Proceedings of the Second International Workshop on

Trust and Privacy on the Social and Semantic Web (SPOT 2010)

co-located with the
7th Annual Extended Semantic Web Conference
(ESWC 2010)
May 31, 2010, Heraklion, Crete, Greece

http://spot.semanticweb.org/2010/

Editors

Philipp Kärger

Daniel Olmedilla

Alexandre Passant

Axel Polleres

Published online as CEUR Workshop Proceedings, ISSN 1613-0073, Vol-576 — http://CEUR-WS.org/Vol-576

Preface

More than ever, the Semantic Web is becoming reality as it is an integrated component of the Web we are browsing everyday - be it the Open Linked Data movement that nowadays exposes over 10 billion triples of RDF or the annotated and structured information available on Web pages used by major search engines, such as Yahoo! SearchMonkey or Google. Moreover, social data about people and their interaction is made available in machine-understandable format in projects like FOAF or SIOC. Facing this amount of data, privacy and trust consideration is an important step to take right now. The challenging research questions arising from this movement include:

– How do people know that the data gathered from several sources for reasoning purposes can be trusted?
– How can one avoid that personal data exposed on the Semantic Web will be combined with other available semantic data in a way that sensitive information may be revealed?
– How shall a safe reasoning process look like that does not end up in a conflict only because a single Semantic Web peer exposed a contradiction?

The Second International Workshop for Trust and Privacy on the Social and Semantic Web (SPOT2010) presents discussions and results concerning questions like these and leads to solutions and research results in the realm of Semantic Web and social data for the pervasive issue of privacy and trust on the Web. SPOT2010 brings together, among others, researchers and developers from the field of Semantic Web, the Social Web, and trust and privacy enforcement. Similar to the successful SPOT2009, this year's workshop provides the opportunity to discuss and analyze important requirements and open research issues for a trustful Semantic Web.

We are grateful to the members of the Program Committee of SPOT2010 for their support. We would like to thank all the authors who submitted their work for their interesting contributions. Further on, we thank the European COST Action IC0801 "Agreement Technologies" and the Science Foundation Ireland for supporting the event (Grant No. SFI/08/CE/I1380 — Líon 2).

May 2010

Philipp Kärger
Daniel Olmedilla
Alexandre Passant
Axel Polleres
Organizing Committee of SPOT2010

Organization

SPOT 2010 is co-located with the Extended Semantic Web Confenference 2010 (ESWC).

Organizing Committee

Philipp Kärger - L3S Research Center, Hannover, Germany
Daniel Olmedilla - Telefonica R&D, Madrid, Spain
Alexandre Passant - DERI, NUI Galway, Galway, Ireland
Axel Polleres - DERI, NUI Galway, Galway, Ireland

Program Committee

Chris Bizer, Freie Universität Berlin, Germany
John Breslin, DERI, NUI Galway, Ireland
Dan Brickley, FOAF Project, World
Juri Luca De Coi, L3S Research Center, Germany
Stefan Decker, DERI, NUI Galway, Ireland
Fabien Gandon, INRIA Sophia-Antipolis, France
Harry Halpin, University of Edinburgh, UK
Olaf Hartig, Humboldt-Universität zu Berlin, Germany
Michael Hausenblas, DERI, NUI Galway, Ireland
Bettina Hoser, Universität Karlsruhe (TH), Germany
Lalana Kagal, Massachusetts Institute of Technology, USA
Jens Lehmann, Universitaet Leipzig, Germany
Javier Lopez, University of Malaga, Spain
Fabio Martinelli, National Research Council - C.N.R., Italy
Sascha Ossowski, Universidad Rey Juan Carlos, Spain
Matthew Rowe, University of Sheffield, UK
Simon Schenk, University of Koblenz-Landau, Germany
Daniel Schwabe, Pontificia Universidade Catolica do Rio de Janeiro, Brazil
Jean-Marc Seigneur, University of Geneva, Switzerland
Nigel Shadbolt, University of Southampton Highfield, UK
Carles Sierra, IIIA CSIC, Spain
Milan Stankovic, Hypios.com and LaLIC, Université Paris IV Sorbonne, France
Henry Story, Sun Microsystems, France
Krishnaprasad Thirunarayan, Kno.e.sis Center, Wright State University, USA
Alessandra Toninelli, Università di Bologna, Italy
Mischa Tuffield, Garlik, UK
Claudia Wagner, Joanneum Research, Austria

Additional Referees

Vinicius Almendra
Aliaksandr Lazouski
Peyman Nasirifard
Sergej Zerr

Sponsoring Organizations

Cost Action IC0801 on Agreement Technologies

Science Foundation Ireland, grant number SFI/08/CE/I1380 (Líon 2)

SWRL-based Access Policies for Linked Data

Hannes Mühleisen, Martin Kost and Johann-Christoph Freytag

Humboldt-Universität zu Berlin
Department of Computer Science
hannes@muehleisen.org, {kost | freytag}@informatik.hu-berlin.de

Abstract. Social applications are one of the fastest growing areas in the Web. However, privacy issues ensue if all information of all users of these applications is stored on a single computer system. With small extensions to Semantic Web technologies and Linked Data concepts, a distributed approach to the social web is possible, where users retain fine-grained control over their data and are still able to combine their data with users on different systems. We describe our concept of a Policy-enabled Linked Data Server (PeLDS) obeying user-defined access policies for the stored information. PeLDS also supports configuration-free distributed authentication. Access policies are expressed in a newly developed compact notation for the Semantic Web Rule Language. Authentication is performed using SSL certificates and the FOAF+SSL verification approach. We evaluate our concept using a prototype implementation and a distributed address book application.

1 Introduction

The Semantic Web as a new generation of the World Wide Web allows its users to share content over the boundaries of applications and web sites. To achieve this goal, the resources of the WWW are annotated using machine-readable meta data. The principles of Linked Data [2] describe a set of conventions how this meta data should be structured and published. So far, no access control mechanism supporting fine-grained access policies is available for Linked Data, although a number of fitting scenarios are conceivable.

Previous web-based systems for the controlled distribution of sensitive information require the presence of information on centralized systems. In order to control the access to the managed information, these systems usually support secured data storage, access policies for the stored data and user authentication. Examples for such systems include various social networks: all users sign in on a central website to store their information there. Users configure their privacy settings on that website, for instance to set their telephone number only to be visible to a particular group of users. However, system operators always have access to all of the stored information. This is an unsatisfactory situation, as the operator's behavior cannot be foreseen. Moreover, if a malicious user manages to circumvent the access restrictions, the information stored by all users is exposed and each user's privacy is endangered.

We will outline an alternative approach, where users store their information on a system under their control. Integration with users on other systems is supported, and the access to the stored information can be controlled by the users themselves. To implement our approach, we use and combine Semantic Web technologies. We also slightly extended some technologies to enable users publishing Linked Data content to specify who is allowed to retrieve their information. This way, distributed social web applications with support for sophisticated access policies can be developed.

The rest of this paper is structured as follows: in Section 2 we describe the related work in the field of access policies for Semantic Web data, Section 3 describes the requirements to formulate access policies for RDF data. Section 4 describes our concept design of the Policy-enabled Linked Data Server (PeLDS). Section 5 shows our results in evaluating a PeLDS prototype and our demonstration application, called the "Distributed Address Book". Finally, Section 6 concludes this paper.

2 Related Work

Tootoonchian et al. describe a privacy management system named "Lockr" especially designed for social networks [18], thus their policy format is limited to describe required social relationships for data access. However, by using a general-purpose format such as W3C's Resource Description Format (RDF) as data format, this domain-specific problem can be handled by a more general approach. Hollenbach, Presbrey and Berners-Lee present an approach where RDF metadata is used to describe general-purpose access policies to RDF files stored on a web server [9]. While their approach mentions the possibility of extending access control to the data model level, they developed and evaluated access policies only for atomic RDF files. Research in the area of access policy languages for RDF data is exhaustively described in an article by Duma, Herzog and Shahmehri [6]. They especially conclude on the need for fine-grained access policy languages for RDF graphs and their elements. Reddivari, Finin and Joshi developed such a language in [16] as well as an implementation of a system evaluating these access policies based on the Jena inferencing engine. Jain and Farkas follow a similar approach [11]. They also show why access policies developed for XML data representations are not applicable to RDF data. Once access policies are defined, their enforcement presents another challenging task. Abel et al. developed such a mechanism [1] using access policies and query expansion. Neither of the mentioned solutions were available as an implementation ready for usage or evaluation at the time of this writing. Additionally, Web Ontology Language (OWL) reasoning and the handling of the inferred information is not supported by these approaches. OWL and its evaluation make powerful access policies possible, as we will show later.

The language Rei [12] was considered most suitable for the task of enabling users to express their access policies for semantic web data, but focuses more on developing an ontology for policy expression than to their actual application. We therefore chose to work on a more abstract level, that is, create a policy language and evaluation methods suitable for any RDF-based access policy expression language such as Rei. At the same time, we stress compatibility to the Linked Data principles, and are thus unable to build upon concepts that require a special form of protocol or trust negotiation, for example Protune [5].

The challenge of authenticating users in the Semantic Web environment is not as straightforward as in the conventional WWW, as browsing activities can require requests to multiple systems. Story et al. [17] have presented the FOAF+SSL concept for distributed authentication designed to make a distributed social web feasible.

3 Access Policies

The enforcement of explicit access policies can be used to limit access to data. In simple cases, these access policies consist of lists of users with access privileges. This eliminates the need for complex access policy evaluation, but only enables very crude access control. A more thorough approach allows users to specify custom access policies on various levels of expressiveness. These access policies are then used to determine which data the current user is allowed to retrieve or manipulate. Most users have an intuitive notion which information should be made publicly available, and which information should only be released to a limited group of people. Expressing intuitive access policies in a formal way rises the challenge of bridging the gap between human intuition and machine-readable definition. We begin our discussion of access policies with the definition of the term "access policy":

Definition 1. *An Access Policy is a set of rules. These rules are evaluated in order to decide whether a user is allowed to access a data object [13].*

3.1 Types of Access Policies

Whereas one could always write a custom program to decide which information should be communicated, a declarative expression of access policies is commonly preferred to keep policy expression and policy evaluation apart. In general, we distinguish three different types of access policy patterns:

Discretionary Access Control (DAC) distinguishes between named users and named objects such as files. A mechanism allows users with access rights to a specific object to award their access rights to other users or groups of users. There is a way to limit the propagation of access rights to sub-objects. The granularity of access rights can be refined to the level of individual users and objects [13]. A well-known example for DAC is the UNIX file systems file permissions model.

Mandatory Access Control (MAC) requires all objects and users to be described by a global access policy. Every object and every user is annotated with a security classification level. These classification levels are organized in a hierarchical way and provide the basis to decide whether an object can be accessed. Users may only access an object, if they possess a security classification equal or higher in the security hierarchy. Users cannot award their access rights to other users [13]. Systems supporting MAC are frequently used by public authorities, for example, to control access to confidential ("classified") information.

Role-based Access Control (RbAC) does not distinguish between single users regarding access rights. Users are simply given a "role" according to their assignments, all access rights are bound to that role. A user can possess multiple roles. RbAC is a simplified form of MAC lacking security classification hierarchies, but derivation and composition of roles are still possible [7].

3.2 Data Classification

In order to express access policies for an RDF graph, the user has to describe those parts of the graph to be affected by a particular rule. This process is referred to as "classification" here. Data classifications can be defined on multiple abstraction levels of an RDF graph. We distinguish three different levels with increasing abstraction: syntactical level, data model level and semantic level. An RDF graph in its serialized form can be assigned to the syntactical level. Graphs can be decomposed into triples containing a subject, a predicate and an object element. These triples belong to the data model level. Resource descriptions, their affiliation with concepts and relationships to other resources are on the semantic level.

Data classification for RDF graphs on the *syntactical level* has been shown to be ineffective [11], mainly because serialization formats permit multiple ways of representing an identical graph. This classification level is therefore not pursued further.

Reasonable data classification for RDF graphs can be performed on the *data model level*. Triples are the smallest units to be classified. They are logically independent of any syntactical representation and can be classified easily through the usage of triple patterns. Triple patterns describe matching conditions for each triple element and can be used to select graph elements. On each request a system is able to classify (and hence control access to) every triple by evaluating all triple patterns currently present, an approach also followed in [11]. Wildcards can be used to classify a set of triples or triples with unknown values. An example for this classification is contained in Listing 1.1 within the following section.

Finally, the *semantic level* allows classification of data based on concepts defined with schema languages such as RDF Schema or OWL. This classification allows for a set of related triples to be classified by a single pattern. As the classification references the concept definition, updates to the concepts are automatically considered for classification. Resources can be classified indirectly by assigning them to a classified concept with OWL statements. One approach allowing data classification on the basis of RDFS concepts is described in [16]. However, OWL support is desirable due to its more powerful expressions for concept and property relationships, for instance transitive properties.

3.3 Semantic Web Rule Language

The Semantic Web Rule Language (SWRL) is a generic rule language for Semantic Web data. SWRL rules can be evaluated by a reasoning program such as Pellet [4], KAON2 or RacerPro. SWRL rules can be represented using an RDF graph, and thus allow easy rule handling along with the RDF graphs containing the information to be protected.

SWRL rules describe implications and consist of two lists of predicates, the antecedent and the consequent part. If all predicates of the antecedent take the Boolean value *true*, all predicates in the consequent part are evaluated. The usable predicates are given by SWRL's language specification. The predicates are listed below with their conditions under which they will be evaluated to *true*:

– $C(x)$ - A resource x is an instance of the concept C.

- $D(z)$ - The value z is of data type D.
- $P(x, y)$ - The resource x has a property P with a reference to the object y.
- $Q(x, z)$ - The resource x has a property Q with the literal value z.
- $sameAs(x, y)$ - x and y identify identical resources.
- $differentFrom(x, y)$ - x and y identify distinct resources.
- $builtin(r, z_1, \ldots, z_n)$ - The built-in function r with the parameters z_* returns *true*. A number of functions providing standard comparisons are defined by the language specification. Additional functions can be added by the user if required.

The RDF or XML representation of SWRL rules is not designed to be human-readable, thus it is usually displayed in a Prolog-like notation [10]. However, this notation is not intended to be interpreted by a computer, thus SWRL rules are usually written using specialized programs. Our PeLDS concept uses SWRL for the expression of access policies, see Section 4.

4 PeLDS System Concept

The main feature for our concept of a Policy-enabled Linked Data Server is to provide a semantic storage system which allows its users to specify which elements of their RDF graphs are published to which user. This is achieved by creating a temporary view on the stored graphs that contain only those elements the querying user has been authorized to retrieve by the publishing user. The access policy is expressed in a custom policy language. This language can be used to implement all types of access policies described in Section 3. The concept can be compared to views on relations in relational databases.

The entire data stored is partitioned into *datasets* using named graphs to support multiple users. To achieve this, every triple stored is assigned to a graph identifier. This way, all triples belonging to a specific graph can be retrieved from the storage component. This mechanism is used here to achieve multi-user capabilities: storage operations require a graph identifier to be specified, and access policies as well as ownership information are bound to each single named graph.

As access policies contain rules, we have decided to use a general-purpose rule language to express our access policies. We start by introducing our descriptive access policy language PsSF based on SWRL, then we describe the algorithms for policy evaluation, and we finish with a description of the various operations provided by PeLDS.

4.1 Policy Language PsSF

Access policies are described as a set of rules defining access conditions for each dataset stored on the server. Users publishing data on the system can define an access policy for each dataset they have created. The system guarantees the enforcement of a valid policy during each operation involving this dataset. To facilitate the easy description of access policies, we have developed a short notation for policies and rules we call “Prolog-style SWRL Format (PsSF)”. Each rule consists of a label, a rule antecedent describing the condition under which the rule is satisfied and a consequent. Both the antecedent and the consequent contain a collection of predicates joined by the logical *AND* condition. In addition to all of SWRL's predicates described in Section 3.3, PsSF supports three predicates enabling data classification on the identified levels. These classification predicates can only be used in rule consequents.

– *permit_triple(subject,predicate,object)* - Access to all triples matching the parameters *subject*, *predicate* and *object* is permitted. All parameters may be replaced with the wildcard character $*$.
– *permit_resource(resourceUri)* - Access to the resource with the identifier *resourceUri* is permitted. The wildcard $*$ can be used to enable access to arbitrary resources (e.g. the complete dataset). This predicate is merely a special case of the first one.
– *permit_instance(conceptUri)* - Access to all instances of the concept identified by *conceptUri* as well as all instances of derived concepts is permitted.

Each access rule can be defined according to different types of access. Currently, we only distinguish query and update actions. Conditions have to be expressed in a positive fashion, negation is currently not supported for decidability reasons [14]. The rule syntax is described in detail with examples in [15]. Using this syntax, users can specify their access policies.

