliographic properties (author names, abstract, titles, etc.) in the triple store, as done by Ingenta, is not essential. As a result, in order to improve triple store query efficiency, MESUR stores such data in a relational database, and the MESUR ontology does not explicitly represent these literals.

The principles espoused by the OntologyX[5] ontology are inspiring. OntologyX uses *context* classes as the "glue" for relating other classes, an approach that was adopted for the MESUR ontology. For instance, the MESUR ontology does not have a direct relationship between an article and its publishing journal. Instead, there exists a publishing context that serves as an N-ary operator uniting a journal, the article, its publication date, its authors, and auxiliary information such as the source of the bibliographic data. The context construct is intuitive and allows for future extensions to the ontology. OntologyX also helped to determine the primary abstract classes for the MESUR ontology. Unfortunately, OntologyX is a proprietary ontology for which very limited public information is available, making direct adoption unfeasible for MESUR. As a matter of fact, all inspiration was derived from a single PowerPoint presentation from the 2005 FBRB Workshop [22].

Finally, in the realm of usage data representation, no ontology-based efforts were found. Nevertheless, the following existing schema-driven approaches were explored and served as inspiration: the OpenURL ContextObject approach to facilitate OAI-PMH-based harvesting of scholarly usage events [6], the XML Log standard to represent digital library logs [11], and the COUNTER schema to express journal level usage statistics [23].

5. LEVERAGING RELATIONAL DATABASE TECHNOLOGY

The MESUR project makes use of a triple store to represent and access its collected data. While the triple store is still a maturing technology, it provides many advantages over the relational database model. For one, the network-based representation supports the use of network analysis algorithms. For the purposes of the MESUR project, a network-based approach to data analysis will play a major role in quantifying the value of the scholarly artifacts contained within it. Other benefits that are found with triple store technologies that are not easily reproducible within the relational database framework include ease of schema extension and ontological inferencing.

A novel contribution of the presented ontology is its solution to the problem of scalability found in modern triple store technologies [14]. While semantic networks provide a flexible medium for representing and searching knowledge, current triple store applications do not support the amount of data that can be represented at the upper limit of what is possible with modern relational database technologies. Therefore, it was necessary to be selective of what information is actually modeled by the MESUR ontology. For the MESUR project, much of the data associated with each scholarly artifact is maintained outside the triple store in a relational database.

The typical bibliographic record contains, for example, an article's identifiers (e.g. DOI, SICI, etc.), authors, title, journal/conference/book, volume, issue, number, and page numbers. Typical usage information contains, for example, the users identifier (e.g. IP address), the time of the usage event, and a session identifier. An example of the various bibliographic and usage properties are outlined in the Table 1 and Table 2, respectively. Note that the connection between the bibliographic record and the usage event occurs through the doc_id (bolded properties). The doc_id is a internally generated identifier created during the MESUR project's ingestion process.

property	**value**
title	The Convergence of Digital Libraries ...
author(s)	Rodriguez, Bollen, Van de Sompel
collection	Journal of Information Science
publisher	Sage Publications
date	2006
start page	149
end page	159
volume	32
issue	2
doi	10.1177/0165551506062327
doc_id	**b5e1ab73-26b5-41f0-a83f-b47b4d737**

Table 1: Example bibliographic properties

property	value
event_id	45563ac2-c7d4-4669-ab9c-ac5129535ee5
time	2006-09-27 00:00:03
agent	4AD2FD457EB59CE08AAAF6EA2A63F
session	C3044206
affiliation	California State University, Los Angeles
doc_id	**b5e1ab73-26b5-41f0-a83f-b47b4d737**

Table 2: Example usage properites

The two tables demonstrate how bibliographic and usage data can be easily represented in a relational database. From the relational database representation, a RDF N-Triple[6] data file can be generated. One such solution for this relational database to triple store mapping is the D2R mapper [2]. However, note that not all data in the relational database is exported to this intermediate format. Instead, only those properties that promote triple store scalability and usage research were included. Thus, article titles, journal issues

[5] OntologyX available at: http://www.ontologyx.com/

[6] N-Triple available at:
http://www.w3.org/2001/sw/RDFCore/ntriples/

and volumes, names of authors, to name a few, are not explicitly represented within the triple store and thus, are not modeled by the ontology. If a particular artifact property that is not in the ontology is required for a computation, the computing algorithm references the relational database holding the complete representation the acquired data. For example, bi-directional resolution of the artifact with doc_id 2 is depicted in Figure 4 where the resolving identifier is specific to the artifact (for the sake of diagram readability, assume that 2 is b5e1ab73-26b5-41f0-a83f-b47b4d737 from Table 1 and 2). This model is counter to what is seen in other scholarly ontologies such as the Ingenta ontology [19]. This design choice was a major factor that prompted the engineering of a new ontology for bibliographic and usage modeling.

