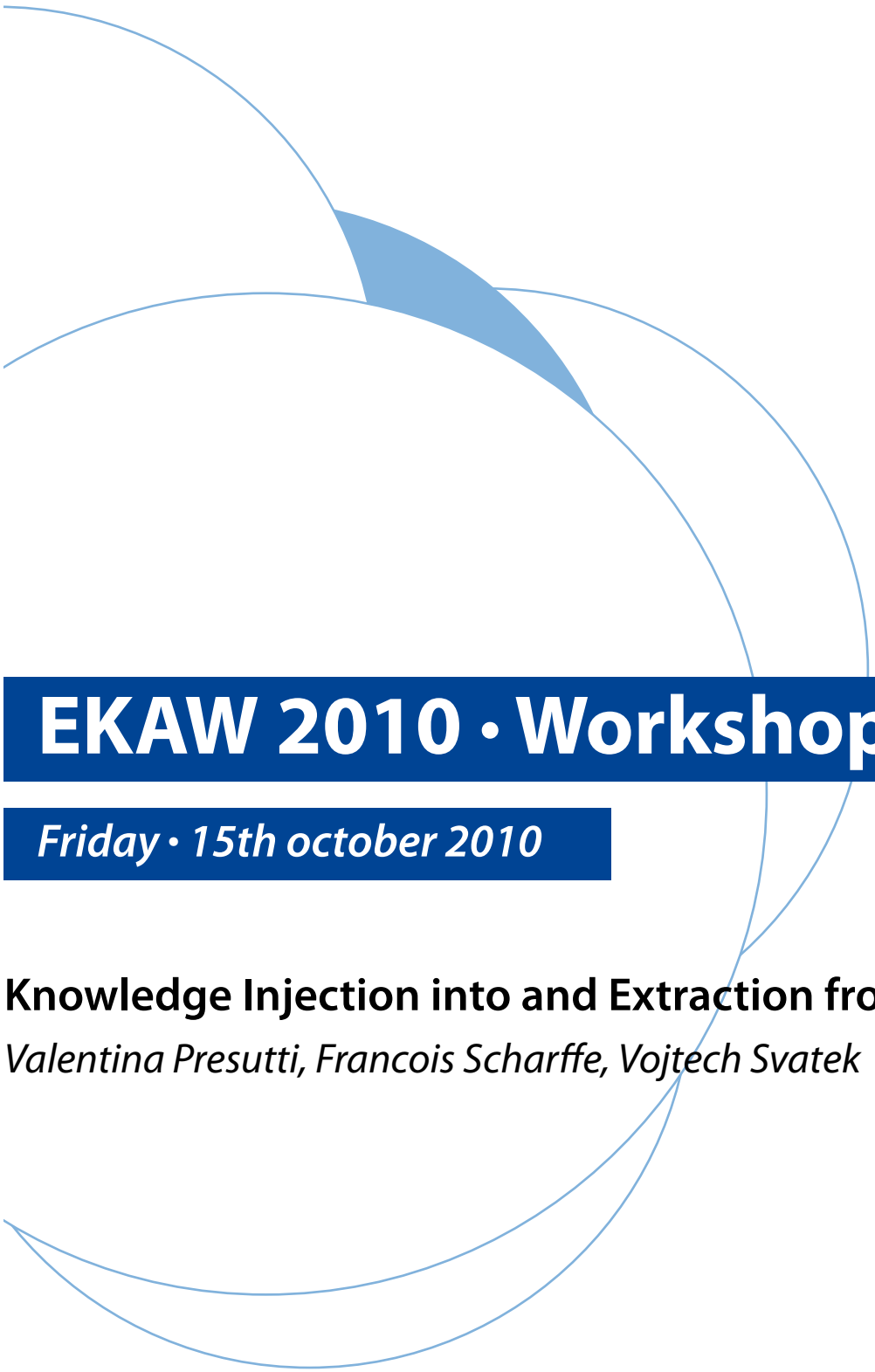

EKAW 2010 · Workshop W5

Friday · 15th october 2010

Knowledge Injection into and Extraction from Linked Data

Valentina Presutti, Francois Scharffe, Vojtech Svatek

KIELD 2010

1^{st} EKAW Workshop on Knowledge Injection into and Extraction from Linked Data

Foreword

The rapid growth of the Linked Data (LD) cloud, in parallel with on-the-fly design of relevant vocabularies, presents new opportunities for traditional research disciplines such as Knowledge Modelling and Knowledge Discovery from Data. Most notably:

- Although the popular vocabularies reflect today needs, they sometimes lack deeper ontological reflections. State-of-the-art knowledge modelling, especially the pattern-based ontology design principles, could help connect Linked Data vocabularies to more sophisticated models, while keeping themselves simple. Furthermore, collaborative ontology design methodologies could find their way to the process of vocabulary design, currently undertaken by VoCamp communities and stand-alone groups.

- The linked data themselves represent a large and growing resource woven from numerous components. Empirical knowledge discovery in linked data, carried out by machine learning and data mining algorithms, could reveal interesting patterns on frequently used structures, which could then be fuelled back to the vocabulary design. Existing techniques for mining network-structured data, such as graph databases or web links, are likely to require adaptation so as to take account of links that are typed according to semantically rich and heterogeneous schemata.

This complex of research challenges was the main incentive for organizing the 1st Workshop on Knowledge Injection into and Extraction from Linked Data (KIELD 2010), collocated with the 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2010), in Lisbon, Portugal.

KIELD 2010 aimed at being an interdisciplinary event, of interest for both researchers and practitioners in all three areas: linked data, knowledge modeling and knowledge discovery. Furthermore, it also assumed space for ongoing and pioneering research activities, which would still be too preliminary as conference publications, although extremely hot as topics in the semantic technology field.

One of the highlights of the half-day event, held in the afternoon of October 15, 2010, following the main EKAW conference, was the keynote talk by Prof. Martin Hepp, "Ontology Engineering for Linked Data: What Makes for a Good Ontology?", which discussed the impact of ontology design choices and ontology quality criteria on the overall impact of the linked data initiatives.

The workshop received six submissions, which were all carefully reviewed by at least 2 (and mostly 3) reviewers. Five of the submissions satisfied the quality standards for being accepted as full papers. There was also an additional call for short late-breaking news, leading to two submissions (which did not undergo a full review but were checked for relevance by the workshop organizers).

Two of the contributed talks focused (following a similar direction as the keynote) on the need to 'inject' more knowledge into the linked data vocabularies. Nuzzolese et al. dealt with expliciting the semantics of (especially, relational) data when putting them to RDF, through meta-modeling such data in OWL; i.e., the linked data are thus 'injected' with knowledge already when the resource is being built, i.e. 'a priori'. In contrast, Vacura&Svátek analyzed some implicit assumptions of vocabularies (specifically for FOAF) and suggested to make them explicit when the given vocabulary is imported as upper-level into a more specific ontology; this corresponds to 'a posteriori injection' of knowledge into linked data (at the reuse time of the vocabulary).

Three of the talks focused on the possibility to 'extract' useful knowledge from linked data, or to intertwine linked data with other data resources in order to increase the quality of these resources (which, presumably, has the potential of 'injecting' the new knowledge back into the original linked data resources). Markotschi&Völker presented a new online game with a purpose, combining the wisdom of crowds with linked data in order to build richer ontological descriptions of concepts. Drăgan et al. showed how semantic desktop data can be published as linked data, via unifying local and web identifiers of entities. Finally, Valle et al. presented a case study in transferring a database of tenders to linked data while exploiting existing LD resources such as DBpedia and Geonames.

The workshop chairs are grateful to all people who contributed to the event, from the PC members, through the presenters (most notably, to the keynote speaker), to all participants. A special thank is due to the local organizers, for their support.