Listing 1.1 gives an example of a PsSF rule. The rule expresses the following notion: the user Horst is permitted to access Anna's phone number. The rule is labeled *phoneRule* and contains two antecedent predicates. The first predicate specifies the *?action* resource to be an instance of the concept *QueryAction*, the second predicate requires the *actor* resource to have `http://example.com/horst` as the value for its *actor* property. The consequent consists of a data classification predicate covering all triples with resource `http://example.com/anna`, property `ex:phone`, and arbitrary values.

```
phoneRule:
QueryAction(?action) && actor(?action ,http://example.com/horst)
 => permit_triple(http://example.com/anna,ex:phone,*);
```

Listing 1.1. Example PsSF rule

4.2 Policy Schema and Evaluation

We have developed a simple OWL schema to describe the actions performed on the stored datasets and make query meta data accessible for PsSF rules. Three main concepts are defined: *Action* for query-related meta data, *Rule* to model single rules as a part of access policies, and *TriplePattern* for defined data classifications. The concepts *UpdateAction* and *QueryAction* are derived concepts to model the different interaction types. Each action holds a user identifier and a one-to-many relationship to the rules defined by the access policy. Each rule contains a reference to its data classifications within the TriplePattern instances.

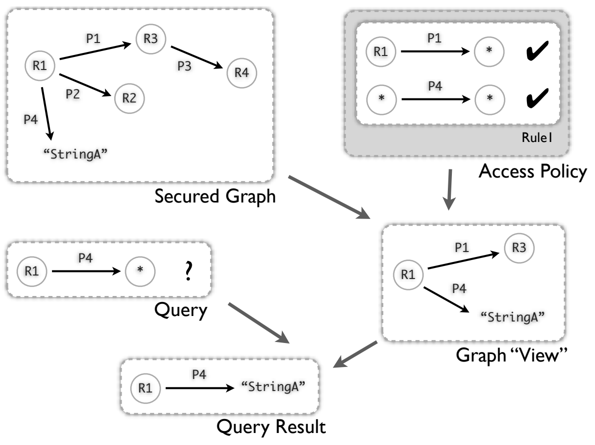


Fig. 1. Query data flow for PeLDS

An RDF graph containing the appropriate instances of the described schema is created for every request and merged with the affected dataset according to the defined access policy. Each rule from the access policy is attributed with an additional consequence to add the rule identifier to a global list of matched rules. If such a rule matches due to sufficient access rights for the current user, it will be added to this list. A reasoner performs rule evaluation by reasoning on the defined OWL and SWRL rules. PeLDS is then able to determine the data classifications defining the graph elements the current user is authorized to retrieve.

Based on an empty result graph, the requested dataset is loaded into memory and the access policy is translated into instances of the policy schema and is added to the dataset. The list of rules is evaluated for the information present in the dataset together with the user identity given in the Action instance. If a rule matches, every triple matching the data classifications contained in the rules consequence predicate list is copied from the dataset to the result graph. The user's query is now executed on the result graph, and the query results are sent back to the user. This process is depicted in Figure 1: a secured graph containing various resources is queried. The specified access policy allows access to all triples with resource *R1* and property *P1* and all triples with the property *P4*. Hence, the triples (*R1*,*P1*,*R3*) and (*R1*,*P4*,`"StringA"`) form the temporary graph view authorized for the current user. The user's query for the value of the property *P4* on the resource *R1* can then be answered with the corresponding element of the graph view.

4.3 Encrypted Communication and Authentication

The Linked Data principles include the principle of dereferencing: resolving a URL found as an identifier within an RDF graph yields another RDF graph describing the resource identified by this URL. To implement arbitrary dereferencing, authentication cannot be based on shared secrets, as any URL may appear within an RDF graph. As a consequence, user name/password or local trust settings for certificates are neither an elegant nor a scalable solution.

We use the HTTP Secure Protocol (HTTPS) for communication as well as certificate exchange and the approach presented by Story et al. [17] to validate SSL client certificates: The URL describing the user making a request is included within the SSL client certificate used to sign the HTTP request to an HTTPS server. Dereferencing this URL yields an RDF graph containing RDF triples defined by the Friend-of-a-Friend (FOAF) vocabulary. This graph also contains meta information about the cryptographic key used to sign the request, which is only available to the owner of the specific key. For RSA keys, this is the modulus and exponent of the private key. The server receiving the request is now able to verify whether this request was issued by the person controlling the URL included in the certificate, which is sufficient to identify the requesting user. This authentication mechanism does not rely on a global trust system or local settings. The concept can be used to authenticate Linked Data clients in a safe manner. Even though we have chosen FOAF+SSL due to its use of Linked Data, other authentication solutions such as OpenID could also be used and integrated.

4.4 Interface and Operations

To maximize compatibility to existing software components, the PeLDS API was designed to be as consistent as possible regarding existing standards for handling RDF graphs. Additional API operations were added to enable policy management. In total, four main operations were identified: policy update, data update, data query, and dereferencing. Users are assumed to be authenticated and identified through their URLs. Datasets are generally created if the update operation is given an unknown graph identifier, they are then annotated with the URL identifying the user issuing the corresponding operation and thus “owned” by this user.

Policy Update - *reponseCode = updatePolicy(datasetUri, policy)*
Only the dataset owner may specify the access policy. The dataset URI has to be specified along with an access policy detailing which parts of the dataset should be disclosed. The new access policy is stored if it is syntactically correct according to the PsSF language specification [15], and the corresponding response code is returned to the user. Any existing access policy is overwritten.

Data Update - *reponseCode = updateData(datasetUri, update)*
RDF graphs can be uploaded, changed and deleted. This is facilitated using an update statement describing which graph elements to change. This is preferable to sending the entire graph for each update operation, as less graph elements have to be communicated. If the dataset owner issues an update, it is approved without further action. If another user tries to update the dataset, all changed graph elements have to be approved by the corresponding access policy. An update is only stored to persistent storage if no error has occurred.

Data Query - *result = queryData(datasetUri, query)*
An RDF graph (or parts of it) stored on a PeLDS instance are retrieved using a query language. The user has to specify a dataset identifier and a query. The query is executed on the elements the authenticated user is authorized to see by the access policy present for the requested dataset. This process was described in Section 4.2. The query

is expressed in a query language suitable for querying RDF graphs. The result value contains authorized graph elements from the specified dataset matching the query. If the specified dataset does not exist, no error is returned, because the awareness of the existence of a dataset can already be sensitive information. If no access policy is set for the dataset, only the dataset owner may issue queries to it. To achieve compatibility with existing software packages, authentication is optional for this operation.

Dereferencing - *result = dereference(resourceUrl)*
In order to fulfill the Linked Data requirements, PeLDS must be able to deliver an RDF graph further describing a resource with only a given URL which may be described in any dataset. This operation takes a resource identifier in URL form as argument. The operation looks up all datasets containing graph elements describing the resource and evaluates their access policies. It then delivers a result containing graph elements describing this resource, if a) elements describing this resource are stored and b) the authenticated user is authorized to view a subset of these elements. Similar to the query operation, authentication is optional here.

5 Evaluation

We have implemented a prototype of the PeLDS system described in Section 4 as a Java application. All operations are performed using the HTTP protocol. This prototype supports all specified API operations and is able to evaluate the PsSF access policy format for each stored dataset, thus satisfying the identified functional requirements. Policy evaluation and OWL reasoning is performed by the Pellet reasoning program [4]. W3C standards are obeyed and supported where applicable, for example the SPARQL query language, the SPARQL results format, the various RDF serializations like RDF/XML and N3, the SPARQL HTTP protocol, and the SPARQL/Update update language. In this section, we evaluate system security, system performance, and describe the distributed address book we have implemented as a demonstration application on the basis of the PeLDS prototype.

5.1 System Performance

Determining the system performance for our prototype is not an obvious task, as the only comparable system mentioned in [16] is not available for testing. However, systems supporting a subset of PeLDS' features are available, hence we were able to test such a system for comparison. The SPARQL server Joseki [8] supports querying and modification of RDF storage systems over an HTTP interface and data separation in multiple datasets, but not evaluation of access policies. Joseki was backed by the triple store Jena TDB and - optionally - the reasoning program Pellet [4] for OWL inference. The test was intended to show the additional effort required for the evaluation of our access policies.

To make query results comparable for all test runs, a special access policy was installed in the PeLDS prototype allowing read and write access to the stored data without any authentication. The test data generator included in the Berlin SPARQL Benchmark

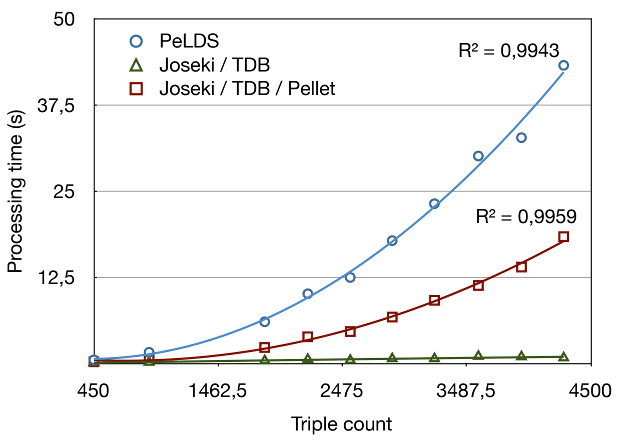


Fig. 2. Query performance

[3] was used to generate a sufficient amount of triples for testing in datasets of different sizes. Each dataset was imported into the respective system and a simple SPARQL query returning all stored and inferred triples was executed at least three times for each dataset. The shortest time required to complete the query was taken as a test result, in order to acknowledge caching strategies. Test results are given as a scatter plot in Figure 2 with the x-axis describing the amount of triples and the y-axis the time required to complete the test query. As the result of the R^2 least squares fitting test, an approximation to a polynom of second degree is also plotted.

The difference in results of the Joseki instance with and without reasoning support illustrates the amount of time required for reasoning in general. The PeLDS prototype requires additional time for access policy evaluation, however, this effort only increases in a linear fashion as the dataset grows. The approximation tests yielded polynomial complexity for both Joseki and PeLDS which is mainly attributed to reasoning activities.

5.2 Security Considerations

PeLDS's security directly depends on secure authentication. If an attacker is able to circumvent the FOAF+SSL authentication scheme, unauthorized access is possible. FOAF+SSL relies on dereferencing the URL identifying a user, if an attacker gains control over that URL, for example by manipulating DNS entries, he is able to take that identity. However, recent improvements to the Internet architecture such as DNSSEC aim at impeding such attacks. Another attack vector are the queries specified. They do not pose a security threat by themselves, as they are not evaluated over the global database. Queries can only "see" a temporary graph containing only the elements of the specified datasets authorized for the current user.

5.3 Demo Application: Distributed Address Book

In order to demonstrate the capabilities of the PeLDS system, we have implemented a demo application with a storage layer solely based on PeLDS. This application implements a distributed address book as a web application. Users can manage their contact profiles and contacts within this address book. Users are identified by a URL and all information about the contacts is retrieved in real-time from the server their corresponding profile is stored on. Users can organize their contacts in groups and assign visibilities to each data item stored in their profile. For example, a user may define her telephone number to be private and only visible to her family.

All user data is stored within a PeLDS instance. Storage communication is handled via SPARQL and SPARQL/Update, respectively. The privacy settings the users define are translated into PeLDS access policies and activated for their personal data. Access to user data is controlled by PeLDS, thus only clients properly identifying themselves and authorized can retrieve protected information. Data integration and user identification is performed using Linked Data principles, all a user has to know to add another user to his address book is the URL describing her.

In contrast to popular systems, our Distributed Address Book leaves all personal data under the control of each user, a central instance is not required. Also, an arbitrary client program capable of displaying RDF information can be used to view and manage address book entries, given that this client supports the usage of SSL certificates.

6 Conclusion and Outlook

Following a survey of existing work in the area of access control for Semantic Web storage systems, we commenced on detailing the different types of access policies and the methods for data classification within the contexts of RDF graphs used to represent Semantic Web content. We then explained the Semantic Web Rule Language as a possible candidate for a rule format in access policies. Our concept of a Policy-enabled Linked Data server was described. PeLDS consists of our access policy language PsSF, which extends SWRL by adding custom predicates for data classification, an OWL policy schema, the FOAF+SSL authentication mechanism and a high-level API definition of the different operations provided. This concept was implemented as a prototype and evaluated for system performance in comparison with an established solution, Joseki. Our Distributed Address Book based on PeLDS was introduced to show the kind of both distributed and privacy aware applications now possible. The PeLDS prototype and the Address Book are available as open source software and can be downloaded at `http://www.pelds.org`.

We would like to extend both the PeLDS concept and prototype with more features such as negation support within our policy language. Performance optimizations are another area of future work, as scalability was not the main goal for the implementation of the prototype. API operations for detailed modifications of single rules instead of whole access policies may also be desirable.

Acknowledgments We would like to thank the anonymous reviewers, Andy Seaborne for his answers, and Benjamin Nowack for the approval of our patches.

References

1. Fabian Abel, Juri Luca De Coi, Nicola Henze, Arne Wolf Koesling, Daniel Krause, and Daniel Olmedilla. Enabling advanced and context-dependent access control in RDF stores. In *ISWC2007/ASWC2007*, pages 1–14, 2007.
2. Tim Berners-Lee. Linked data, 2006. http://www.w3.org/DesignIssues/LinkedData.html accessed on 2010-04-20.
3. Christian Bizer and Andreas Schultz. The Berlin SPARQL benchmark. *International Journal On Semantic Web and Information Systems - Special Issue on Scalability and Performance of Semantic Web Systems*, 2009.
4. Clark and Parsia. Pellet: The open source OWL reasoner, 2009. http://clarkparsia.com/pellet accessed on 2010-04-20.
5. Juri Luca De Coi, Daniel Olmedilla, Piero A. Bonatti, and Luigi Sauro. Protune: A framework for semantic web policies. In Christian Bizer and Anupam Joshi, editors, *International Semantic Web Conference (Posters & Demos)*, volume 401 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
6. Claudiu Duma, Almut Herzog, and Nahid Shahmehri. Privacy in the semantic web: What policy languages have to offer. *Policies for Distributed Systems and Networks, IEEE International Workshop on*, 0:109–118, 2007.
7. David Ferraiolo and Richard Kuhn. Role-based access control. In *Proceedings of 15th National Computer Security Conference*, 1992.
8. Hewlett-Packard Development Company. Joseki - a SPARQL server for jena, 2009. http://www.joseki.org accessed on 2010-04-20.
9. James Hollenbach, Joe Presbrey, and Tim Berners-Lee. Using RDF metadata to enable access control on the social semantic web. In *Proceedings of the Workshop on Collaborative Construction, Management and Linking of Structured Knowledge (CK2009)*, October 2009.
10. Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. SWRL: A semantic web rule language, 2004. http://www.w3.org/Submission/SWRL accessed on 2010-04-20.
11. Amit Jain and Csilla Farkas. Secure resource description framework: an access control model. In *SACMAT'06*, 2006.
12. Lalana Kagal, Timothy Finin, and Anupam Joshi. A policy language for a pervasive computing environment. In *IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, pages 63–74, 2003.
13. Donald C. Latham. *Trusted Computer System Evaluation Criteria (Orange Book)*. Department of Denfense, 1985.
14. Boris Motik, Ulrike Sattler, and Rudi Studer. Query answering for OWL-DL with rules. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2005.
15. Hannes Mühleisen. Zugriffsrichtlinien und Authentifizierung für Linked-Data-Systeme. Master's thesis, Humboldt-Universität zu Berlin, http://muehleisen.org/da-hm-ld.pdf, 2009.
16. Pavan Reddivari, Tim Finin, and Anupam Joshi. Policy-based access control for an RDF store. In *Proceedings of the IJCAI-07 Workshop on Semantic Web for Collaborative Knowledge Acquisition*, January 2007.
17. Henry Story, Bruno Harbulot, Ian Jacobi, and Mike Jones. FOAF+SSL: RESTful authentication for the social web. Submitted to Semantic Web Conference 2009, 2009.
18. Amin Tootoonchian, Stefan Saroiu, Yashar Ganjali, and Alec Wolman. Lockr: better privacy for social networks. In *CoNEXT '09: Proceedings*, pages 169–180, New York, NY, USA, 2009. ACM.

Semantic Monitoring of Personal Web Activity to Support the Management of Trust and Privacy

Mathieu d'Aquin, Salman Elahi, Enrico Motta

Knowledge Media Institute, The Open University, Milton Keynes, UK
{m.daquin, s.elahi, e.motta}@open.ac.uk

Abstract. For individual Web users, understanding and controlling their exchange of personal data is a very complex task as they interact, sometimes unknowingly, with hundreds of different websites. In this paper, we present a set of tools and an experiment dedicated to monitoring a user's Web activity in order to build an observed model of his behavior in terms of the trust given to accessed websites and of the criticality of the data exchanged. By exposing such a model to the users and allowing them to interact with it, we provide ways for users to be better informed about their own behavior with respect to privacy, and ultimately, to better control their own data exchange.

1 Introduction

Web users send data of varying degrees of criticality, to websites which they trust to various levels. Reasonable users would for example agree for a well known online retailer website to know about their address, while would normally not be conformable with sending data more critical than their screen resolution to a completely unknown or untrusted website. Indeed, it is expected that an informed and rational user naturally implements a "personal policy" relying on an implicit model of the trust relationship they have with websites, and of the criticality of their own data. However, the inherent complexity, the fragmentation and the implicitness of Web data exchange makes it almost impossible for any user to be adequately informed. Not many users would be able to list the websites to which they have sent a particular information such as their e-mail address for example. Even more difficult is to know how much information is transfered to an unknown website, as a side effect of accessing a trusted one.