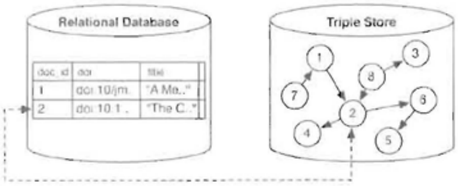

Figure 4: The relationship between the relational database and the triple store

6. THE MESUR ONTOLOGY

The MESUR ontology is currently at version 2007-01 at `http://www.mesur.org/schemas/2007-01/mesur` (abbreviated `mesur`). Full HTML documentation of the ontology can be found at the namespace URI. The following sections will describe how bibliographic and usage data is modeled to meet the requirements of understanding large-scale usage behavior, while at the same time promoting scalability.

6.1 The Primary Classes

The most general class in OWL is `owl:Thing`. The MESUR ontology provides three subclasses of `owl:Thing`. These MESUR classes are `mesur:Agent`, `mesur:Document`, and `mesur:Context`[7]. This is represented in Figure 5 where an edge denotes a `rdfs:subClassOf` relationship.

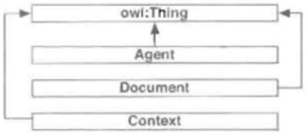

Figure 5: The primary classes of the MESUR ontology

The `Context` classes serve as the "glue" by which `Agents` and `Documents` interact. A `Context` is analogous to `rdf:Bag` in that it is an N-ary operator unifying the literals and objects pointed to by its respective properties. All relationships between `Agents` and `Documents` occurs through a particular `Context`. However, as will be demonstrated, direct relationships can be inferred. All inferred properties are denoted by the "(i)" notation in the following UML class diagrams. All inferred properties are superfluous relationships since there is no loss of information by excluding their instantiation (the information is contained in other relationships). The algorithms for inferring them will be discussed in their respective `Context` subsection.

Currently, all the MESUR classes are specifications or generalizations of other classes. No holonymy/meronymy (composite) class definitions are used at this stage of the ontology's development. Figure 6 presents the complete taxonomy of the MESUR ontology. This diagram primarily serves as a reference. Each class will be discussed in the following sections.

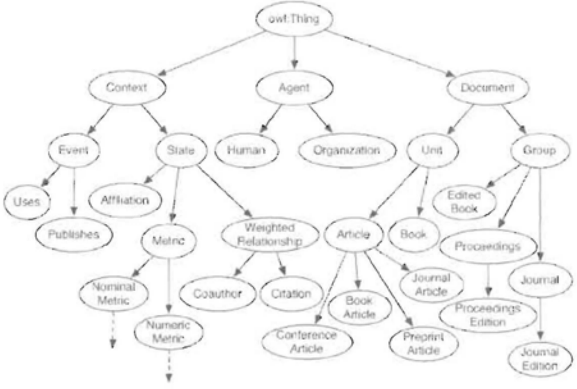

Figure 6: MESUR taxonomy

6.2 The Agent Classes

The `Agent` taxonomy is diagrammed in Figure 7. An `Agent` can either be a `Human` or an `Organization`. A `Human` is an actual individual whether that individual can be uniquely identified (e.g. an document author) or not (e.g. a document user). The `authored` property is an inferred relationship and denotes that an `Agent` authored a particular `Document` and the `published` property denotes that an `Agent` has published a `Document`. The `authored` and `published` property can be inferred by information within the `Publishes` context discussed later. Similarly, the `used` property denotes that an `Agent` has used a particular `Document`. The `used` property can be inferred from the `Uses` context.

An `Organization` is a class that is used for both bibliographic and usage provenance purposes. Given that bibliographic and usage data, at the large-scale, must be harvested from multiple institutions, it is necessary to make a distinction between the various data providers. In many cases, an `Organization` can be both a bibliographic (e.g. a publisher) and a usage (e.g. a repository) provider. Furthermore, an `Organization` can also be an author's academic institution (e.g. a university).

Finally, all `Agents` can have any number of affiliations. For an `Organization`, this is a recursive definition which allows an Organization to have many affiliate `Organizations` while at the same time allowing for the `Human` leaf nodes of an `Organization` to be represented by the same construct.

[7] For the remainder of this article, all classes that are not explicitly namespaced are from the `mesur` namespace.

The rules governing the inference of the hasAffiliation and hasAffiliate properties are discussed in the section describing the Affiliation context.