Lisbon, October 15, 2010

Valentina Presutti
François Scharffe
Vojtěch Svátek

Program Committee Members

- Eva Blomqvist, STLab ISTC-CNR, Italy
- Ciro Cattuto, ISI Foundation, Italy
- Claudia d'Amato, University of Bari, Italy
- Mathieu d'Aquin, KMI Open University, UK
- Nicola Fanizzi, University of Bari, Italy
- Aldo Gangemi, STLab ISTC-CNR, Italy
- Alfio Gliozzo, STLab ISTC-CNR, Italy
- Marko Grobelnik, Jozef Stefan Institute, Slovenia
- Tom Heath, Talis, UK
- Luigi Iannone, Manchester University, UK
- David Jensen, University of Massachusetts Amherst, USA
- Agnieszka Lawrynowicz, Poznan University of Technology, Poland
- Pascal Poncelet, LIRMM, Universit Montpellier 2, France
- Marko Rodriguez, AT&T, and Vrije Universiteit Brussels, Belgium
- Steffen Staab, University of Koblenz-Landau, Germany

Additional Reviewers

Alessandro Adamou, STLab ISTC-CNR, Italy
Enrico Daga, STLab ISTC-CNR, Italy
Andrea Giovanni Nuzzolese, STLab ISTC-CNR, Italy

Workshop Homepage

`http://ontologydesignpatterns.org/wiki/Odp:KIELD2010`

Table of Contents

Keynote Talk

Research Papers

Late Breaking News

Ontology Engineering for Linked Data: What Makes for a Good Ontology?

Martin Hepp

Universität der Bundeswehr, Munich, Germany
mhepp@computer.org

The talk will discuss the impact of ontology design choices and ontology quality criteria on the overall impact of the linked data initiatives. In particular, it will analyze whether there are good and bad ontologies from a practical standpoint, what quality criteria will matter the most, and how this relates to the "raw data now" movement in the community.

Fine-tuning triplification with Semion

Andrea Giovanni Nuzzolese[1], Aldo Gangemi[1],
Valentina Presutti[1], Paolo Ciancarini[2]

[1] Semantic Technology Lab, ISTC-CNR, Rome. Italy
[2] Dep. of Computer Science, Università di Bologna. Italy

Abstract The Web of Data is fed mainly by "triplifiers (or RDFizers)", tools able to transform content (usually from databases) to linked data. Current triplifiers implement diverse methods, and are usually based on bulk recipes, which make fixed assumptions on the domain semantics. They focus more on syntactic than on semantic transformation, and allow for limited (sometimes no) customization of the process. We present Semion, a method and a tool for triplifying content sources that overcomes such limitations. It focuses on applying good practices of design, provides high customizability of the transformation process, and exploits OWL expressivity for describing the domain of interest.

1 Introduction

In the traditional hypertext Web, which can be identified by the expression "Web of documents", the nature of the relationships between two linked documents is implicit: the usual encoding language used i.e. HTML, is not sufficiently expressive to enable individual entities, described in a particular document, to be connected by typed links to other related entities [10].
In recent years, the Web has evolved from a global information space of linked documents to one where both documents and data are linked. Underpinning this evolution is a set of best practices for publishing and connecting structured data on the Web known as Linked Data [8]. The aim of Linked Data is to bootstrap the Web of Data by identifying existing data sets that are available under open licenses, converting them to RDF according to [8], and publishing them on the Web. There are commonly accepted solutions for transforming non-RDF data sources into RDF datasets. They rely on predetermined implicit assumptions on the domain semantics of the non-RDF data source. For example, relational databases are associated with ontologies where each table is a `rdfs:Class`, each table record is an `owl:Individual` of that class, and each table column is a `rdf:Property`, regardless the intensional semantics of the database tables, records, and columns as it was conceived in the original conceptual model of the database. Such implicit assumptions imply:

- limited customization of the transformation process (e.g. a user cannot map a table to a property)
- difficulty in adopting good practices of knowledge reengineering and ontology design (e.g. ontology design patterns [19])

– limited exploitation of OWL [14] expressivity for describing the domain, so that the services of OWL inference engines result to be sometimes limited.