In this paper, we present a tool and an experiment intended to demonstrate how we can derive *implicit models* of trust in domains (i.e., websites) and criticality of data from locally generated traces of the user's activity on the Web. We rely on data generated through a logging mechanism that keeps track in RDF of any communication through the HTTP protocol occurring on the user's computer [1]. We then develop simple models and tools to extract and represent data transfers from the generated logs, and to map these data transfers onto a semantic profile of the user.

The main contribution of this paper is to show how we can derive from this data notions of the *observed trust in domains* and *observed criticality of data*. Indeed, intuitively, these two notions relate with each other in the sense that we consider that a website is highly trusted if it has been sent very critical data, while a piece of data is not critical if it was send to many untrusted websites. Making such notions explicit to the user and allowing them to explore the underlying data can be very useful as it makes emerge implicit relationships between the user, websites and his data, which might not be expected or intended by the user. In addition, we propose an interactive tool allowing users to 'align' the observed behavior with their 'intended' model of trust and criticality, in order to identify conflicts that can be acted upon.

As a concrete experiment for these tools, we detail at each step the results obtained using the Web activity logs generated by the first author of the paper over a period of 2.5 months.

2 Related Work

As lengthly described in [2], trust is a central element of any social interaction, and therefore, of any exchange on the Web. Indeed, beyond the Web 2.0 emphasis on the Web as a social platform, where people can exchange and share information, experience and more, any communication on the Web appears to be a social interaction between a person (Web user) and a website, which ultimately represents another person, group of people or organization. Therefore, a lot of attention has been dedicated to the notion of trust in the research community on the Web [3, 4]. While it would be out of the scope of this paper to detail these initiatives, we can mention as examples works where trust is considered a value attached to information, or to the provider of information, and that quantify the confidence one has that the information provided is correct (see e.g., [5]). In other cases, trust relates more to the notion of privacy, where it is attached to the recipient of some (usually critical, personal) data and corresponds to the confidence one has that the data would not be used for unintended purposes. For example, the *Platform for Privacy Preferences*[1] (P3p) provides a framework for websites to express their privacy practices, declaring explicitly in which way they can be trusted in handling user data. The work presented here is more directly related to this second category. However, contrary to P3P which takes a 'website-centric' perspective on trust, we consider here a user-centric view on trust in websites (domains) and on the criticality of the data sent to these websites. The intent is to derive from the traces of the user's activity a model of his own trust relationship with the various websites he interacts with.

A range of tools exist already to support a user in monitoring his own Web activity, including tools used to debug communication protocols. More related to our approach here, we can mention for example Google Web History[2] and

[1] http://www.w3.org/P3P/
[2] https://www.google.com/history/

the Attention Recorder[3]. Both take the form of local applications (e.g., a plugin for popular web browsers), and record accesses to websites in order to build a record of Web activities. However, such tools are still limited in the sense that they record only a restricted amount of information (websites explicitly accessed through the Web browser) and only allow usage of the data which is directly intended by the tool (provide reports and improve the results of search in one case, sharing 'attention data' in the other). As it appears in our experiment (see below), data exchange on the Web is a very complex activity, often fragmented and partly implicit. Dedicated tools are therefore needed to monitor them and derive equally complex trust relationships between the user and the corresponding websites.

3 Tracking User Web Activities for Data Transfer

Our goal in this paper is to use the traces of users' Web activity to build a model of their trust relationship with websites, and of the criticality of their own data. We first need to briefly introduce the underlying technology allowing us to obtain complete RDF models of Web activity and data transfer. We also detail at each step the results obtained in our experiment realized over a period of 2.5 months with the Web activity of the first author of this paper.

3.1 Logging Web Activity

In order to represent a sufficiently broad overview of personal data transfer on the Web, we need a tool which would fulfill two main requirements: 1- it needs to be transparent to the user, acting in background without disrupting normal Web activities; and 2- it needs to collect information as complete as possible, in particular, independently from the Web agent used (various Web browsers, but also many other tools such as online music programs—e.g., iTunes[4] and spotify[5], e-mail clients—getting Web images and other content from inside e-mails, etc.) For these reasons, we implemented our logging mechanism as a Web proxy, running on the local computer of the user. A proxy is a tool that acts as an intermediary between a client and external servers the client is trying to connect to. Web proxies are often used in organizations to implement cache control mechanisms for all the Web users inside the organization's network.

Here however, we use a Web proxy locally. Web communications can be redirected to it through the system preferences so that any HTTP request going out of the user's computer (and any response back) is intercepted, logged and re-directed to the right destination (which could be another Web proxy). As shown in Figure 1, the logs of the Web activity collected through this tool are

[3] https://addons.mozilla.org/en-US/firefox/addon/3569
[4] http://www.apple.com/itunes/
[5] http://spotify.com

represented in RDF, using a simple, ad-hoc HTTP ontology[6]. These logs record the complete information included as part of the HTTP protocol (e.g., destination, agent, cache information, referrers, etc.), as well as pointers to the actual data exchanged, which is saved on the local file system.

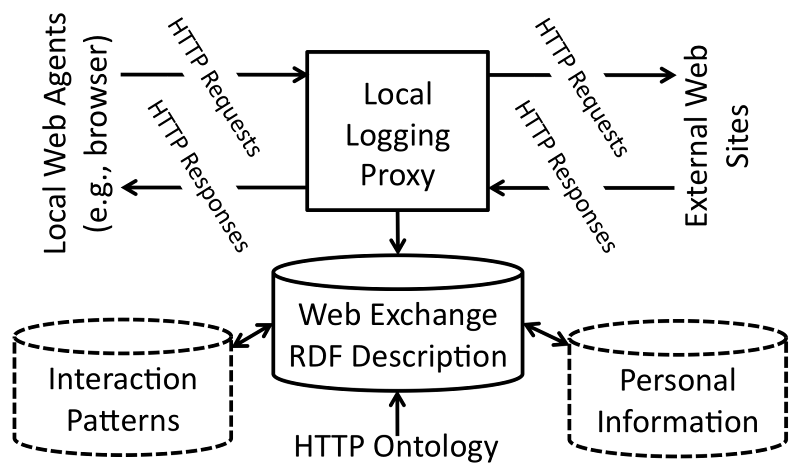


Fig. 1. Overview of the Web activity logging system.

In our experiment, this tool has recorded over 3 million HTTP requests during a period of 2.5 months, spanning over many different Web agents on the user's local computer, and representing all together 100 million RDF triples and 9GB of data (in the RDF/XML syntax). The scalability of the tool and its ability to process such data in real time represents a major challenge for this work. This is however outside the scope of this paper and will be treated as future work.

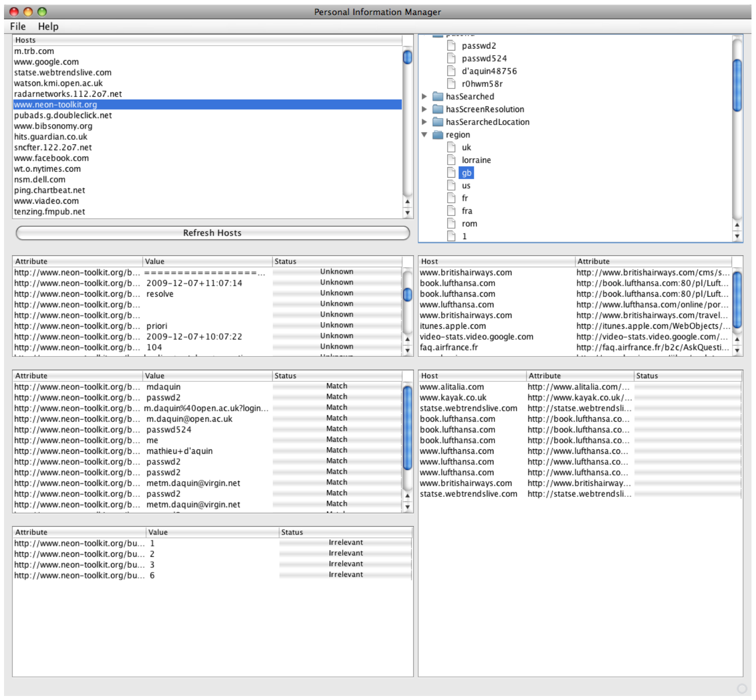
3.2 Investigating Data Transfer

Of course, the data collected using the tool described above contains a lot more information than necessary for the purpose of investigating data transfer to model trust in websites and criticality of data. We therefore extract from this data a subset that corresponds to elements of data that are being sent by the user's Web agents to external websites. We use a simple SPARQL query to obtain the list of requests to which data was attached. This includes HTTP GET requests with parameters (e.g., in http://www.google.co.uk/search?q=keywords

[6] This ontology was built for the purpose of the tool, to fit the data, but can be seen as an extension of http://www.w3.org/TR/HTTP-in-RDF/

the parameter is `q=keyword`), as well as HTTP POST requests where the same kind of parameters are enclosed in the content (data) part of the request. Parsing these parameters in both the URLs of GET requests and the data content of POST requests, we build a smaller dataset made of triples of the form $< website, \texttt{hasReceivedWithParam-}P, v >$, where *website* is the host to which the request was addressed, P is the attribute part of a parameter, and v is the value part.

Based on the activity logs in our experiment, we extracted more than 33,000 of such triples.



Hosts
m.trb.com
www.google.com
statse.webtrendslive.com
watson.kmi.open.ac.uk
radarnetworks.112.2o7.net
www.neon-toolkit.org
pubads.g.doubleclick.net
www.bibsonomy.org
hits.guardian.co.uk
sncfter.122.2o7.net
www.facebook.com
wt.o.nytimes.com
nsm.dell.com
ping.chartbeat.net
www.viadeo.com
tenzing.fmpub.net

```
passwd2
passwd524
d'aquin48756
r0hwm58r
hasSearched
hasScreenResolution
hasSerarchedLocation
region
    uk
    lorraine
    gb
    us
    fr
    fra
    rom
    1
```

Attribute	Value	Status
http://www.neon-toolkit.org/b...	==================...	Unknown
http://www.neon-toolkit.org/b...	2009-12-07+11:07:14	Unknown
http://www.neon-toolkit.org/b...	resolve	Unknown
http://www.neon-toolkit.org/b...		Unknown
http://www.neon-toolkit.org/b...		Unknown
http://www.neon-toolkit.org/b...	priori	Unknown
http://www.neon-toolkit.org/b...	2009-12-07+10:07:22	Unknown
http://www.neon-toolkit.org/b...	104	Unknown

Host	Attribute
www.britishairways.com	http://www.britishairways.com/cms/s...
book.lufthansa.com	http://book.lufthansa.com:80/pl/Luft...
book.lufthansa.com	http://book.lufthansa.com:80/pl/Luft...
www.lufthansa.com	http://www.lufthansa.com/online/por...
www.britishairways.com	http://www.britishairways.com/travel...
itunes.apple.com	http://itunes.apple.com/WebObjects/...
video-stats.video.google.com	http://video-stats.video.google.com/...
faq.airfrance.fr	http://faq.airfrance.fr/b2c/AskQuesti...

Attribute	Value	Status
http://www.neon-toolkit.org/b...	mdaquin	Match
http://www.neon-toolkit.org/b...	passwd2	Match
http://www.neon-toolkit.org/b...	m.daquin%40open.ac.uk?login...	Match
http://www.neon-toolkit.org/b...	m.daquin@open.ac.uk	Match
http://www.neon-toolkit.org/b...	passwd524	Match
http://www.neon-toolkit.org/b...	me	Match
http://www.neon-toolkit.org/b...	mathieu+d'aquin	Match
http://www.neon-toolkit.org/b...	passwd2	Match
http://www.neon-toolkit.org/b...	passwd2	Match
http://www.neon-toolkit.org/b...	metm.daquin@virgin.net	Match
http://www.neon-toolkit.org/b...	passwd2	Match
http://www.neon-toolkit.org/b...	metm.daquin@virgin.net	Match

Host	Attribute	Status
www.alitalia.com	http://www.alitalia.com/...	
www.kayak.co.uk	http://www.kayak.co.uk/...	
statse.webtrendslive.com	http://statse.webtrendsli...	
book.lufthansa.com	http://book.lufthansa.co...	
book.lufthansa.com	http://book.lufthansa.co...	
book.lufthansa.com	http://book.lufthansa.co...	
www.lufthansa.com	http://www.lufthansa.co...	
www.lufthansa.com	http://www.lufthansa.co...	
www.lufthansa.com	http://www.lufthansa.co...	
www.britishairways.com	http://www.britishairway...	
statse.webtrendslive.com	http://statse.webtrendsli...	

Attribute	Value	Status
http://www.neon-toolkit.org/bu...	1	Irrelevant
http://www.neon-toolkit.org/bu...	2	Irrelevant
http://www.neon-toolkit.org/bu...	3	Irrelevant
http://www.neon-toolkit.org/bu...	6	Irrelevant

Fig. 2. Screenshot of the Data Transfer Log to User Profile mapping tool. On the right hand side is the data transfer log, showing different websites and the data they received. On the left hand side is the profile created from mapping this log (top), as well as suggestions for additional mappings (bottom).

3.3 Mapping to Personal Data

While the data extracted above only concerns data sent from the user to external websites, not all of it relates to personal information and it is not in this form easily interpretable as such. In order to extract from such a data transfer log *relevant data*, we built a tool that allows the user to easily identify personal information in it. Without going into the details (this tool is described in [6]), it provides mechanisms for the user to create mappings between the parameters used in particular websites (e.g., `http://qdos.com/signin#username`, `http://spreadsheets.google.com/ccc #email`) and attributes of a very simple model of the user profile (e.g., `UserName`, `e-mail`). The attributes in the profile might initially exist or might be created on demand, as required by the creation of a mapping. Once a mapping is created, the role of the tool is first to use it to populate the user profile with values from the data (e.g., `e-mail=m.daquin@open.ac.uk`). It also suggests additional mappings by looking, in the data transfer log, at where values already added to the profile appear.

From the data extracted for our experiment and using the interactive interface described above (see screenshot Figure 2), we built a profile made of 36 attributes and 598 values, creating 1,113 mappings to 184 different websites. Re-integrating such information into the RDF log data could allow for many different ways of studying and analyzing the user behavior [1]. Here, we focus on deriving from it models of observed trust in websites and data criticality.

4 Observed Trust in Websites and Criticality of Data

The data obtained from the tool above contains information about both the user profile and, through the mappings, about the websites to which each piece of information has been sent. This constitutes the basis of our model of trust in websites and criticality of data. First, we introduce some definitions concerning websites, domains and data pieces.

4.1 Basic Notions

We identify websites through their second level *domain* (SLD[7]) names (e.g., `google.com`, `sncf.fr`). The list of domains is automatically extracted from the data obtained in the previous step, using the URLs to which data was sent through HTTP requests. We call D the set of all the domains d_i in our dataset. In our experiment, there were 123 different domains that received data from the user profile.

We consider the notion of *data piece* to represent an element of information from the user profile, which was sent to one or more domains. Here, we use the attributes of the profile to correspond to data pieces (e.g., `passwd` or `e-mail` are data pieces). We call P the set of data pieces p_i present in our dataset. In our experiment, there were as many data pieces as attributes in the profile (i.e., 36).

Finally, we define two simple functions, to represent the list of data pieces received by a particular domain, and the list of domains to which a piece of data was sent, i.e.,

- $R(d_i) \subseteq P$ represents the set of data pieces received by the domain d_i
- $S(p_i) \subseteq D$ represents the set of domains to which the piece of data p_i was sent

[7] http://en.wikipedia.org/wiki/Second-level_domain

For example, in our experiment, $R(\texttt{lip6.fr}) = \{\texttt{username}, \texttt{passwd}, \texttt{hasReviewed}\}$ and $S(\texttt{city}) = \{\texttt{2o7.net}, \texttt{britishairways.com}, \texttt{ter-sncf.com}, \texttt{google.com}\}$, all this information being extracted directly from the Web activity logs and the mappings to the user profile.

4.2 Computing the Observed Trust in Domains and Data Criticality

Our goal here is, relying on the simple notion of data transfer defined above, to analyse the behavior of the user and derive what is expected to be his implicit trust relationship with the considered websites, and the correlated levels of criticality he seems to associate to each of the considered pieces of data. Crucially, these two notions are highly inter-dependent. Indeed, on the one hand, it is natural for an external observer to assess the trust somebody has in another agent based on the information he is prepared to disclose to this external agent. For example, if I consider my mobile phone number as a critical information, and disclose it to a particular website, this seems to indicate a high level of trust in this particular website. On the other hand, assessing the criticality of a piece of data can be done by considering how much this information is disclosed to external, varyingly trusted agents. The information about my screen resolution for instance might not be considered very critical, since I have provided it to many different website, most of them not very trusted.

On the basis of these simple intuitions, we define two functions, $T(d_i) \in [0..1]$ and $C(p_i) \in [0..1]$, representing the levels of observed trust in a domain d_i and of criticality of a piece of data p_i respectively. These measures are dependent on each other according to the two equations (1) and (2) below:

$$T(d_i) = max_{p_j \in R(d_i)}(C(p_j)) \tag{1}$$

$$C(p_i) = \frac{1}{1 + \sum_{d_j \in S(p_i)} 1 - T(d_j)} \tag{2}$$

Intuitively, (1) translates the idea that the level of trust associated with a domain d_i corresponds to the level of criticality of the most critical piece of data d_i has received. Equation (2) is slightly more complex. It is meant to give a high value of criticality to a piece of data p_i if p_i was sent only to a small number of highly trusted domains, and a low value if it has been sent to a high number of not trusted domains.