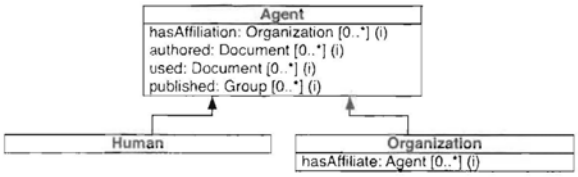
Agent
hasAffiliation: Organization [0..*] (i)
authored: Document [0..*] (i)
used: Document [0..*] (i)
published: Group [0..*] (i)
Human
Organization
hasAffiliate: Agent [0..*] (i)

Figure 7: Classes of Agent and their properties

6.3 The Document Classes

A Document is an abstract concept of a particular scholarly product such as those depicted in Figure 8.

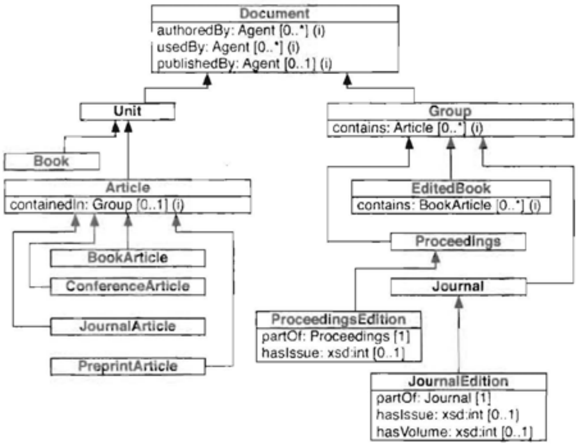

Figure 8: Classes of Document and their properties

In general, Document objects are those artifacts that are written, used, and published by Agents. Thus, a Document can be a specific article, a book, or some grouping such as a Journal, conference Proceedings, or an EditedBook. There are two Document subclasses to denote whether the Document is a collection (Group) or an individually written work (Unit). A Journal and Proceedings is an abstract concept of a collection of volumes/issues. An edition to a proceedings or journal is associated with its abstract Group by the partOf property. The authoredBy, containedIn, publishedBy, and contains properties can be inferred from the Publishes context. Also, the usedBy property can be inferred from the Uses context.

6.4 The Context Classes

As previously stated, all properties from the Agent and Document classes that are marked by the "(i)" notation are inferred properties. These properties can be automatically generated by inference algorithms and thus, are not required for insertion into the triple store. What this means is that inherent in the triple store is the data necessary to infer such relationships. Depending on the time (e.g. query complexity) and space (e.g. disk space allocation) constraints, the inclusion of these inferred properties is determined. At any time, these properties can be inserted or removed from the triple store. The various inferred properties are determined from their respective Context objects. Therefore, the MESUR owl:ObjectProperty taxonomy provides two types of object properties: ContextProperty and InferredProperty (see Figure 9).

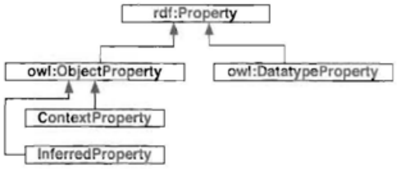

Figure 9: The abstract MESUR property classes

A Context class is an N-ary operator much like an rdf:Bag. Current triple store technology expresses tertiary relationships. That means that only three resources are related by a semantic network edge (i.e. a subject URI, predicate URI, and object URI). However, many real-world relationships are the product of multiple interacting objects. It is the role of the various Context classes to provide relationships for more than three URIs. The Context classes are represented in Figure 10.

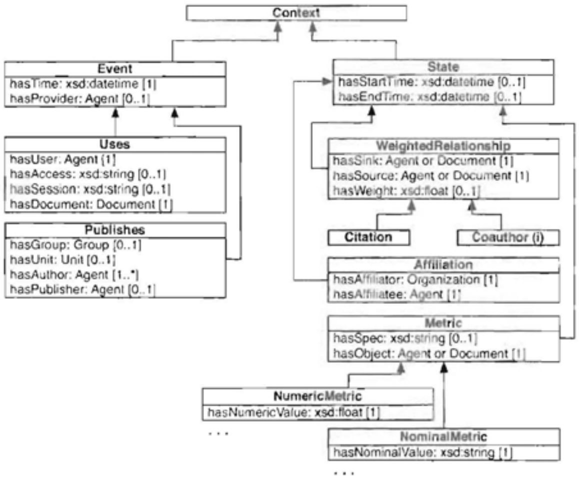
Context
Event
hasTime: xsd:datetime [1]
hasProvider: Agent [0..1]
State
hasStartTime: xsd:datetime [0..1]
hasEndTime: xsd:datetime [0..1]
Uses
hasUser: Agent [1]
hasAccess: xsd:string [0..1]
hasSession: xsd:string [0..1]
hasDocument: Document [1]
WeightedRelationship
hasSink: Agent or Document [1]
hasSource: Agent or Document [1]
hasWeight: xsd:float [0..1]
Publishes
hasGroup: Group [0..1]
hasUnit: Unit [0..1]
hasAuthor: Agent [1..*]
hasPublisher: Agent [0..1]
Citation
Coauthor (i)
Affiliation
hasAffiliator: Organization [1]
hasAffiliatee: Agent [1]
Metric
hasSpec: xsd:string [0..1]
hasObject: Agent or Document [1]
NumericMetric
hasNumericValue: xsd:float [1]
. . .
NominalMetric
hasNominalValue: xsd:string [1]
. . .

Figure 10: Classes of Context and their properties

The Context class has two subclasses: Event and State. An Event is some measurement done by some provider at a particular point in time. For example, the Publishes and Uses events are recorded by publisher and repositories at some point in time. As a side note, the hasProvider property of the Event class is an efficient model for the representation of provenance constructs. Instead of reifying every statement with provenance data (e.g. triple x was supplied by provider y [19]), a single triple is provided for each Event (e.g. event x was supplied by provider y).

On the other side of the Context taxonomy are the State contexts. A State is some measurement that can, in some cases, occur over a span of time and are used to represent complex relationships between artifacts or as a way of attaching high-level properties (i.e. metadata) to an artifact. The next sections will provide a detailed description of each Context class along with SPAQRL queries for inferring all the aforementioned InferredProperty properties.

6.4.1 The Publishes Context

A Publishes event states, in words, that a particular bibliographic data provider has acknowledged that a set of authors have authored a unit that was published in a group by some publisher at a particular point in time. A Publishes object relates a single bibliographic data provider, Agent authors, a Unit, an Agent publisher, a Group, and a publication ISO-8601 date time literal[8]. Figure 11 represents a Publishes context and the inferable properties (dashed edges) of the various associated artifacts. All inferred properties have a respective inverse relationship. Note that both PreprintArticle and Book publishing are represented with OWL restrictions (i.e. they are not published in a Group). The details of these restrictions can be found in the actual ontology definition.

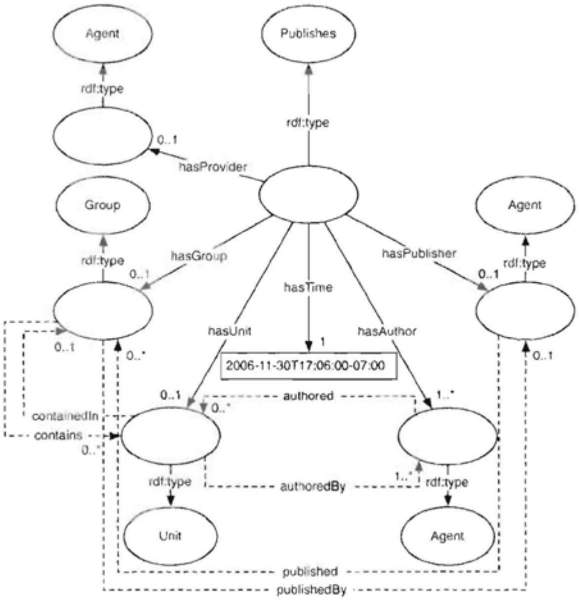

Figure 11: Example Publishes Context

The dashed edges in Figure 11 denote properties that are a rdfs:subClassOf the InferredProperty. For instance, the abstract triple ⟨Author, authors, Document⟩ is inferred given the results of the following SPARQL query, where for the sake of brevity, the PREFIX declarations are removed and the INSERT statement represents the insert of its triple argument into the triple store[9].

```
SELECT   ?a ?b
WHERE
          ( ?x  rdf:type  mesur:Publishes  )
          ( ?x  mesur:hasUnit  ?a  )
          ( ?x  mesur:hasAuthor  ?b  )

INSERT  <  ?a  mesur:authoredBy  ?b  >
INSERT  <  ?b  mesur:authored  ?a  >  .
```

To infer the Group property contains and Unit property containedIn, the following SPARQL query and INSERT statements suffice.

```
SELECT   ?a ?b
WHERE
          ( ?x  rdf:type  mesur:Publishes  )
          ( ?x  mesur:hasUnit  ?a  )
          ( ?x  mesur:hasGroup  ?b  )

INSERT  <  ?a  mesur:containedIn  ?b  >
INSERT  <  ?b  mesur:contains  ?a  >  .
```

Finally, the published and publishedBy properties are inferred by:

```
SELECT   ?a ?b
WHERE
          ( ?x  rdf:type  mesur:Publishes  )
          ( ?x  mesur:hasPublisher  ?a  )
          ( ?x  mesur:hasGroup  ?b  )

INSERT  <  ?a  mesur:published  ?b  >
INSERT  <  ?b  mesur:publishedBy  ?a  >  .
```

6.4.2 The Uses Context

The Uses context denotes a single usage event where an Agent uses a Document at a particular point in time. The Uses context is diagrammed in Figure 12. Like the Publishes context, the Uses context is an N-ary construct. Depending on the usage provider, a session identifier and access type is recorded. A session identifier denotes the user's login session. An access type denotes, for example, whether the used Document had its abstract viewed or was fully downloaded.