The most obvious problem with these measures is of course their interdependence. In practice, we consider them as sequences with the values of criticality $C(p_i)$ for each $p_i \in P$ at a time t calculated on the basis of the values of trust $T(d_j)$ for domains $d_j \in D$ at the time $t-1$. Using initial values of 0.5 for both trust and criticality, these measures converge to a stable state (with a precision of 0.0001) in 285 iterations on our dataset. The result is that each domain and each piece of data is associated with a level of observed trust and criticality respectively, which the user can then inspect to check to which extent it corresponds to his own intended, implicit model of trust and criticality. An interactive, visual tool to support the user in such a task is presented in the next section.

5 Visualizing and Interacting with Trust and Criticality Models to Detect Conflicts

Ultimately, the goal of computing the model of observed trust described above is to allow the user to explore it, getting informed about his apparent behavior, and compare this apparent behavior with his own view on trust and data criticality. In other terms, a method to visually explore and interact with the measures of trust and criticality is needed to get the benefit of the observation of Web activity back to the user.

5.1 Visualizing Sets of Measures

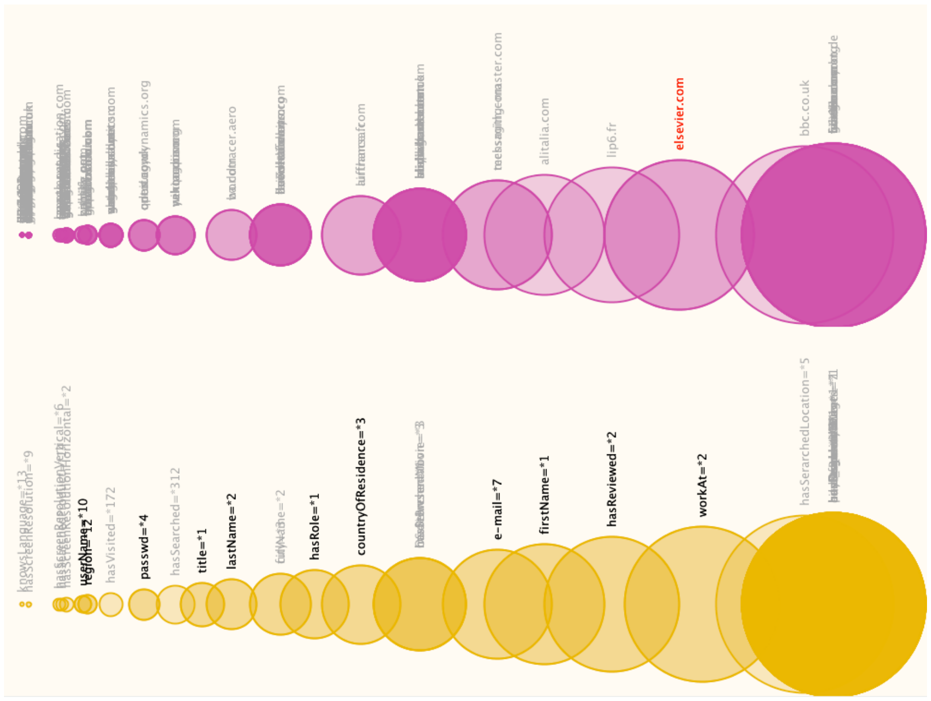


Fig. 3. Visualization of the observed trust in domains (top) and the observed data criticality (bottom). See http://people.kmi.open.ac.uk/mathieu/trustvisu.hml

While our model is relatively simple (a measure for each domain accessed and each piece of data considered), showing it in a way that provides meaningful information and interactions is not a trivial task. For this purpose, we developed a visualization technique to display a set of objects with a score between 0 and 1 (in our case, domain trust and data criticality). This representation shows each object as a 'bubble' along an axis (representing the score), with the size of each bubble also representing the score. Applied on our data, Figure 3 shows in the top half the visualization of the computed trust in domains (purple bubbles) and in the bottom half the computed criticality for the considered pieces of data (orange bubbles). While relatively simple, this visualization allows the user to quickly identify for example which are the most trusted domains and what is the relation between the values of criticality for different pieces of data (e.g., `first name` is less critical than `full name`, or `e-mail` is half as critical as `postal address`).

5.2 Exploring the Data

In addition to providing a simple representation for the measures constituting our model of observed trust in domains and data criticality, it is important to provide to the user ways to interact with the data, so that he can explore further the relation between websites' domains, the data they received, trust and criticality. Indeed, one important information hidden in our model concerns what information has been sent to which domains. Here, by selecting in the top panel the bubble corresponding to a given domain, the list of pieces of data it has received is highlighted in the bottom panel. In our example Figure 3, the domain `elsevier.com` was selected, showing that this domain has received data of very varying levels of criticality. In the same way by selecting a piece of data in the bottom panel, the domains which have received this particular piece of data would be highlighted.

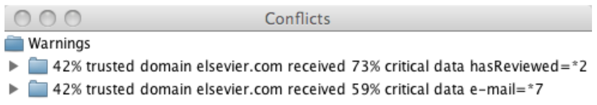
Such a simple way to explore the collected data is crucial to our approach to managing trust and privacy. Indeed, it allows the user to answer questions in a way that would not be possible without the appropriate monitoring of Web traffic, such as "Which websites know my e-mail address?" or "What does `google-analytics.com` know about me?". In a more abstract way, it also allows the user to explore his own behavior, by showing him the criticality of the data sent to particular websites, and what it says about the trust he, in appearance, is giving to them.

5.3 Interacting with the Model and Detecting Conflicts

One of the most important advantage of exposing the observed behavior of the user with respect to trust in domains and data criticality is that it gives him the ability to compare it to his intended behavior. In other terms, the user should be given the ability to disagree with the model, to try to correct it, and to detect fundamental conflicts between his own view and what the model can derive from the observed behavior. Indeed, it appears obvious that the computed model sometimes comes up with values which are fundamentally different from what the user would have expected, considering for example the information about his e-mail address as being not very critical and associating high values of trust to websites with relatively unclear privacy policies (e.g., `lip6.fr`).

To support the user in expressing this mismatch between his intended behavior and the observed model, our tool allows him to manually set the values of the trust in a domain or criticality of a piece of data. In practice, he can drag one of the bubbles into another place along the axis, fixing the value for the considered measure. While a manually set measure will not anymore be affected by the model described in Section 4, it will continue to impact on the trust and criticality of other, not modified domains and pieces of data. In other terms, as the user moves a domain or a piece of data around, the influence of this modification on the model can be directly seen through other bubbles moving in the same direction. For example, reducing the trust value for `elsevier.com` would directly affect the criticality associated with the pieces of data `elsevier.com` has received, and indirectly, the trust that is associated with other domains. As the visualization is updated dynamically, this is translated into a number of bubbles moving in both panels, following the movement of the one being selected and set manually. Interestingly, while this provides a way to interact with the model and understand the relation between trust and criticality, it can also be used to identify interesting correlations resulting in simultaneous movements, derived from indirect calculations. Indeed, moving for example the bubble corresponding to `last-name` in the bottom panel not only makes domain bubbles to move accordingly in the top panel, but also results in the `first-name` data piece being updated, showing a strong relationship between these two pieces of data and their exchange with various websites.

Using the mechanisms described above, the user, starting from the computed model translating the observed behavior, can build his intended model based his own view on trust and criticality. However, manually setting values for trust and criticality inevitably results in conflicts between this intended, declared model and the fundamental assumption underlying the computed model: That untrusted websites should not receive critical data. We therefore define and detect a conflict as a significant positive difference between the manually set value for the criticality of a piece of data and the manually set value for trust in a domain to which this piece of data was sent. More formally, we define the set C of conflicts in a user-corrected model as $C = \{(d_i, p_j) | d_i \in MD \wedge p_j \in MP \wedge p_j \in R(d_i) \wedge C_m(p_j) - T_m(d_i) > \epsilon\}$, where MD is the set of domains for which the trust is manually set, MP the set of pieces of data for which the criticality is manually set, $C_m(p_j)$ is the manually set value of criticality for a piece of data p_j, $T_m(d_i)$ is the manually set value of trust for the domain d_i and ϵ is a given constant above which the level of conflict is considered significant. Figure 4 shows two examples of detected conflicts, as they appear to the user and ranked according to the value of the difference $C_m(p_j) - T_m(d_i)$. In these examples, the user has indicated that the domain `elsevier.com` was less trusted than observed, and that the pieces of data `hasReviewed` (journal articles that were reviewed by the user) and `e-mail` were more critical then computed by the model, resulting in the displayed conflicts.



```
42% trusted domain elsevier.com received 73% critical data hasReviewed=*2
42% trusted domain elsevier.com received 59% critical data e-mail=*7
```

Fig. 4. Examples of conflicts detected after the user had modified the model.

6 Going Further: Semantically Enriching Traces of Web Activity for Personal Privacy Policies

As shown in the previous sections, tools keeping track of the user's Web activities can make emerge models of the observed trust in websites and criticality of the data. Exposing these models to the user and allowing him to interact with them provides the user with a way to better control his own privacy, by informing him of possibly unintended behaviors (i.e., conflicts) so that he can act on them. Indeed, in our experiment, many of such conflicts arise and elements that emerged from the data appeared sometimes very surprising (e.g., the amount of critical information being unknowingly sent to `google-analytics.com` as a result of other websites using this tool).

In building the above described techniques, the use of semantic technologies appears very useful, in order to provide flexible models of Web activity logs, which are mapped onto users' personal information. In addition, it provides us with the possibility to enrich the produced models with external data, to interlink it, so that richer ways to explore the user's own activity logs and trust models can be used. One simple example would be to integrate for each domain the corresponding 'semantified' information from its registrar (i.e., using the *whois* utility). Such information describes the people/companies who own the domain with contact information and addresses. By relating it to geo-localization data (e.g., the geonames dataset[8]), pieces of data could be reconnected not only to where they were sent on the Web, but could also to more concrete elements, allowing for example to explore the implications in terms of the privacy laws applying to different Web interactions.

One of the obvious next steps for this work is to provide the user with support not only to understand his own behavior, but also to directly act on it. It would be for example easy to adapt the logging tool described in Section 3.1 to implement user-defined rules instructing the tool to reject any attempt to send data to particularly untrusted website or to alert the user before sending data of certain levels of criticality to unknown websites. Here as well, the use of semantic technologies would appear crucial in order that these rules are defined with the appropriate levels of expressivity and granularity. This would allow for example a user to declare that no data above a certain level of criticality should be sent

[8] http://geonames.org

to any analytics website, while defining clearly the conditions for a domain to be classified as an analytics website and exploiting information pulled from, e.g., DBPedia[9] to enrich the information known about a given domain so that it is sufficient to test these conditions.

7 Conclusion

In this paper, we showed how tracking the data exchanges for an individual user can help in defining personal models of trust in websites and data criticality, providing the first elements of a platform to better monitor and control the exchange of personal information on the Web. In the short term, the realization of such a platform raises many technical challenges, including the ability to process such amounts of data in real time, without impacting on the user's Web experience. In the longer term, many refinements are currently being investigated to provide more support to the user in exploring and controlling their own data exchange, which implies relating the user data with external, semantic data. Another interesting further step consists in integrating a social aspect to the management of personal privacy policies, allowing users to share their experience with particular domains. This can include sharing trust information concerning particular websites, but also making the mappings between websites' attributes and personal information available for others to reuse.

In addition, this work on using Web activity logs to better understand the behavior of Web users with regards to privacy and trust can be seen as considering one aspect of a broader domain of study, observing the full range of activities on the Web to derive useful models, both for the users themselves, but also for researchers in technological and non-technological areas. As such, it appears important to open similar experiments to the one described in this paper to larger groups of users, preferably with different backgrounds, interests and uses of the Web.

References

1. d'Aquin, M., Elahi, S., Motta, E.: Personal monitoring of web information exchange: Towards web lifelogging. In: Web Science Conference (poster presentation, to appear). (2010)
2. O'Hara, K.: Trust – from Socrates to Spin. Icon Books (2004)
3. Hoffman, R.R., Lee, J.D., Woods, D.D., Shadbolt, N., Miller, J., Bradshaw, J.M.: The dynamics of trust in cyberdomains. IEEE Intelligent Systems **24** (2009) 5–11
4. Golbeck, J.A.: Computing and applying trust in web-based social networks. PhD thesis, College Park, MD, USA (2005) Chair-Hendler, James.
5. Rowe, M., Butters, J.: Assessing trust: Contextual accountability. In: First Workshop on Trust and Privacy on the Social and Semantic Web. (2009)
6. Salman Elahi, M.d., Motta, E.: Who want a piece of me? reconstructing a user profile from personal web activity logs. In: International ESWC Workshop on Linking of User Profiles and Applications in the Social Semantic Web. (2010)

[9] http://dbpedia.org

Provenance Aware Linked Sensor Data

Harshal Patni, Satya S. Sahoo, Cory Henson, Amit Sheth

Kno.e.sis Center, Computer Science and Engineering Department, Wright State University, Dayton, OH 45435 USA
{harshal, satya, cory, amit}@knoesis.org

Abstract. Provenance, from the French word "*provenir*", describes the lineage or history of a data entity. Provenance is critical information in the sensors domain to identify a sensor and analyze the observation data over time and geographical space. In this paper, we present a framework to model and query the provenance information associated with the sensor data exposed as part of the Web of Data using the Linked Open Data conventions. This is accomplished by developing an ontology-driven provenance management infrastructure that includes a representation model and query infrastructure. This provenance infrastructure, called *Sensor* Provenance Management System (PMS), is underpinned by a domain specific provenance ontology called Sensor Provenance (SP) ontology. The SP ontology extends the Provenir upper level provenance ontology to model domain-specific provenance in the sensor domain. In this paper, we describe the implementation of the Sensor PMS for provenance tracking in the Linked Sensor Data.

Keywords: Provenance Management Framework, provenir ontology, Provenance, Lineage, Linked Data, Semantic Sensor Web, Sensor Data, Sensor Web Enablement, Dataset Generation, Resource Description Framework (RDF)

1. INTRODUCTION

The first North American blizzard of 2010 was tracked from the state of California to Arizona, through northern Mexico, and across the continental United States. The storm produced historic snowfall levels in the Mid-Atlantic States, as well as extensive flooding and landslides in Mexico. During this time, a number of weather stations collected data from thousands of sensors deployed in the United States. Semantic Sensor Web[1] proposes to annotate this sensor data with semantic metadata to provide contextual information essential for situational awareness. Such semantic metadata data can be used to answer aggregate queries spanning both temporal and geographical areas.

Let us consider the following scenario. *We are interested in finding all the sensors which have observations related to a blizzard of interest. In order to accomplish this task, we would need to know the properties associated with a phenomenon to be classified as a blizzard, the time period for which the blizzard was active, the location where the blizzard occurred, and sensors deployed in this location during this time period.*

[1] http://wiki.knoesis.org/index.php/SSW

This is an example of a sensor discovery query. Sensor discovery has been identified as a top-priority use case by the W3C Semantic Sensor Network Incubator Group[2], which is tasked with development of sensor ontology. In the sensors domain, the capabilities of the sensor, observation location (spatial parameter), time of observation (temporal parameter), and phenomenon measurement (domain parameter) are important to answer discovery queries. This data related to the sensor is the provenance metadata about the sensor. Provenance describes the history or the lineage of an entity and is derived from the French word "*provenir*" meaning "to come from". Provenance information enables applications to answer the "what", "where", "why", "who", "which", "when", and "how" queries to accurately interpret and process data entities.

Provenance has been studied from multiple perspectives, including (a) workflow provenance and (b) database provenance as discussed in Tan [1]. Workflow provenance represents "the entire history of the derivation of the final output of" [1] a workflow. Davidson et al. [2] addresses issues related to provenance in workflow systems. In contrast, database provenance refers to the process of tracing and recording the origins of data and its movement between databases [3]. In Sahoo et al. [4], we introduced the notion of semantic provenance to define provenance information that incorporates domain semantics to closely reflect the knowledge of an application domain.

In this paper, we use the observations from the 20,000 sensors within the United States (Figure 1) in the context of a blizzard as a running example.

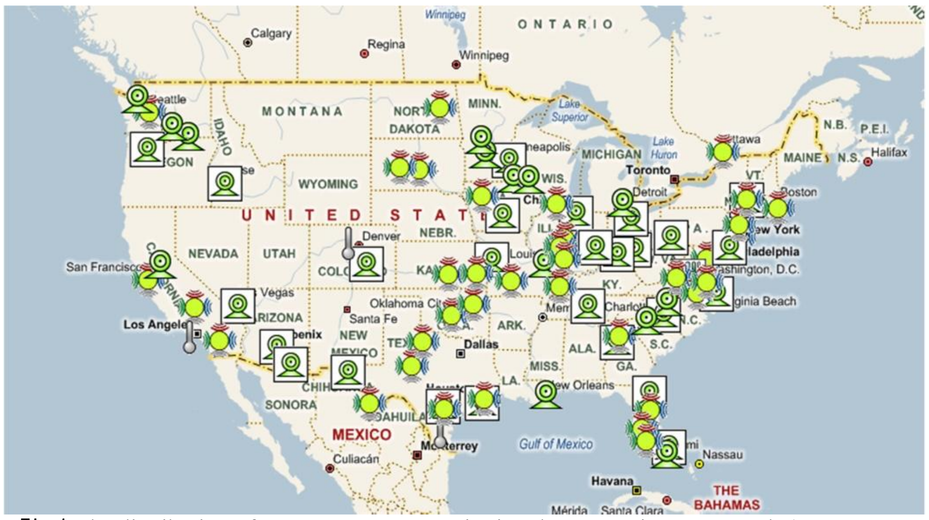


Fig.1.The distribution of 20,000 Sensors constituting the Semantic Sensor Web (SensorMap Image [5])

We use the definition of a blizzard provided by the NOAA[3], which describes it as:

BLIZZARD = High WindSpeed (exceeding 35 mph) AND Snow Precipitation AND Low Visibility (less than ¼ mile), for at minimum 3 hours.

Fig.2. Blizzard Composition

[2] http://www.w3.org/2005/Incubator/ssn/wiki/Main_Page

[3] http://www.noaa.gov/

A blizzard exists if the above conditions hold true for at least 3 hours within some geospatial region. Hence, the provenance of sensor observations describing the geospatial information of the sensors that record the observations, the time stamp of the observations, and the attributes of the sensor itself (for example, a motion sensor is not useful in context of a blizzard) are important for a sensor discovery query.
With a view of capturing the provenance information related to a sensor, the main objective of this paper is to implement a Sensor Provenance Management System (Sensor PMS). In this paper, we describe the creation of this infrastructure using the theoretical underpinning of the Provenance Management Framework (PMF) [4]. The key contributions of the paper are described below:

1. Implementing Sensor PMS to track provenance in the linked sensor data
2. Developing a domain specific ontology for Sensor PMS called Sensor Provenance (SP) ontology. The SP ontology uses concepts within the Provenir upper level ontology defined in PMF [4] to add provenance information within the sensors domain.
3. An evaluation of the Sensor PMS capabilities to answer provenance queries over the sensor datasets generated is provided.

The rest of the paper is organized as follows: Section 2 discusses background concepts. In section 3, we describe current infrastructure for generating sensor datasets and section 4 discusses the sensor datasets generated. Section 5 integrates the current infrastructure described in section 3 with the provenance management system and describes the architecture of Sensor PMS. Section 6 introduces the SP ontology and section 7 discusses the kind of queries that can be answered with the help of provenance information. Section 8 gives related work and section 9 concludes with summary and future work.

2. Background

In this section, we describe the resources used in our work including the Sensor ontology and the Linked Open Data initiative.

2.1 Ontology Model of Sensor Data – In computer science and information science, ontology is a formal representation of the knowledge by a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to describe the domain. [6] Our sensors ontology uses the concepts within the O&M standard to define sensor observations. Within the O&M standard, an observation (*om:Observation*) is defined as an *act of observing a property or phenomenon, with the goal of producing an estimate of the value of the property*, and a feature (*om:Feature*) is defined as an *abstraction of real world phenomenon*. (Note: *om* is used as a prefix for Observations and Measurements). The major properties of an observation include feature of interest (*om:featureOfInterest*), observed property (*om:observedProperty*), sampling time (*om:samplingTime*), result (*om:result*), and procedure (*om:procedure*). Often these properties can be complex entities that may be defined in an external document. For example, *om:FeatureOfInterest* could refer to any real-world entity such as a coverage region, vehicle, or weather-storm, and *om:Procedure* often refers to a sensor or system of sensors defined within a SensorML[4] document. Therefore, these properties are better described as relationships of an observation. Concepts described above and their relationships within the sensor ontology can be found in figure 2. The Sensor ontology can be found at [7]. Section 5 extends the Sensor Ontology with provenance related concepts found in the Provenir upper level ontology defined in the Provenance Management Framework (PMF) [4].

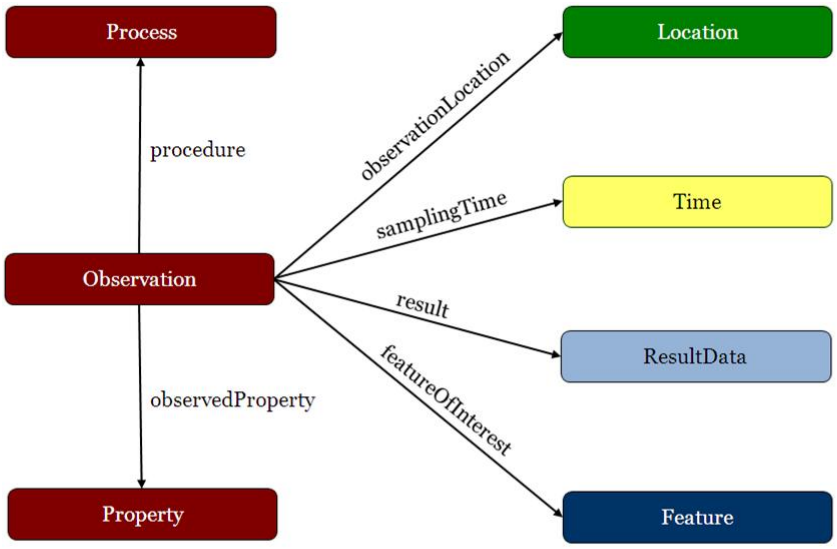


Fig.2. Concepts and their relationships within the Sensor Ontology

2.2 Semantic Web –The Semantic Web is an evolving development of the World Wide Web[5] derived from the World Wide Web consortium (W3C)[6] in which the meaning of information and services on the web is defined, making it possible for the web to understand and satisfy the request of people and machines that use the web content. [8] Resource Description Framework (RDF) is a publishing language within the Semantic Web, specially designed for data. RDF has now come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax formats. [9]. It is also a standard model for data interchange on the web. [10] SPARQL[7] is a protocol and a query language for semantic web data sources. [8] In its usage, SPARQL is a syntactically-SQL-like language for querying RDF graphs. [11] Since Semantic Web is not just about putting data on the web but also linking the data, Linked Data is used to connect the Semantic Web[8]. Wikipedia defines Linked Data as "*a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF*." [12] Linked Data is a large and growing collection of interlinked public datasets encoded in RDF spanning diverse areas such as: life sciences, nature, science, geography and entertainment.

[4] http://www.opengeospatial.org/standards/sensorml
[5] http://en.wikipedia.org/wiki/World_Wide_Web
[6] http://www.w3.org/
[7] http://www.w3.org/TR/rdf-sparql-query/
[8] http://www.w3.org/DesignIssues/LinkedData.html

3. Current Infrastructure

The lifespan of sensor data starts as observable properties of objects and events in the real-world which are detected by sensors through observation. These observation values are then encoded in several formats of varying degrees of expressivity, as needed by applications that may utilize the data. The data generation workflow is comprised of four main parts, as shown in figure 3. The workflow begins with sensors deployed across the United States measuring environmental phenomena. Observations generated from these sensors are aggregated at MesoWest [13] which provides access to past sensor observations encoded as comma separated numerical values. These sensor observations are then converted to Observations and Measurements (O&M). O&M is an encoding standard and a technical framework that defines an abstract model and an XML schema encoding for sensor descriptions and observations. It is one of OGC[9] Sensor Web Enablement (SWE)[10] suite of standards that is widely accepted within the sensors community for encoding sensor observations. [14] In order to add semantics to the sensor descriptions and observations the O&M is converted to RDF. O&M is converted to RDF using the *O&M2RDF-Converter API* described in [15]. Two RDF datasets, *LinkedSensorData* and *LinkedObservationData* containing over a billion triples were generated. The datasets are described in the section 4. The RDF generated is then stored in a Virtuoso RDF knowledgebase [16]. The RDF datasets are made available on the Linked Open Data Cloud to provide public access. The data generation workflow is the main component of the Provenance Capture phase discussed in Section 5.

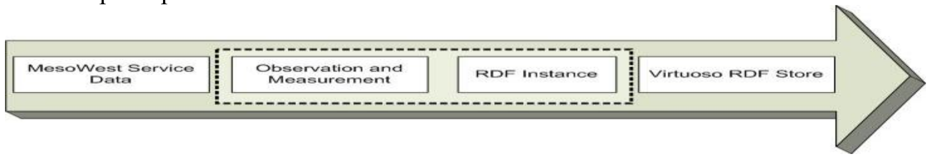


Fig.3. Data Generation Workflow. The O&M to RDF conversion (dotted portion) forms the main part of the workflow that uses the *O&M2RDF-CONVERTER API*.

3.1 Phase 1 – The first phase is comprised of querying MesoWest [13] for observational data and parsing the result. MesoWest provides a service to access past sensor data and returns an HTML page with the observational values encoded within a comma-separated list. The resulting HTML page is then parsed to extract the sensor observations.

3.2 Phase 2 – The second phase consists of converting the raw textual data retrieved from MesoWest into O&M. The sensor observations parsed from the HTML page in phase 1 are fed to an XML parser. We used the SAX (Simple API for XML) parser[11] to generate the O&M. Here we also query GeoNames [17] with the sensor coordinates to get GeoNames location that is closest to the sensor. The O&M generated in this phase is the input for the *O&M2RDF-Converter API*.

[9] http://www.opengeospatial.org/
[10] http://www.opengeospatial.org/projects/groups/sensorweb
[11] http://www.saxproject.org/

3.3 Phase 3 – The third phase consists of converting sensor observations encoded in O&M to RDF. Since both O&M and RDF have XML syntax, XSLT is used to convert O&M to RDF. XSLT is a language for transforming XML documents into other XML documents [18]. The XSLT performs the conversion for our *O&M2RDF-Converter* API.

3.4 Phase 4 - The fourth phase consists of storing the RDF in Virtuoso RDF store. Virtuoso RDF is a native triple store available in both open source and commercial licenses. It provides command line loaders, a connection API, support for SPARQL and web server to perform SPARQL queries and uploading of data over HTTP. It has been tested to scale up to a billion triple. A more detailed description of the data generation workflow can be found in [15].

4. Sensor Dataset Description

The data generation workflow described in section 3 lead to the generation of 2 RDF datasets *LinkedSensorData* and *LinkedObservationData* containing over a billion triples described in detail below.

4.1 Linked Sensor Data - LinkedSensorData is an RDF dataset containing expressive descriptions of ~20,000 weather stations in the United States. The data originated at MesoWest, a project within the Department of Meteorology at the University of Utah that has been aggregating weather data since 2002. [13] On average, there are five sensors per weather station measuring phenomena such as temperature, visibility, precipitation, pressure, wind speed, humidity, etc. In addition to location attributes such as latitude, longitude, and elevation, there are links to locations in Geonames [17] near the weather station. The distance from the Geonames location to the weather station is also provided. The data set also contains links to the most current observation for each weather station provided by MesoWest [13]. This sensors description dataset is now part of the LOD.

4.2 Linked Observation Data - LinkedObservationData is an RDF dataset containing expressive descriptions of hurricane and blizzard observations in the United States. The data again originated at MesoWest. [13] The observations collected include measurements of phenomena such as temperature, visibility, precipitation, pressure, wind speed, humidity, etc. The weather station's observations also include the unit of measurement for each of these phenomena as well as the time instant at which the measurements were taken. The dataset includes observations within the entire United States during the time periods that several major storms were active -- including Hurricane Katrina, Ike, Bill, Bertha, Wilma, Charley, Gustav, and a major blizzard in Nevada in 2002. These observations are generated by weather stations described in the LinkedSensorData dataset introduced above. Currently, this dataset contains more than a billion triples. The RDF dataset for each of the above storms is available for download in gzip format at [19]. The statistics for each of the storms can also be found at [19]

5. Sensor Provenance Management System

The Sensor PMS infrastructure uses the data generation workflow described above (section 3) and addresses three aspects of provenance management as identified by [20]. See Figure 4 for an architecture of Sensor PMS.

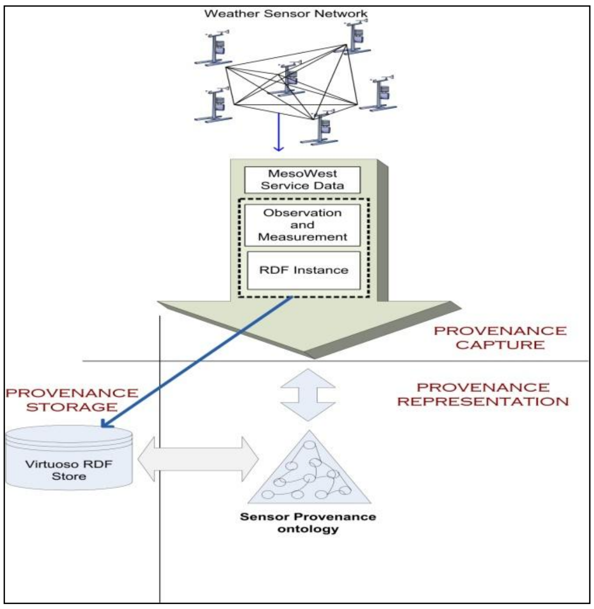


Fig.4. The architecture of the *Sensor* PMS addressing
Three aspects of provenance management

1. Provenance Capture – The provenance information associated with the sensor is captured within the data workflow as described in section 3. The time related information (temporal parameter) is obtained from MesoWest [13] and location related information (spatial parameter) is obtained by querying GeoNames [17] with the sensor coordinates.
2. Provenance Representation – The Sensor Provenance ontology (SP) is used to model the provenance information related to the sensor. The SP ontology extends the Provenir upper level provenance ontology defined in PMF [4] to support interoperability with provenance ontology in different domains.
3. Provenance Storage – The provenance information is stored in the Virtuoso RDF store. Virtuoso RDF is an open source triple store provided by OpenLink Software.[16] The Virtuoso RDF store currently contains over a billion triples of sensor observational data. Virtuoso RDF provides a SPARQL endpoint to query these dataset discussed in section 4, which can be found at [21]. More information about querying the dataset can be found at [19].

6. Sensor Provenance Ontology

In this section we discuss the Sensor Provenance Ontology that forms the key component of the Sensor PMS. As discussed above, provenance information includes the location of the sensor, the time when the observations were taken by the sensor and the sensor observation values. Since SP ontology extends the provenir ontology, we discuss the provenir ontology in section 6.1 followed by SP ontology in section 6.2

6.1 Provenir Ontology - Provenir ontology is a common provenance model which forms the core component of the provenance management framework. [4] This modular framework forms a scalable and flexible approach to provenance modeling that can be adapted to the specific requirement of different domains. Use of Provenir ontology as the reference model to built domain-specific provenance ontologies ensures (a) common modeling approach, (b) conceptual clarity of provenance terms, and (c) use of design patterns for consistent provenance modeling

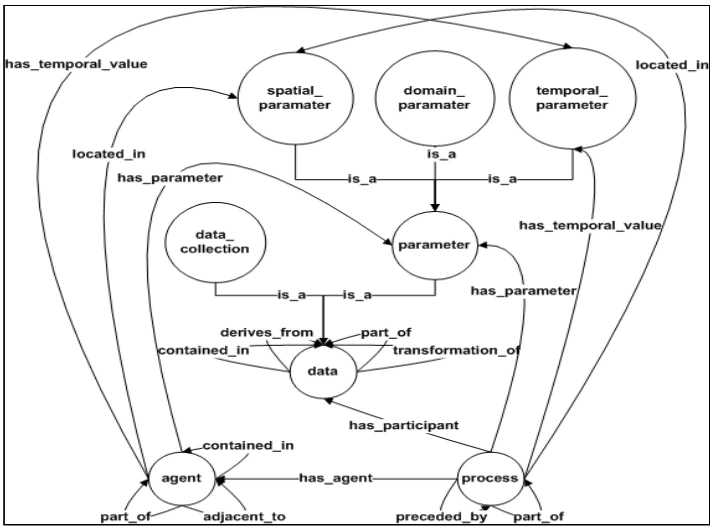


Fig.5. Provenir Upper Level Ontology [4]

The ontology defines three base classes *data*, *agent* and *process* using the well defined, primitive concepts of *occurent* and *continuant*. [22] *Continuant* is defined as "entities which endure, or continue to exist, through time while undergoing different sorts of changes, including changes of place" [22] while *Occurrent* is defined as "entities that unfold themselves in successive temporal phases". [22]. The two base classes, *data* and *agent* are defined as specialization (sub-class) of *continuant* class while the third base class *process* is a synonym of *occurent*. The *data* class has two sub-classes, *data_collection* -- that represents the datasets that undergo modification during an experiment -- and *parameter* -- that influences the execution of an experiment. The *parameter* class has three sub-classes representing the spatial, temporal, and thematic (domain-specific) dimensions, namely *spatial_parameter*, *temporal_parameter*, and *domain_parameter*. Instead of defining a new set of properties, the ontology reuses and adapts properties defined in the Relation ontology (RO)[12] from the Open Biomedical Ontologies (OBO) Foundry[13] such as *part_of*, *contained_in*, *preceded_by*, and *has_participant*. The Provenir ontology is defined using OWL-DL[14] that is complaint with the DL profile of OWL2[15], with an expressivity of

[12] http://www.obofoundry.org/ro/
[13] http://www.obofoundry.org/
[14] http://www.w3.org/TR/owl-features/

$\mathcal{ALCH}$; further details of the ontology can be found at [23]. Figure 5 shows the Provenir ontology schema obtained from [4].