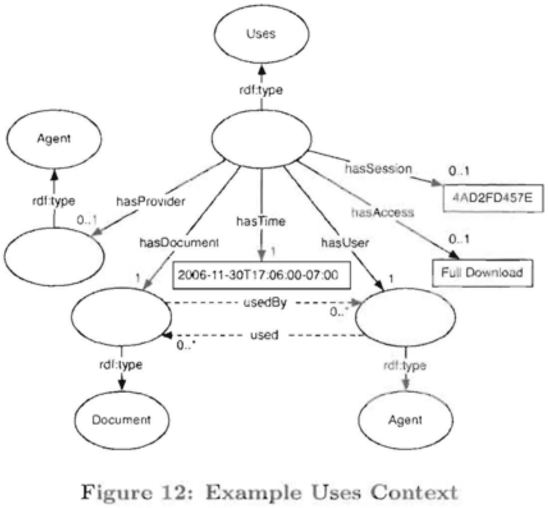

Figure 12: Example Uses Context

The following SPARQL query and INSERT statements represent the inference of the usedBy and used inverse properties of an Article document and Agent, respectively. Also, note the last two INSERT statements. These statements demonstrate how Group usage information can also be inferred.

[8] ISO-8601 available at: http://www.w3.org/TR/NOTE-datetime/

[9] Please note that all the presented SPARQL queries are not optimized for speed, but instead, are optimized for readability.

```
SELECT   ?a ?b ?c
WHERE
         ( ?x rdf:type mesur:Uses )
         ( ?x mesur:hasDocument ?a )
         ( ?a rdf:type mesur:Article )
         ( ?x mesur:hasUser ?b )
         ( ?y rdf:type mesur:Publishes )
         ( ?y mesur:hasUnit ?a )
         ( ?y mesur:hasGroup ?c )

INSERT < ?a mesur:usedBy ?b >
INSERT < ?b mesur:used ?a >
INSERT < ?c mesur:usedBy ?b >
INSERT < ?b mesur:used ?c > .
```

6.4.3 *The Weighted Relationship Context*

In many instances, one artifact is related to another by a particular semantic. However, in some instance, one artifact is related to another by a semantic label and a floating point weight value. Furthermore, that weighted relationship may have been recorded over some period of time. The `WeightedRelationship` state context is used to represent such relationships.

The `Citation` state context denotes a weighted citation and is a `rdfs:subClassOf` the `WeightedRelationship`. For `Unit` to `Unit` citation, the weight value is 1.0 (or no weight property to reduce the triple store footprint) and there are no start and end time points. However, for `Group` to `Group` citations, the weight of the `Citation` represents how many times a particular `Group` cites another over some period of time. Hence, it is necessary to denote the start and end points of both the source and the sink nodes. Figure 13 diagrams a `Citation` context. Furthermore, the sink and source types can be either an `Agent` or a `Document`, thus, `Organization` to `Organization` citations can be represented.