5.2 Sensor Ontology - Extending Provenir Ontology

The Provenir ontology has been extended to create the Sensor ontology that models the domain-specific provenance information for the sensor domain. The Sensor ontology extends the relevant Provenir ontology terms using the *rdfs:subClassOf* and *rdfs:subPropertyOf* relationships to create appropriate classes and properties. For example, the *sensor:ResultData* (representing the observation value) is a subclass of *provenir:data_collection*, the *sensor:Location* class (representing the geographical location) is defined as a subclass of *provenir:spatial_parameter*. Similarly, *sensor:samplingTime* is defined as a subproperty of *provenir:has_temporal_value*.
The sensor ontology has been defined in OWL-DL and consists of 89 classes, 53 properties with a DL expressivity of $\mathcal{ALEHIF}+(\mathcal{D})$. By extending the Provenir ontology, the sensor ontology ensures coherent modeling of concepts, consistent use of provenance terminology, and compatibility with other existing domain-specific provenance ontologies. For example, the Trident ontology extends the Provenir ontology to model provenance information in the Neptune oceanography project [24]. In the next section, we describe the queries that utilize the provenance information modeled in the sensor ontology.

7. Provenance Queries

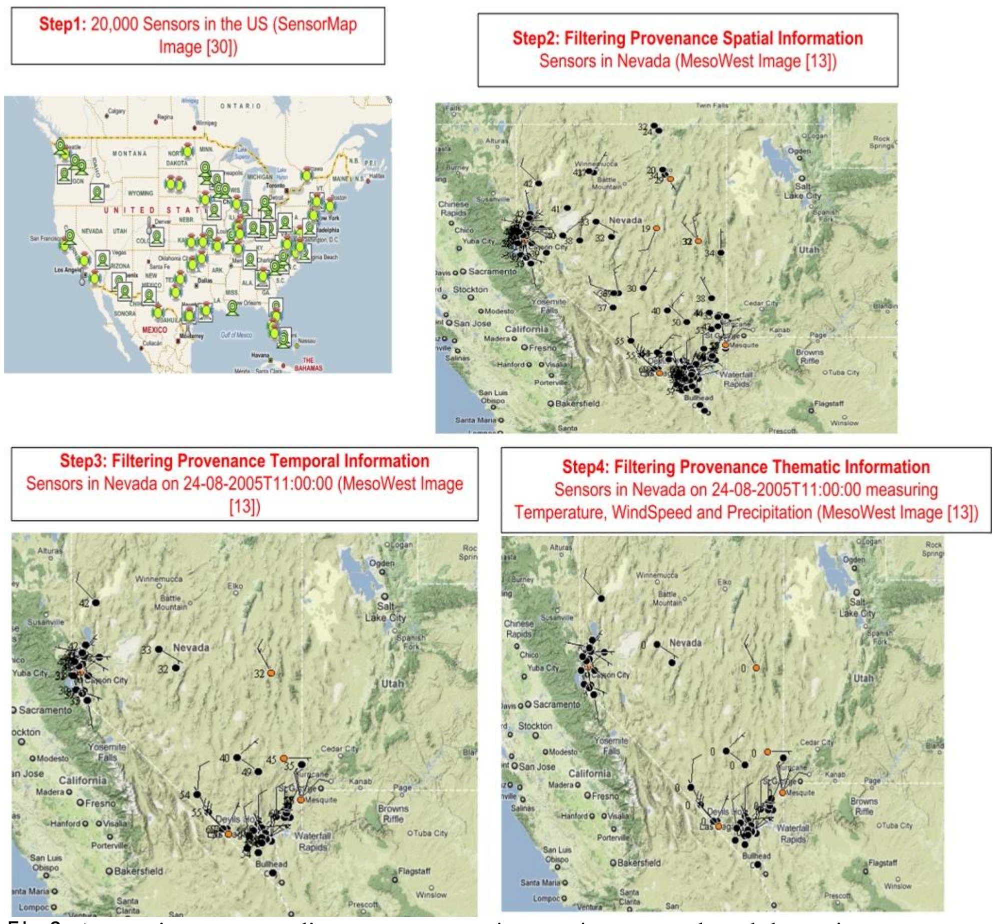
Two classes of Provenance queries have been categorized by PMF [4]. Corresponding queries in the sensors domain that could not be answered without provenance information have been provided.

1. Query for provenance metadata: Given a data entity, this category of queries returns the complete set of provenance information associated with a data entity. Example: "*Given an observation value, give me the provenance information about the all the sensors that recorded this observation*"

```
SELECT ?sensor ?ID ?geonamesLocation ?geonamesDistance
            ?geonamesDistanceMeasure ?latitude ?longitude
            ?observedProperty ?XSDTime
WHERE
   {?sensor om-owl:generatedObservation ?generatedObservation .
    ?generatedObservation om-owl:observedProperty ?observedProperty .
    ?generatedObservation om-owl:result ?measureData .
    ?measureData om-owl:floatValue ?value .
    FILTER(?value = "78.0"^^xsd:float) .
    ?generatedObservation om-owl:samplingTime ?timeInstant .
    ?timeInstant owl-time:inXSDDateTime ?XSDTime .
    ?sensor om-owl:ID ?ID .
    ?sensor om-owl:hasLocatedNearRel ?locatedNear .
    ?locatedNear om-owl:hasLocation ?geonamesLocation .
    ?locatedNear om-owl:distance ?geonamesDistance .
    ?locatedNear om-owl:distanceUOM ?geonamesDistanceMeasure .
    ?sensor om-owl:processLocation ?sensorLocation .
    ?sensorLocation wgs84:lat ?latitude .
    ?sensorLocation wgs84:long ?longitude .
    }
```

[15] http://www.w3.org/TR/owl2-profiles/

2. Query for data using provenance information: An opposite perspective to the first category of query is, given a set of constraints defined over provenance information retrieve a set of data entities satisfying some set of constraints. Example: *"Find all the sensors which have observations related to a blizzard occurring in Nevada on 24th August 2005 at 11 AM"*
   To solve this sensor discover query, provenance information describing the spatio-temporal and thematic aspects of sensor observations and sensors can be analyzed. Figure 6 describes the multiple steps followed in identifying the appropriate sensor. In Step 1, sensors located in the "Nevada" region are identified (from a pool of 20,000 sensors located across the United State). In Step 2, the sensors that were active during the blizzard are identified, and finally in Step 3 provenance information describing the capabilities of a sensor help identify the observations that are relevant for the blizzard under study (for example, a wind speed sensor is considered relevant while a motion sensor is not considered relevant.)


Step1: 20,000 Sensors in the US (SensorMap Image [30])
Step2: Filtering Provenance Spatial Information
Sensors in Nevada (MesoWest Image [13])
Step3: Filtering Provenance Temporal Information
Sensors in Nevada on 24-08-2005T11:00:00 (MesoWest Image [13])
Step4: Filtering Provenance Thematic Information
Sensors in Nevada on 24-08-2005T11:00:00 measuring Temperature, WindSpeed and Precipitation (MesoWest Image [13])
Fig.6. Answering a sensor-discovery query using spatio-temporal, and thematic provenance information

8. Related Work

Although this is the first attempt to develop an infrastructure for Sensor Provenance Management, there have been successful attempts to do the same in the domain of e-science. Within the sensors domain, provenance has been addressed from the storage point of view.

Provenance management within the eScience community has primarily been addressed in the context of workflow engines [25] while provenance management issues have been surveyed by Simmhan et al. [26]. The database community has also addressed the issue of provenance and defined various types of provenance, for example "why provenance" [27] and "where provenance" [27]. A detailed comparison of PMF (that underpins the *Sensor* PMS) with both workflow and database provenance is presented in [4].

The Semantic Provenance Capture in Data Ingest Systems (SPCDIS) [28] is an example of eScience project with dedicated infrastructure for provenance management. In contrast to the *Sensor* PMS, the SPCDIS project uses the proof markup language (PML) [29] to capture provenance information. The Inference Web toolkit [29] features a set of tools to generate, register and search proofs encoded in PML. Both *Sensor* PMS and the SPCDIS have common objectives but use different approaches to achieve them, specifically the *Sensor* PMS uses an ontology-driven approach with robust query infrastructure for provenance management.

In the Sensors community, Ledlie et al. [30] show how provenance addresses the naming and indexing issues related to sensor data storage. Park et al. [31] explore the need for data provenance in *Sensornet Republishing,* a process of transforming online sensor data and sharing the filtered, aggregated, or improved data with others.

9. Conclusion

This paper introduces an in-use ontology-driven provenance management infrastructure for Sensor data called *Sensor* PMS. We have developed a domain specific sensor provenance ontology by extending the provenir ontology. Due to this extension, SP ontology can interoperate with other domain-specific provenance ontologies to facilitate sharing and integration of provenance information from different domains and projects. We also show how provenance information can help answer complex queries within the sensors domain.

Acknowledgments. This work is funded in part by NIH RO1 Grant# 1R01HL087795-01A1 The Dayton Area Graduate Studies Institute (DAGSI), AFRL/DAGSI Research Topic SN08-8: "Architectures for Secure Semantic Sensor Networks for Multi-Layered Sensing.".

References

[1] W. C. Tan. Provenance in Databases: Past, Current, and Future. *IEEE Data Engineering Bulletin*, 30(4):3–12, Dec. 2007.

[2] S. B. Davidson, S. C. Boulakia, A. Eyal, B.Lud¨ascher, T. M. McPhillips, S. Bowers, M. K.Anand, and J. Freire. Provenance in Scientific Workflow Systems. *IEEE Data Engineering Bulletin*, 30(4):44–50, Dec. 2007.

[3] P. Buneman, S. Khanna, and W. C. Tan. Data Provenance: Some Basic Issues. In *Proceedings of the 20th Conference on Foundations of Software Technology and Theoretical Computer Science (FST TCS)*. Springer, Dec. 2000.
[4] S.S. Sahoo, R.S. Barga, J. Goldstein, A.P. Sheth, K. Thirunarayan, "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.
[5] SensorMap. http://atom.research.microsoft.com/sensewebv3/sensormap/, Retrieved March 22 2010
[6] Wikipedia Article on Ontology. http://en.wikipedia.org/wiki/Ontology_(information_science), Retrieved March 21 2010
[7] Sensor Data Ontology Model. http://knoesis.wright.edu/research/semsci/application_domain/sem_sensor/ont/sensor-observation.owl, Retrieved March 21 2010
[8] Semantic Web Wikipedia. http://en.wikipedia.org/wiki/Semantic_Web, Retrieved March 15 2010
[9] Wikipedia Article on RDF. http://en.wikipedia.org/wiki/Resource_Description_Framework, Retrieved March 19 2010
[10] Resource Description Framework. http://www.w3.org/RDF/ Retrieved March 15 2010
[11] SPARQL Protocol and Language: Frequently Asked Questions. http://www.thefigtrees.net/lee/sw/sparql-faq#what-is, Retrieved March 15 2010
[12] Linked Open Data Cloud, http://linkeddata.org/, Retrieved March 20 2010
[13] MesoWest. http://mesowest.utah.edu/index.html, Retrieved March 20 2010
[14] Observation and Measurements (O&M). http://www.opengeospatial.org/standards/om, Retrieved March 18 2010
[15] H. Patni, C. Henson, A. Sheth, 'Linked Sensor Data,' In: Proceedings of 2010 International Symposium on Collaborative Technologies and Systems (CTS 2010), Chicago, IL, May 17-21, 2010.
[16] Openlink Software. http://www.openlinksw.com/, Retrieved March 12 2010
[17] GeoNames. http://www.geonames.org/, Retrieved March 12 2010
[18] XSLT. http://www.w3.org/TR/xslt, Retrieved March 12 2010
[19] SSW Dataset. http://wiki.knoesis.org/index.php/SSW_Datasets, Retrieved March 12 2010
[20] S. S. Sahoo, D. B. Weatherly, R. Mutharaju, P. Anantharam, A. P. Sheth, R. L. Tarleton, "Ontology-Driven Provenance Management in eScience: An Application in Parasite Research." OTM Conferences (2) 2009: 992-1009
[21] Virtuoso SPARQL endpoint. http://harp.cs.wright.edu:8890/sparql, Retrieved March 14 2010
[22] B. Smith, W. Ceusters, B. Klagges, J. Kohler, A. Kumar, J. Lomax, et al., "Relations in biomedical ontologies." Genome Biol 2005;6(5):R46.
[23] Provenir Ontology. http://knoesis.wright.edu/library/ontologies/provenir/provenir.owl, Retrieved March 13 2010
[24] S. S. Sahoo, A. Sheth, "Provenir ontology: Towards a Framework for eScience Provenance Management", Microsoft eScience Workshop, USA, Oct 2009
[25] Provenance Challenge Wiki. http://twiki.ipaw.info/bin/view/Challenge/WebHome, Retrieved March 11 2010
[26] Y.L. Simmhan, B. Plale, D. Gannon, "A survey of data provenance in e-science" SIGMOD Rec. 2005;34(3):31 - 36
[27] P. Buneman, S. Khanna, W.C. Tan, "Why and Where: A Characterization of Data Provenance." In: 8th International Conference on Database Theory; 2001; 2001. p. 316 - 330
[28] SPCDIS. http://spcdis.hao.ucar.edu/, Retrieved March 11 2010
[29] Inference Web. http://iw.stanford.edu/2.0/, Retrieved March 11 2010
[30] J. Ledlie, C. Ng, D. A. Holland, K.-K. Muniswamy-Reddy, U. Braun, and M. Seltzer. "Provenance-aware sensor data storage." In *NetDB 2005*, April 2005.
[31] U. Park, J. Heidemann. "Provenance in Sensornet Republishing." In Proceedings of the 2nd International Provenance and Annotation Workshop , pp. 208-292. Salt Lake City, Utah, USA, Springer-Verlag. June, 2008

Tracing the Provenance of Object-Oriented Computations on RDF Data

Matthias Quasthoff, Christoph Meinel

Hasso Plattner Institute, University of Potsdam
{matthias.quasthoff, christoph.meinel}@hpi.uni-potsdam.de

Abstract. This paper presents a new method for tracing the provenance of RDF data within object-oriented computations. The proposed approach eliminates the burden of manually keeping track of data provenance from software developers. Using object triple mapping, the source data items used for generating result data can be identified efficiently. The integration into an existing object triple mapping framework shows the feasibility of the approach. The work presented will help developing compliant RDF-enabled applications using object-oriented programming and may support the efforts getting the Web of data mainstream.

1 Introduction

The meteoric rise of interoperable websites backed by massive amounts of user-generated data over the last decade clearly demonstrates people's interest in using integrated services over the World Wide Web. With the advent of user-generated content and so-called Social Web application programming interfaces it became also clear that some degree of decoupling between data on the Web and the services processing these data is desirable for a variety of reasons. For instance, users should not be forced to publish their data to some kind of "black box" Web services. Also, a shift from service control to user control on how user-generated data can be reused is desirable. The Linking Open Data initiative has shown that Semantic Web technologies such as the Resource Description Framework (RDF) and the SPARQL query language are the tools of choice for designing such interoperable Web applications. However, it is widely known that developing RDF-enabled applications usually requires some non-standard software engineering knowledge. This is still regarded one big hindering factor for getting Semantic Web technologies picked up by non-academic software engineers and a wider range of commercial products.

Besides the complexity introduced by using RDF technologies alone, using so-called linked data [1] from various data sources on the Web requires the consuming software to handle and process additional metadata about this data, mainly

- check data licenses and privacy policies attached to the data,
- assess provenance information about the data consumed and publish provenance information along with the data created,

– assess trust and quality of the data consumed, e. g., based on provenance information.

Such metadata processing is not necessary in traditional software projects building upon rather centralized data storage technologies. Hence it is considered something new and an extra-effort by software developers. We argue that besides making the development of RDF-enabled applications simpler, also higher-level operations like such metadata handling must be simplified for software developers. In this paper, we use Object Triple Mapping (OTM) [2–4], which has been proven to simplify the development of Semantic Web applications [5], and show how this approach does as well contribute to handling metadata in Semantic Web applications.

We show that Object Triple Mapping (OTM) is suitable to process and generate metadata required for distributed Web applications in object-oriented programs. OTM [4] lets developers focus on object-oriented concepts representing actual business logic and hides the most frequent RDF operations from the source code. In Section 3 we extend existing basic formalizations of OTM to be able to attach metadata to object-oriented data items. Afterwards in Section 4, we show how to pass such metadata through the object-oriented program's control flow. Section 4 also contains a detailed example of how provenance information will be collected during a specific object-oriented computation. The paper is concluded in Section 5.

2 Related work

This paper builds upon work from different fields of research on the Semantic Web. The Linking Open Data movement is driven by the observation that Semantic Web standards and technologies have matured to allow building real-world applications upon them, and that the publication of openly usable, real-world data on the Web might foster the development of such applications [6]. Inspite of the success of the initiative resulting in myriad widely usable datasets on the Web, we argue that the development of Semantic Web applications itself still needs to get simpler.