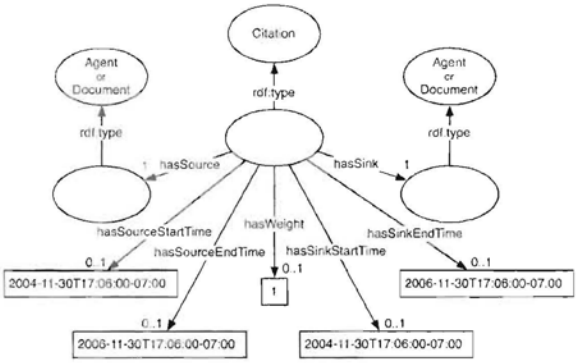

Figure 13: Example Citation Context

Given `Unit` to `Unit` citations, the `Citation` weight between any two `Groups` can be inferred. The following example SPARQL query generates the `Citation` object for citations from 2007 articles in the Journal of Informetrics (ISSN: 1751-1577) to 2005-2006 articles in Scientometrics (ISSN: 0138-9130). Assume that the URI of the journals are their ISSN numbers, the date time is represented as a year instead of the lengthy ISO-8601 representation, and the `COUNT` command is analogous to the SQL `COUNT` command (i.e. returns the number of elements returned by the variable binding).

```
SELECT   ?x
WHERE
         ( ?x rdf:type mesur:Citation )
         ( ?x mesur:hasSource ?a )
         ( ?x mesur:hasSink ?b )
         ( ?a rdf:type mesur:Article )
         ( ?b rdf:type mesur:Article )
         ( ?y rdf:type mesur:Publishes )
         ( ?z rdf:type mesur:Publishes )
         ( ?y mesur:hasTime ?t)
                  AND (?t > 2004 AND ?t < 2007)
         ( ?z mesur:hasTime ?u) AND ?u = 2007
         ( ?y mesur:hasUnit ?a )
         ( ?z mesur:hasUnit ?b )
         ( ?y mesur:hasGroup ?c )
         ( ?z mesur:hasGroup ?d )
         ( ?c mesur:partOf urn:issn:1751-1577 )
         ( ?d mesur:partOf urn:issn:0138-9130 )

INSERT < _123 rdf:type mesur:Citation >
INSERT < _123 mesur:hasSource urn:issn:1751-1577 >
INSERT < _123 mesur:hasSink urn:issn:0138-9130 >
INSERT < _123 mesur:hasWeight COUNT(?x) >
INSERT < _123 mesur.hasSourceStartTime 2007 >
INSERT < _123 mesur:hasSourceEndTime 2007 >
INSERT < _123 mesur.hasSinkStartTime 2005 >
INSERT < _123 mesur:hasSinkEndTime 2006 > .
```

Figure 14 diagrams the `Coauthor` weighted relationship context. The weight value of this relationship denotes the number of times two authors have coauthored together over a some period of time.

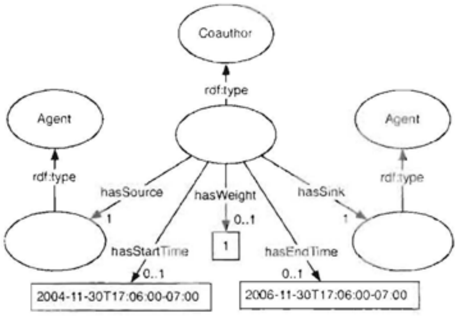

Figure 14: Example Coauthor Context

The following SPARQL query demonstrates how to infer the weighted `Coauthor` relationship between the authors Marko (`lanl:marko`) and Herbert (`lanl:herbertv`) over all time. A time period for coauthorship counting can be inserted in a fashion similar to the `Citation` example previous.

```
SELECT ?x
WHERE
         ( ?x rdf:type mesur:Publishes )
         ( ?x mesur:hasAuthor lanl:marko )
         ( ?x mesur:hasAuthor lanl:herbertv )

INSERT < _123 rdf:type mesur:Coauthor >
INSERT < _123 mesur:hasSource lanl:marko >
INSERT < _123 mesur:hasSink lanl:herbertv >
INSERT < _123 mesur:hasWeight COUNT(?x) >
INSERT < _456 rdf:type mesur:Coauthor >
INSERT < _456 mesur:hasSource lanl:herbertv >
INSERT < _456 mesur:hasSink lanl:marko >
INSERT < _456 mesur:hasWeight COUNT(?x) > .
```

6.4.4 *The Affiliation Context*

An `Affiliation` context denotes that a particular `Human` is affiliated with an `Organization` or that an `Organization` is affiliated with another Organization. An Affiliation can be represented as occurring over a particular period of time. An example of an Affiliation state context is diagrammed in Figure 15.

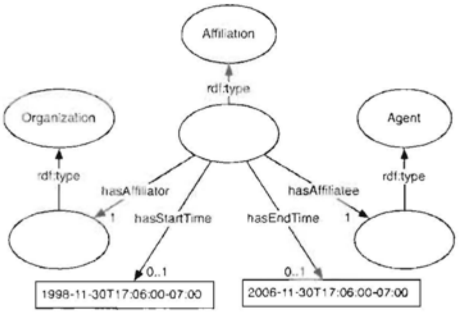

Figure 15: Example Affiliation Context

The hasAffiliate and hasAffiliation properties of the Agent classes can be inferred by the following SPARQL query.

```
SELECT   ?a ?b
WHERE
          ( ?x rdf:type mesur:Affiliation )
          ( ?x mesur:hasAffiliator ?a )
          ( ?x mesur:hasAffiliatee ?b )

INSERT < ?a mesur:hasAffiliate?b >
INSERT < ?b mesur:hasAffiliation?a > .
```

6.4.5 The Metric Context

The primary objective of the MESUR project is to study the relationship between usage-based value metrics (e.g. Usage Impact Factor [5]) and citation-based value metrics (e.g. ISI Impact Factor [10] and the Y-Factor [3]). The Metric context allows for the explicit representation of such metrics. The Metric context has both the NumericMetric and NominalMetric subclasses. Figure 16 diagrams the 2007 ImpactFactor numeric metric context for a Group. Note that the Context hierarchy in Figure 10 does not represent the set of Metrics explored by the MESUR project. This taxonomy will be presented in a future publication.

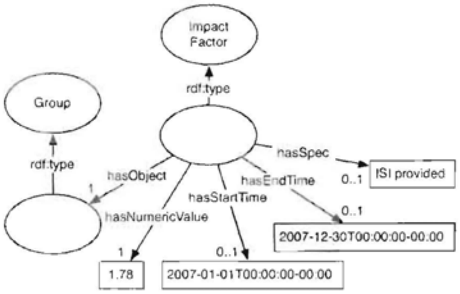

Figure 16: Example Impact Factor Context

The example SPARQL query and respective INSERT statements demonstrate how to calculate the 2007 Impact Factor for the Proceedings of the Joint Conference on Digital Libraries (JCDL ISSN: 1082-9873). The 2007 Impact Factor for the JCDL is defined as the number of citations from any Unit published in 2007 to articles in the JCDL proceedings published in either 2005 or 2006 normalized by the total number of articles published by JCDL in 2005 and 2006 [10].

```
SELECT   ?x
WHERE
          ( ?x rdf:type mesur:Publishes )
          ( ?x mesur:hasUnit ?a )
          ( ?x mesur:hasGroup ?b )
          ( ?b mesur:partOf urn:issn:1082-9873 )
          ( ?x mesur:hasTime ?t ) AND
               (?t > 2004 AND ?t < 2007)
          ( ?y rdf:type mesur:Citation )
          ( ?y mesur:hasSource ?c )
          ( ?y mesur:hasSink ?a )
          ( ?z rdf:type mesur:Publishes )
          ( ?z mesur:hasUnit ?c )
          ( ?z mesur:hasTime ?u) AND ?u = 2007

SELECT   ?y
WHERE
          ( ?y rdf:type mesur:Publishes )
          ( ?y mesur:hasGroup ?a )
          ( ?a mesur:partOf urn:issn:1082-9873 )
          ( ?y mesur:hasTime ?t ) AND
               (?t > 2004 AND ?t < 2007)

INSERT < _123 rdf:type mesur:ImpactFactor >
INSERT < _123 mesur:hasObject urn:issn:1082-9873 >
INSERT < _123 mesur:hasStartTime 2007 >
INSERT < _123 mesur:hasEndTime 2007 >
INSERT < _123 mesur:hasNumbericValue
                              (COUNT(?x) / COUNT(?y)) > .
```

The 2007 Usage Impact Factor for the JCDL Proceedings can be calculated by using the following SPARQL queries and INSERT commands. The 2007 Usage Impact Factor for the JCDL is defined as the number of usage events in 2007 that pertain to articles published in the JCDL proceedings in either 2005 or 2006 normalized by the total number of articles published by the JCDL in 2005 and 2006 [5].

```
SELECT   ?x
WHERE
          ( ?x rdf:type mesur:Uses )
          ( ?x mesur:hasDocument ?a )
          ( ?x mesur:hasTime ?t ) AND ?t = 2007
          ( ?y rdf:type mesur:Publishes )
          ( ?y mesur:hasUnit ?a )
          ( ?y mesur:hasGroup ?c )
          ( ?c mesur:partOf urn:issn:1082-9873 )
          ( ?y mesur:hasTime ?u ) AND
               (?u > 2004 AND ?u < 2007)

SELECT   ?y
WHERE
          ( ?y rdf:type mesur:Publishes )
          ( ?y mesur:hasGroup ?a )
          ( ?a mesur:partOf urn:issn:1082-9873 )
          ( ?y mesur:hasTime ?t ) AND
               (?t > 2004 OR ?t < 2007)

INSERT < _123 rdf:type mesur:UsageImpactFactor >
INSERT < _123 mesur:hasObject urn:issn:1082-9873 >
INSERT < _123 mesur:hasNumericValue
                              (COUNT(?x) / COUNT(?y)) > .
```

As demonstrated, the presented metrics can be easily calculated using simple SPARQL queries. However, more complex metrics, such as those that are recursive in definition, can be computed using other semantic network algorithms. For example, the eigenvector-based Y-Factor [3] can be computed in semantic networks using the grammar-based random walker framework presented in [21]. The objective of the MESUR project is to understand the space of such metrics and their application to valuing artifacts in the scholarly community. Future work in this area will report the finding that are derived from such algorithms.