Lassila investigated the design of Semantic Web software, especially with regards to how to organize rather generic operations on RDF data such as reasoning and other operations more related to specific business logic [7]. Oren et al. address on a related problem: in their work on ActiveRDF they showed how to access RDF data in an object-oriented manner, hiding most basic access patterns to RDF from software developers [3]. In their work, they use the PathLog language [8] to explain the handling of object-oriented concepts. The idea of Object Triple Mapping (OTM) is partly inspired from similar work on accessing relational databases [9]. Other, more implementation-centric approaches to OTM include So(m)mer [2] and Elmo for the Java programming language, and Surf RDF for Python. An experimental evaluation showed that developing simple Semantic Web applications using OTM is much easier for the Semantic Web beginner and leads to better source code [5]. The actual semantics of

OTM and possible desirable extensions of the mapping are however not yet fully understood. This is still an open issue, especially to identify the most common patterns implemented in different types of Semantic Web software and separating the core concepts of OTM from optional, use-case-dependent extensions.

In this paper, we investigate how to deal with provenance information of RDF data in object-oriented programs. Provenance information helps consumers to assess data with regard to various issues: Is the data source trustworthy? Has the data been generated using high-quality algorithms? But also, are we allowed to use the data for our purposes? The latter question can, e. g., refer to data licenses or privacy policies. In this paper, we show how Hartig and Zhao's work on Web data provenance [10] can be integrated into OTM. We will mainly focus on the question what data items have been used to construct other data items, i. .e., tracking the input RDF data of an object-oriented computation and see the relation of this input to the output RDF data of the computation.

3 Object-oriented access to Web data formalized

To formally describe how to attach provenance information to the results of object-oriented computations on RDF data, a formal representation of the RDF and OO data models is required as well as a formal mapping between the two data models. In this section, the required formal concepts are presented. First, the RDF data model has an established formal notation building upon the following concepts [11].

Definition 1 (RDF data model) *Let U be the set of* URI references*, B an infinite set of* blank nodes*, and L the set of* literals*.*

- *$V := U \cup B \cup L$ is the set of RDF* nodes*,*
- *$R := (U \cup B) \times U \times V$ is the set of all* triples *or* statements*, that is, arcs connecting two nodes being labelled with a URI,*
- *any $G \subseteq R$ is an* RDF graph*.*

For their work on OTM, Oren et al. use PathLog [8] to describe object-oriented access to data [3]. PathLog itself differentiates between scalar and set-valued class members. For the scope of this paper however, dealing with set-valued class members will be sufficient. Also, we will use a simplified version of the *semantic structure* explained in [8].

Definition 2 (OO data model) *Let $\mathcal{N}$ be a set of names. The set of PathLog references $\mathcal{R}_{\mathcal{N}}$ is defined inductively as follows.*

- *$n \in \mathcal{N}$ is a reference, also called a* simple reference*.*
- *for references t_0, t_1 and a simple reference s*
  - *$t_0.s$ is a reference, called a* path
  - *$t_0 : s$ and $t_0[s \rightarrow t_1]$ are references, called* molecules*.*

Furthermore, a semantic structure is a triple $(\mathcal{N}, O, I)$ such that

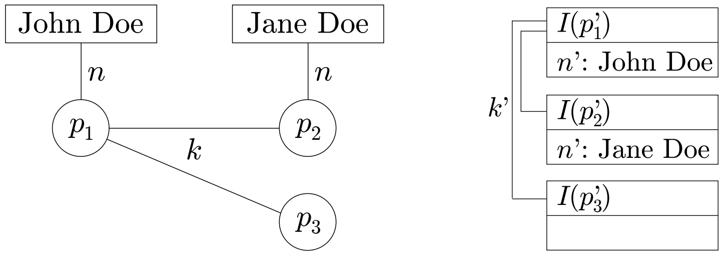


Fig. 1. Representing social information in RDF (left) and OOP (right).

- *O is a set of objects and*
- *The* interpretation $I : \mathcal{R}_\mathcal{N} \to 2^O$ *relates references to objects.*

In the following, we will show how RDF and PathLog can be used to express information in the respective data model.

Example 1 (comparison of RDF and OO data model) *Let p_1, p_2, $p_3 \in U$ be URI denoting three people, $n \in U$ be the URI* `foaf:name` *and $k \in U$ be the URI* `foaf:knows`[1]. *An RDF graph describing p_1 and p_2 might look the following (Fig. 1, left).*

$$G := \Big\{ \langle p_1, n, \text{“John Doe”} \rangle,\ \langle p_1, k, p_2 \rangle,\ \langle p_1, k, p_3 \rangle,\ \langle p_2, n, \text{“Jane Doe”} \rangle \Big\}$$

Let furthermore p_1', p_2', $p_3' \in \mathcal{N}$ be object names denoting three people and n', $k' \in \mathcal{N}$ fields labelled `name` *and* `knownPeople`*. The OO representation of G (Fig. 1, right) requires a semantic structure $(\mathcal{N}, O, I)$ such that*

$$I(p_1'.n') = \{\text{“John Doe”}\}$$
$$I(p_2'.n') = \{\text{“Jane Doe”}\}$$
$$I(p_1'.k') = I(p_2') \cup I(p_3')$$

The mapping of RDF data to OOP concepts as shown in Example 1 and vice versa has been formalized [5]. This formalization can be adopted to the concepts of PathLog as follows.

Definition 3 (Object triple mapping, OTM) *An* object triple mapping *for an RDF graph $G \subseteq (U \cup B) \times U \times V$ is a tuple $(\mathcal{N}, O, I, m_t, m_a)$, such that*

- *$(\mathcal{N}, O, I)$ is a semantic structure,*
- *the* vocabulary map $m_t : F \to U$ *maps a field names $F \subseteq \mathcal{N}$ to properties,*
- *the* instance map $m_a : O \to U$ *maps objects to resources,*
- *and the following holds for all $o \in O$, $n \in I^{-1}(\{s\})$, $f \in F$:*
  - *the mapping is complete:*
    $\forall u \in U : \langle m_a(o), m_t(f), u \rangle \in G \to \exists o' \in I(n.f) : m_a(o') = u$
    $\forall o' \in O : o' \in I(n.f) \to \langle m_a(o), m_t(f), m_a(o') \rangle \in G$

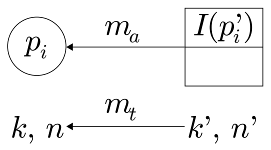


Fig. 2. Mapping OO concepts (right) on RDF (left).

- *the mapping is injective, i. e.* $m_{a\,|I(n.f)}$ *is injective.*

The idea of OTM is illustrated in Fig. 2 using the names from Example 1 and Definition 3. Besides such purely formal description of Object Triple Mapping, concrete implementations need to consider additional aspects such as object equivalence. Depending on the context, deciding on semantic equivalence can be hard. For the scope of this paper on metadata processing, considering syntactical equivalence based solely on the URI returned by m_a is sufficient. Also, concrete implementations need to map the semantics of RDF, such as lists or reification, of RDF Schema, such as class hierarchies, and OWL, such as constraints on classes and properties, to the semantics of their supported programming language. The general feasibility of such features is discussed in [12].

4 Tracing the provenance of object-oriented computations

4.1 Attaching metadata to RDF and OO data

RDF metadata will be attached to *data items* [10] like RDF graphs or triples. This applies for different kinds of metadata: Data licenses and policies specify whether a specific data item can be used in specific contexts or for specific purposes. Similarly, provenance metadata describe how a data item has been derived, and what other data items have been employed for such derivation. It is important to see that such kind of information must be attached to data items like RDF graphs or triples, not only to RDF resources. A similar concept is required to express metadata information about object-oriented concepts. In this section, we introduce the concept of *object-oriented data items.* The OO data items corresponding to RDF triples will be values together with the variables they are assigned to, or the values returned by a method together with the method called. Both cases can be represented by the PathLog reference used to obtain a certain object.

Definition 4 (OO data item) *Let* $(\mathcal{N}, O, I, m_t, m_a)$ *be an object triple mapping for some RDF graph* $G \subseteq (U \cup B) \times U \times V$, *and* $r \in \mathcal{R}_{\mathcal{N}}$ *a reference. Whenever we access an* $o \in I(r)$, *we say* o *has been obtained using the* OO data item $(r, I) \in \mathcal{R}_{\mathcal{N}} \times O^{\mathcal{R}_{\mathcal{N}}}$.

[1] The `foaf` prefix denotes the URI namespace http://xmlns.com/foaf/0.1/

References $r = n.f$ for names $n, f \in \mathcal{N}$ with n referring to a single object $I(n) = \{o\}$ are essential in object-oriented source code. Due to the nature of OTM, an object o' obtained using the data item $(n.f, I)$ has actually been obtained using the RDF triple

$$\langle m_a(o), m_t(f), m_a(o') \rangle$$

More complex references might be translated to other types of RDF data items, e. g., SPARQL query results. OO data items can now be used to process or update metadata attached to them. Such metadata can be obligations or restrictions of use due to privacy or license regulations, but also trust and quality assessments. Due to the mapping defined between the object-oriented and RDF data model, not only objects can be mapped to resources, but also OO data items to their corresponding RDF data items.

4.2 Aggregation of object-oriented provenance information

A significant part of tracing and generating provenance information is to relate the data items serving as input for a computation to its output data items. Such computation, or *data creation* [10], could, e. g., be performed by a method call in an OO program. However, from the object-oriented program flow within and around such methods further valuable information on data provenance can be aggregated.

Aggregation on data flow. If an object o has been obtained from a data item (r, I), a number of names appearing in r potentially identify objects $o'_1, \ldots, o'_\ell$ (a simple case is $r = n.f$, such that $I(n) = \{o'_1\}$). Naturally, all data items used to obtain $o'_1, \ldots, o'_\ell$ have also been used to eventually get hold of o.

Aggregation on object equality. A very common and important operation on objects o_1, o_2 is checking for equality. This can explicitly be triggered in source code by the software developer, but is also performed by higher-level operations provided by programming libraries, such as constructing the intersection of two sets of objects. If some action is taken because o_1 and o_2 have been detected being equal, all data items used to obtain o_1 and o_2, and the ones used to detect equality have actually been used to assign the value. Hence, for subsequent uses of o_1 and o_2, this aggregation of source data items should be considered. In implementations, this can be realized by adding side effects to equality checking.

4.3 Example: tracing provenance information of social network data

In this section, we present a comprehensive example of how to benefit from object-oriented provenance information on RDF data. The example illustrates that OO provenance information can be gathered during program execution without placing provenance-related artifacts in the OO source code. Given aquaintance information, we want to suggest new contacts for a person if the new contact is a mutual friend of at least to of the person's friends. The pseudo code for this computation is shown in Fig. 3. The relationships are described using the `foaf:knows` property k, and we introduce an additional RDF property s for suggested contacts. Consider the following RDF graph $G \subseteq (U \cup B) \times U \times V$ and person resources $p_1, \ldots, p_4 \in U$. If G contains

$$\langle p_1, k, p_2\rangle, \quad \langle p_1, k, p_3\rangle, \quad \langle p_2, k, p_4\rangle, \quad \langle p_3, k, p_4\rangle$$

then we want the suggestion $\langle p_1, s, p_4\rangle$ to be created. Additionally, we want to be able to see the facts have been used to derive this statement, e. g., for later trust or quality assessments based on these provenance information.

To see what data items are used for the computation, let $(\mathcal{N}, O, I, m_t, m_a)$ be an object triple mapping and $o_1, \ldots, o_4, o'_4 \in O$ be objects representing $p_1, \ldots, p_4$. Going through the computation of `find_friends_for(`o_1`)` as described in Figure 3, the following OO provenance information is aggregated.

– In line 1, o_1 passed via `person`, i. e. $I(\texttt{person}) = o_1$.
– In line 2, the OTM interpretation returns $I(\texttt{person.friends}) = \{o_2, o_3\}$. The objects o_2, o_3 assigned to `friend1` will have the provenance information $(\texttt{person.friends}, I)$.
– In line 3, the loop-local interpretations return $I_1(\texttt{friend1.friends}) = \{o_4\}$, $I_2(\texttt{friend1.friends}) = \{o'_4\}$. The provenance information of o_4 will be $(\texttt{friend1.friends}, I_1)$ and for o'_4 it will be $(\texttt{friend1.friends}, I_2)$.
– In line 4, o_4 is not found equal to a friend of o_1.
– In line 6 in the first iteration, o_4 is not a friendship candidate, but is added to the candidates in line 9.
– In line 6 in the second iteration, o'_4 is found being equal to the friendship candidate o_4, hence o_4's and o'_4's provenance are both aggregated to $(\texttt{friend1.friends}, I_1)$, $(\texttt{friend1.friends}, I_2)$.
– In line 7 in the second iteration, o'_4 is suggested as a friend including this combined provenance information from line 6.

When the friend suggestion o'_4 to o_1 is mapped to the RDF statement $\langle p_1, s, p_4\rangle$, this statement will eventually have been created using all OO data items and the corresponding RDF statements listed in Table 1. With the help of trust-

```
1   find_friends_for(person):
2       foreach friend1 in person.friends
3           foreach friend2 in friend1.friends
4               if person.friends contains friend2
5                   nothing to do, continue
6               else if candidates contains friend2
7                   add friend2 to suggestions
8               else
9                   add friend2 to candidates
```

Fig. 3. Object-oriented search for new contacts in a social network

OO data item	RDF statement	Mapping from OO to RDF data
($\texttt{person.friends}$, I)	$\langle p_1, k, p_2\rangle$ $\langle p_1, k, p_3\rangle$	$m_a(I(\texttt{person})) = \{p_1\}$ $m_t(\texttt{friends}) = k$ $m_a(I(\texttt{person.friends})) = \{p_2, p_3\}$
($\texttt{friend1.friends}$, I_1)	$\langle p_2, k, p_4\rangle$	$m_a(I_1(\texttt{friend1})) = \{p_2\}$ $m_t(\texttt{friends}) = k$ $m_a(I_1(\texttt{friend1.friends})) = \{p_4\}$
($\texttt{friend1.friends}$, I_2)	$\langle p_3, k, p_4\rangle$	$m_a(I_2(\texttt{friend1})) = \{p_3\}$ $m_t(\texttt{friends}) = k$ $m_a(I_2(\texttt{friend1.friends})) = \{p_4\}$

Table 1. Data items leading to the suggestion $\langle p_1, s, p_4\rangle$ in Fig. 3

related metadata potentially attached to these source triples, the result statement $\langle p_1, s, p_4\rangle$ can now be dealt with appropriately in further steps of the computation.

4.4 Implementation

The proposed approach for tracing the provenance of object-oriented computations has been integrated in our Java object triple mapping implementation OTMj [5]. In OTMj, an RDF resource u is mapped to a dynamic proxy object o implementing the interfaces corresponding to the known RDF types of u. This allowed for a straight-forward integration of the OO provenance model proposed: If o is accessed using a data item (r, I), OTMj returns an object o' acting as a proxy for o. In addition to o, the proxy o' has the data item (r, I) attached. OTMj can be obtained as open source software.[2]

Our implementation uses reification to link source and result triples using the concept of *data creations* as defined by Hartig and Zhao [10]. Given a triple $\langle s_1, p_1, o_1\rangle$ having been created using another triple $\langle s_2, p_2, o_2\rangle$, our implementation creates metadata as illustrated in Fig. 4[3].

5 Conclusion

In this paper we showed how metadata about RDF data can be processed and generated within object-oriented computations on these data using Object Triple Mapping (OTM). We extended existing formalisms to OTM and introduced the notion of object-oriented data items. Using these concepts we showed how to trace the provenance of RDF data items on their way through object-oriented computations and presented a detailed example and a ready-to-use implementation of the approach. Our next steps in research include a tighter integration with potential sources of provenance information such as SQUIN [13] or context-based

[2] http://projects.quasthoffs.de/otm-j

[3] The `prv` prefix denotes the URI namespace http://trdf.sourceforge.net/provenance/ns#

```
_:t1  a                 rdf:Statement ;
      rdf:subject       s1 ;
      rdf:predicate     p1 ;
      rdf:object        o1 ;
      prv:createdBy     _:c1 .
_:t2  a                 rdf:Statement ;
      rdf:subject       s2 ;
      rdf:predicate     p2 ;
      rdf:object        o2 .
_:c1  a                 prv:DataCreation;
      prv:usedData      _:t2 ;
      prv:performedBy   ... ;
      prv:performedAt   ... .
```

Fig. 4. N3 representation of metadata generated by OTMj during data creation.

reasoners [14], and with programming frameworks oriented towards generating complete applications instead of focusing on the data backend only, such as the Grails RDFa plugin[4].

This paper shows that the development of Semantic Web applications cannot only be simplified by focusing on established and widely understood abstractions like object-oriented programming, but also, functionality essential for a working Web of data can be integrated in these abstraction layers without adding further burden to software developers. Without these abstractions, metadata handling would have to be implemented per use-case, a task potentially being skipped due to time constraints or deferred for unlimited time by software engineers. We're looking forward to learning what will eventually make non-expert software developers use Semantic Web technologies and how our work will contribute to making this happen.