7. CONCLUSION

This article presented the MESUR ontology which has been engineered to provide an integrated model of bibliographic, citation, and usage aspects of the scholarly community. The ontology focuses only on that information for which large-scale real world data exists, supports usage research, and whose instantiation is scalable to an estimated 50 million articles and 1 billion usage events. A novel approach to data representation was defined that leverages both relational database and triple store technology. The MESUR project was started in October of 2006 and thus, is still in its early stages of development. While a trim ontology has been presented, the effects of this ontology on load and query times is still inconclusive. Future work will present benchmark results of the MESUR triple store.

8. ACKNOWLEDGMENTS

This research is supported by a grant from the Andrew W. Mellon Foundation.

9. REFERENCES

[1] H. P. Alesso and C. F. Smith. *Developing Semantic Web Services*. A.K. Peters LTD, Wellesey, MA, 2005.

[2] C. Bizer. D2R - a database to RDF mapping language. In *The Twelth International World Wide Web Conference (WWW03)*, Budapest, Hungary, May 2003.

[3] J. Bollen, M. A. Rodriguez, and H. Van de Sompel. Journal status. *Scientometrics*, 69(3), December 2006.

[4] J. Bollen and H. Van de Sompel. Mapping the structure of science through usage. *Scientometrics*, 69(2), 2006.

[5] J. Bollen and H. Van de Sompel. Usage impact factor: the effects of sample characteristics on usage-based impact metrics. *[in review]*, 2006.

[6] J. Bollen and H. Van de Sompel. An architecture for the aggregation and analysis of scholarly usage data. In *Joint Conference on Digital Libraries (JCDL06)*, pages 298–307, Chapel Hill, NC, June 2008.

[7] J. Bollen, H. Van de Sompel, J. Smith, and R. Luce. Toward alternative metrics of journal impact: a comparison of download and citation data. *Information Processing and Management*, 41(6):1419–1440, 2005.

[8] T. Brody, S. Harnad, and L. Carr. Earlier web usage statistics as predictors of later citation impact. *Journal of the American Society for Information Science and Technology*, 57(8):1060 - 1072, 2006.

[9] T. Burners-Lee, W3C/MIT, R. Fielding, D. Software, L. Masinter, and A. Systems. Uniform Resource Identifier (URI): Generic Syntax, January 2005.

[10] E. Garfield. Journal impact factor: a brief review. *Canadian Medical Association Journal*, 161:979-980, 1999.

[11] M. A. Goncalves, M. Luo, R. Shen, M. F. Ali, and E. A. Fox. An XML log standard and tool for digital library logging analysis. In M. Agosti and C. Thanos, editors, *ECDL 2002: LNCS 2458*, pages 129–143, Berlin, September 2002. Springer-Verlag.

[12] M. J. Kurtz, G. Eichhorn, A. Accomazzi, C. S. Grant, M. Demleitner, and S. S. Murray. The bibliometric properties of article readership information. *Journal of the American Society for Information Science and Technology*, 56(2):111–128, 2005.

[13] C. Lagoze and J. Hunter. The ABC ontology and model. *Journal of Digital Information*, 2(2), 2001.

[14] R. Lee. Scalability report on triple store applications. Technical report, Massachusetts Institute of Technology, July 2004.

[15] A. Magkanaraki, G. Karvounarakis, T. T. Anh, V. Christophides, and D. Plexousakis. Ontology ontology storage and querying. Technical report, École Nationale Supérieure des Télécommunications, April 2002.

[16] F. Manola and E. Miller. RDF primer: W3C recommendation, February 2004.

[17] D. L. McGuinness and F. van Harmelen. OWL web ontology language overview, February 2004.

[18] N. F. Noy, W. Grosso, and M. A. Musen. The knowledge model of Protege-2000: Combining interoperability and flexibility. In *International Conference on Knowledge Engineering and Knowledge Management*, Juan-les-Pins, France, 2000.

[19] K. Portwin and P. Parvatikar. Building and managing a massive triple store: An experience report. In *XTech: Building Web 2.0*, Amsterdam, Netherlands, 2006.

[20] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF, October 2004.

[21] M. A. Rodriguez. Grammar-based random walkers in semantic networks. *http://tinyurl.com/278jtf*, 2007.

[22] G. Rust. Ontologyx. In *Functional Requirements for Bibliographic Records Workshop Proceedings*, Dublin, Ohio, May 2005.

[23] P. T. Shepherd. Project COUNTER - Setting international standards for online usage statistics. *Journal of Information Processing and Management*, 47(4):245 – 257, 2004.

[24] S. B. Shum, E. Motta, and J. Domingue. Scholonto: an ontology-based digital library server for research documents and discourse. *International Journal on Digital Libraries*, 3(3):237–248, 2000.

[25] J. F. Sowa. *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[26] S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H. P. Schnurr, R. Studer, and Y. Sure. Semantic community web portals. In *9th International World Wide Web Conference*, Amsterdam, Netherlands, May 2000.