References

1. Berners-Lee, T.: Linked data. http://www.w3.org/DesignIssues/LinkedData.html (2006)
2. Story, H.: Java annotations and the semantic web. http://blogs.sun.com/bblfish/entry/java_annotations_the_semantic_web (2005)
3. Oren, E., Heitmann, B., Decker, S.: Activerdf: Embedding semantic web data into object-oriented languages. J. Web Sem. **6**(3) (2008) 191–202
4. Quasthoff, M., Meinel, C.: Design patterns for object triple mapping. In: Proc. of IEEE SCC 2009. (2009)
5. Quasthoff, M., Sack, H., Meinel, C.: How to simplify building semantic web applications. In: Proc. of the 5th International Workshop on Semantic Web Enabled Software Engineering, CEUR-WS.org (2009)
6. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: Proceedings of 6th International Semantic Web

[4] http://grails.org/plugin/rdfa

Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC 2007). Springer (2008) 722–735
7. Lassila, O.: Programming semantic web applications: A synthesis of knowledge representation and semi-structured data. ISBN 978-951-22-8984-4 (2007)
8. Frohn, J., Lausen, G., Uphoff, H.: Access to objects by path expressions and rules. In Bocca, J.B., Jarke, M., Zaniolo, C., eds.: VLDB, Morgan Kaufmann (1994) 273–284
9. Fowler, M., Rice, D.: Patterns of Enterprise Application Architecture. Addison-Wesley (2003)
10. Hartig, O., Zhao, J.: Publishing and Consuming Provenance Metadata on the Web of Linked Data. In: Proceedings of the 3rd International Provenance and Annotation Workshop (IPAW). (2010)
11. Manola, F., Miller, E.: Rdf primer. w3c recommendation 10 february 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/ (2004)
12. Quasthoff, M., Sack, H., Meinel, C.: Can software developers use linked data vocabulary? In: Proc. of I-Semantics '09. (2009)
13. Hartig, O., Bizer, C., Freytag, J.C.: Executing sparql queries over the web of linked data. In: Proc. of the 8th International Semantic Web Conference, Springer (2009)
14. Delbru, R., Polleres, A., Tummarello, G., Decker, S.: Context dependent reasoning for semantic documents in sindice. In: Proceedings of the 4th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2008), 7th International Semantic Web Conference, Kalrsruhe, Germany (10 2008)

Towards a Data-Centric Notion of Trust in the Semantic Web
(A Position Statement)

Olaf Hartig

Department of Computer Science
Humboldt-Universität zu Berlin
hartig@informatik.hu-berlin.de

Abstract. Existing research on trust in the Semantic Web extensively studies trustworthiness and trust in the context of active entities such as persons and agents. However, few work exist that focus on the content in the Semantic Web and that study trustworthiness as an information quality criterion. Hence, computer systems that use the trustworthiness of Semantic Web data for filtering or decision making usually apply a very simple assessment approach: each data object is related to some kind of a source for which a trust score can be determined using one of the methods that exist for active entities; this score is then adopted for the trustworthiness of the data object. In this position paper we argue that such a simple notion of trustworthiness for data is insufficient and we propose to adjust the focus of trust research for the Semantic Web from an actor-centric view to a data-centric perspective.

1 Introduction

Today, a large amount of RDF data is published on the Web; large datasets are interlinked; new applications emerge that utilize this data in novel and innovative ways. However, the openness of the Web and the ease to combine Linked Data from different sources creates new challenges. Unreliable data could dominate the result of queries, taint inferred data, affect local knowledge bases, or may have negative or misleading impact on software agents. Hence, questions of reliability and trustworthiness must be addressed.

A great many approaches exist that allow for a calculation of trust values for active entities such as persons, software agents, or peers in a P2P scenario [1]. While several of these approaches can be applied to consider trustworthiness of data providers in the Semantic Web (e.g. [2,3,4]), little has been done considering the data itself. Existing work applies a very simple assessment approach: each data object is related to some kind of a source for which a trust score can be determined using one of the methods that exist for active entities; this score is then adopted as the trustworthiness of the data object. However, simply adopting the trustworthiness of a source for its data does not consider cases where statements have multiple sources, where providers (re)publish data aggregated from the original sources, or where inference engines discover implicit facts from statements of other sources. Hence, source-level approaches are too coarse-grained and, thus, insufficient for the Web of data. Furthermore, our knowledge of the provenance of a data object is not the only criterion that can be applied to assess the trustworthiness of the object. Other factors such as the correctness of the data or the opinion of another data consumer may affect our decision.

In this paper we argue for making data the central subject of research on trust in the Semantic Web. Therefore, we propose to reconsider the actor-centric trust research for the Semantic Web and conceive trust in the Semantic Web more as an effort that fits in the wider area of information quality (IQ) research. IQ reflects the fitness for use of information [5]. Since this fitness for use may depend on various factors, IQ is a multi-dimensional concept which includes different IQ criteria such as accuracy, completeness, and timeliness [6]. Consequently, we consider the trustworthiness of Linked Data as another such IQ criterion.

Our fundamental understanding of the trustworthiness of data is the subjective belief or disbelief in the truth of the information represented by this data [7]. The decision to believe or to disbelieve is affected by a broad variety of influences. Notice, this complexity renders the actor-centric idea of simply representing the trustworthiness of data by adopting the trust value of an actor as insufficient. We propose to classify the influences in three categories: i) information quality, ii) provenance, and iii) others' opinions. In the remainder of this paper we discuss these categories in more detail (cf. Sections 3 to 5). As a basis for this discussion we review existing approaches that focus on the trustworthiness of data or on content in general (cf. Section 2).

2 Existing Research

In this paper we propose to focus trust research in the Semantic Web on the trustworthiness of data. Similarily, Gil and Artz [8] identify that the majority of existing work on trust "focuses on entity-centered issues such as authentication and reputation and does not take into account the content." Therefore, the authors propose to study *content trust* which "is a trust judgment on a particular piece of information in a given context." As the units of content that are being judged Gil and Artz identify Web resources in general.

To the best of our knowledge, there is still only very few work on content trust in the Semantic Web community. With IWTrust and FilmTrust, two systems have been proposed that consider the trustworthiness of statements during processing tasks and for decisions. IWTrust [9], the trust component of the Inference Web answering engine, understands trust in answers as the trust in sources and in users. Similarily, FilmTrust [10] represents the trustworthiness of movie reviews by a user's trust in the reviewer and in other users' competence to recommend movies. A similar understanding of the trustworthiness of statements published on the Semantic Web has been presented by Rowe and Butters [11]. Their approach adopts a contextual trust value determined for the person who asserted a statement as the trustworthiness of the statement itself. Hence, even if these approaches take the trustworthiness of statements into account they still apply an actor-centric view.

Systems that explicitly focus on trust assessments for statements are TRELLIS and QUATRO Plus. The TRELLIS [12] system assesses the truth of statements by considering their provenance and related statements. Users can rate information sources and follow the assessments that are presented with the corresponding analysis and the influencing facts. The QUATRO Plus [13] system enables trust assessments for descriptions of Web resources; trust assessment is based on user ratings of these descriptions. Both approaches, however, do not provide a trust model that explicity represents the trustworthiness of content.

Mazzieri [14] and Richardson et al. [15] propose such trust models; they represent content trust for RDF data on the level of RDF statements. Mazzieri introduces fuzzy RDF; a *membership value* associated with each statement represents the likelyhood that the statement belongs to the RDF graph. By equating those membership values with trustworthiness of statements Mazzieri inappropriately mixes two different concepts; trustworthiness is not the same as a fuzzy notion of truth nor is trustworthiness of RDF statements tied to a specific RDF graph. Richardson et al. [15] represent a user's personal belief in a statement by a value in the interval [0,1]. Besides the vague explanation that a "high value means [...] the statement is accurate, credible, and/or relevant" the approach lacks a more formal definition of those values. Thus, what is missing in all cases is a well-founded definition of the meaning of trustworthiness of RDF data.

Another related system is the WIQA framework [5] that permits quality based filtering of data aggregated from the Web. Filtering is based on policies; these policies are constraints that are enforced during query evaluation and that restrict the resultset of queries. Furthermore, the system explaines why data should be trusted, more precisely, why results passed the filters. The WIQA approach does not use explicit scores for IQ criteria. However, missing scores prevent comparisons of the trustworthiness of different pieces of data; moreover, without explicit ratings it is impossible to compare the opinions of multiple data consumers regarding the trustworthiness of the same data. Instead of a filtering approach other work focuses on the ranking of Linked Data [16,17].

Other relevant research is provided by the IQ community where trustworthiness of data is often considered synonymous to believability [6]. Lee et al. [18] decompose believability into three sub-dimensions: trustworthiness of source, reasonableness of data, and temporality of data. Following this differentiation, Prat and Madnick [19] propose a provenance based approach to measure believability by aggregating quality scores for the sub-dimensions. Another provenance based approach has been proposed by Dai et al. [20]. Their main idea is to determine the trustworthiness of a data item by considering source data from which the item has been derived. Furthermore, the approach compares data items to other, similar, but also to conflicting data items. In [21] we present a generic approach for methods that assess IQ of Web data and we apply this approach for the IQ criterion timeliness. Similarily, this generic approach can also be used as the base for a method to assess the trustworthiness of Web data.

3 Influence Category: Provenance

The decision to believe that a data object represents the truth includes considering questions such as the following:

– How was the creation of the data conducted?
– Who or what participated in the creation of the data and how much do I trust this participant?
– To what extend does the input from which the data was produced represents the truth?
– What happened to the data since its creation; how likely is a manipulation?

These questions refer to the provenance of the data object. We understand the provenance of a data object as everything that is related to how the object in its current state came to be. Hence, provenance information about a data object is information about the whole history of this object. This history may start long before the object has been created itself because the provenance of source artifacts used for the creation is also a relevant part of this history. Hence, this history includes multiple actors that participated in various, different roles. All of these actors had a certain influence on the data object and the current state of the object in which it is available to us.

Traditionally, there are two main areas in which researchers study provenance of data: workflows and databases [22]. Research in these areas usually focuses on the creation of data, be it a data product generated by a workflow [23] or the query results created by a database query engine [24]. This focus is reasonable given that workflows and databases are self-contained systems. The Web, in contrast, is a much more open environment. A data object on the Web may have passed through many (virtual) hands before it is finally available in the current application. Hence, the history of a data object includes more aspects than the creation. This additional information is of interest when it comes to assessing the trustworthiness of data objects from the Web as the last of the aforementioned, provenance-related questions illustrates. For this reason we propose a new model for Web data provenance in [25]; this model considers the Web based access to data and the creation of this data equally important. Based on this model we present concepts and tools to integrate provenance information into the Web of data in [26]. This information can then be used to apply provenance based assessment approaches as we introduce in [21].

4 Influence Category: Information Quality

Even if we consider trustworthiness as a criterion of information quality other IQ criteria are likely to affect our trustworthiness assessment. Knowing about a lack of correctness, accuracy, or consistency in the data reduces our belief in the truth of the information represented by that data. Apart from these obvious influences it often depends on the context if a specific criterion is relevant for the trustworthiness assessment. An example is completeness: knowing that a dataset is incomplete may give rise to doubt because missing data may change the information in the dataset. On the other hand, the part of the data that is available might still be trustworthy and, thus, be usable in some application. Similarily, the relevancy of time-related criteria depends on context and application: old data might not be believed to be true anymore in some context; in another application the low currency might not cause a reduced trustworthiness score for the same data, for instance, when the development of certain data values over time should be analyzed.

As can be seen from these examples, trustworthiness can be understood as a more abstract kind of IQ criteria. Compared to other, independent criteria, trustworthiness comprises multiple other criteria. This characteristic means that scores for other, relevant IQ criteria have to be determined as a prerequisite to assess trustworthiness. These scores, then, have to be weighted during the actual trustworthiness assessment so that the context-dependent relevancy of the corresponding IQ criteria is reflected.

5 Influence Category: Other Opinions

An additional factor that can be used to assess a user's belief in the truth of a data object is the opinion of other consumers of this data. This approach is similar to the idea of determining the trustworthiness of actors using trust assertions in a Web of trust. In addition to the trust assertions about the actor in question, these Web of trust approaches also take the trustworthiness of the actors into account that provided the assertions. This principle must be adopted for opinion based assessment of the trustworthiness of data: the assessment is either based on the opinion of trusted consumers only or opinions must be weigthed by the trust in the corresponding consumer to provide reliable trustworthiness scores for data. Furthermore, we note that the development of trustworthiness assessment approaches that take other consumers' opinion into account can benefit from existing work on recommendation systems.

6 Conclusions

In this paper we argue to apply a data-centric view for further research on trust for the Semantic Web. We understand trustworthiness of Semantic Web data as a criterion of information quality and identify main categories of factors that affect the assessment of this criterion. Even if we discuss these categories separately we suggest that an actual assessment approach should take factors from all categories into account. As an additional requirement for such trustworthiness assessment approaches we note that the assessment system should be able to explain a determined trustworthiness score to end users.

References

1. Artz, D., Gil, Y.: A Survey of Trust in Computer Science and the Semantic Web. Journal of Web Semantics **5**(2) (2007)
2. Golbeck, J., Parsia, B., Hendler, J.A.: Trust Networks on the Semantic Web. In: Proc. of the 7th Int. Workshop on Cooperative Information Agents. (2003)
3. Ziegler, C.N., Lausen, G.: Spreading Activation Models for Trust Propagation. In: Proc. of the Int. Conference on e-Technology, e-Commerce, and e-Service (EEE)

4. Brondsema, D., Schamp, A.: Konfidi: Trust Networks Using PGP and RDF. In: Proc. of the Workshop on Models of Trust for the Web at WWW. (2006)
5. Bizer, C., Cyganiak, R.: Quality-Driven Information Filtering using the WIQA Policy Framework. Journal of Web Semantics **7**(1) (2009)
6. Naumann, F.: Quality-Driven Query Answering for Integrated Information Systems. Springer Verlag (2002)
7. Hartig, O.: Querying Trust in RDF Data with tSPARQL. In: Proc. of the 6th European Semantic Web Conference (ESWC). (2009)
8. Gil, Y., Artz, D.: Towards Content Trust of Web Resources. Journal of Web Semantics **5**(4) (2007)
9. Zaihrayeu, I., da Silva, P.P., McGuinness, D.L.: IWTrust: Improving User Trust in Answers from the Web. In: Proc. of the 3rd International Conference on Trust Management (iTrust). (2005)
10. Golbeck, J., Hendler, J.: FilmTrust: Movie Recommendations using Trust in Web-based Social Networks. In: Proc. of CCNC. (2006)
11. Rowe, M., Butters, J.: Assessing Trust: Contextual Accountability. In: Proc. of the 1st Workshop Trust and Privacy on the Social and Sem. Web at ESWC. (2009)
12. Gil, Y., Ratnakar, V.: Trusting Information Sources One Citizen at a Time. In: Proc. of the 1st International Semantic Web Conference (ISWC). (2002)
13. Archer, P., Ferrari, E., Karkaletsis, V., Konstantopoulos, S., Koukourikos, A., Perego, A.: QUATRO Plus: Quality You Can Trust? In: Proc. of the 1st Workshop on Trust and Privacy on the Social and Semantic Web at ESWC. (2009)
14. Mazzieri, M.: A Fuzzy RDF Semantics to Represent Trust Metadata. In: Proc. of the Italian Workshop on Semantic Web Applications and Perspectives. (2004)
15. Richardson, M., Agrawal, R., Domingos, P.: Trust Management for the Semantic Web. In: Proc. of the 2nd International Semantic Web Conference (ISWC). (2003)
16. Harth, A., Kinsella, S., Decker, S.: Using Naming Authority to Rank Data and Ontologies for Web Search. In: Proc. of the Int. Semantic Web Conference. (2009)
17. Toupikov, N., Umbrich, J., Delbru, R., Hausenblas, M., Tummarello, G.: DING! Dataset Ranking using Formal Descriptions. In: Proc. of the Linked Data on the Web Workshop at WWW. (2009)
18. Lee, Y., Pipino, L., Funk, J., Wang, R.: Journey to Data Quality. MIT Press, Cambridge, MA, USA (2006)
19. Prat, N., Madnick, S.: Measuring Data Believability: A Provenance Approach. In: Proc. of the 41st Hawaii Int. Conference on System Sciences (HICSS). (2008)
20. Dai, C., Lin, D., Bertino, E., Kantarcioglu, M.: An Approach to Evaluate Data Trustworthiness Based on Data Provenance. In: Proc. of the 5th VLDB Workshop on Secure Data Management. (2008)
21. Hartig, O., Zhao, J.: Using Web Data Provenance for Quality Assessment. In: Proc. of the Role of Sem. Web in Provenance Management at ISWC. (2009)
22. Tan, W.C.: Provenance in Databases: Past, Current, and Future. IEEE Data Engineering Bulletin **30**(4) (2007)
23. Davidson, S.B., Boulakia, S.C., Eyal, A., Ludäscher, B., McPhillips, T.M., Bowers, S., Anand, M.K., Freire, J.: Provenance in Scientific Workflow Systems. IEEE Data Engineering Bulletin **30**(4) (2007)
24. Cheney, J., Chiticariu, L., Tan, W.C.: Provenance in Databases: Why, How, and Where. Foundations and Trends in Databases **1**(4) (2009)
25. Hartig, O.: Provenance Information in the Web of Data. In: Proc. of the Linked Data on the Web Workshop at WWW. (2009)
26. Hartig, O., Zhao, J.: Publishing and Consuming Provenance Metadata on the Web of Linked Data. In: Proc. of 3rd Int. Provenance and Annotation Workshop. (2010)