The tool described here, Semion, implements a method that overcomes the above issues. The Semion method allows to reengineer any data source to RDF triples, without fixing assumptions on the domain semantics, which can be customized by the user. It is based on three main steps: (i) a **syntactic transformation** of the data source to RDF datasets according to an OWL ontology that represents the data source structure i.e. the source meta-model. For example, the OWL ontology for a relational database would include the classes "table", "column", and "row". The ontology can be either provided by the user, or reused from a repository of existing ones. The transformation is therefore independent from any assumption about the domain semantics. (ii) A **first refactoring** step that allows to transform the obtained RDF dataset according to a so called "mediator". A mediator is any ontology that represents the **informal semantics** that we use for organizing our knowledge, i.e. the semantics of human semiotic processes. The value of a semiotic representation is its ability to support the representation of different knowledge sources developed according to different underlying semiotic perspectives. In [11] some examples are provided of knowledge representation schemata, either formal or informal, which can be aligned to semiotic concepts and relations. However, there can be many ways of expressing such informal semantics. A popular example of a mediator is SKOS [16], which addresses the organization of knowledge by means of narrower/broader "concepts". This step is analogous to a reverse engineering action performed in order to get the knowledge out of the constraints of a specific data structure. In this paper, we focus on the use of the Linguistic Meta-Model (LMM) [11] as mediator ontology, an OWL-DL ontology that formalizes some semiotic relations. (iii) A **second refactoring** step that maps the RDF model expressed in terms of the mediator, to a formal language, e.g. OWL, also enabling custom semantic choices (e.g. relation as a logical class or relation). (iv) A **third refactoring** step that allows to express the resulting OWL model according to specific domain ontologies e.g. DOLCE, FOAF, the Gene Ontology, indicated by the user. This last action results in a RDF dataset, which expresses the knowledge stored in the original data source, according to a set of assumptions on the domain semantics selected and customized by the user.
The contribution of this paper is threefold: the Semion method, a tool that implements such method, and the tool evaluation. The evaluation has been performed by comparing Semion to existing tools that address similar requirements, and by applying it to a use cases i.e. triplification of WordNet database.
The paper is organized as follows: Section 2 discusses related work. Section 3 describes the Semion method, while Section 4 contains details regarding the Semion tool and its application to two use cases. Section 5 describes the results of a feature-based comparison of Semion with other related tools. Finally section 6 discusses conclusion and future work.

2 Related work

The wide spreading of Linked Data led to the development of several methods and tools for transforming non-RDF (legacy) data sources to linked data, and publish them on the Web of data. In this section we briefly describe the most popular and used ones.

D2R Server [9] is a tool for publishing relational databases on the Semantic Web. It enables RDF and HTML browsers to navigate the content of a database, and allows other applications to query a database through SPARQL. D2R Server uses the D2RQ Mapping Language to map the content of a relational database to RDF. A D2RQ mapping rule specifies how to assign URIs to resources, and which properties are used to describe them. The **Talis Platform** [3] provides Linked Data-compliant hosting for content, and its associated RDF data. Data held in the platform are organized into "stores" that can be individually secured if needed. The content and metadata become immediately accessible over the Web and discoverable using both SPARQL [20], and a keyword-based search engine. **Triplify** [7] is a PHP plug-in for Web applications. It implements a black-box recipe for making database content available on the Web as RDF, JSON or Linked Data. **Virtuoso** [1] combines functionalities of traditional RDBMS, ORDBMS, virtual database, RDF, XML, free-text, Web application server, and file server in a single system. It enables a single multithreaded server process that implements multiple protocols. **QuOnto** [6] is a Java-based tool for ontology representation and reasoning. It implements the DL-Lite family of ontology representation languages, and uses relational DBMSs to store the extensional level of the ontologies. It relies on the Ontology Based Data Access (OBDA), which provides access to heterogeneous data sources through an intermediate ontology. **METAmorphoses** [21] is a set of tools for flexible and easy-to-use generation of RDF metadata directly from a relational database. Metadata are generated according to the mapping from an existing database schema to a particular ontology. **Krextor** [15] is an extensible XSLT-based framework for extracting RDF from XML, supporting multiple input languages as well as multiple output RDF notations. A relevant project related to the topic of reengineering legacy data sources to RDF is the **RDFizer** [2] project, an on-line directory that collects accepted tools for converting data from various formats, i.e. BibTEX, Java, Javadoc, etc., to RDF.

Although such existing tools served successfully the requirement of bootstrapping the Web of Data, they share two limitations: *adaptability* to heterogeneous data source structures, and *customizability* of the transformation process.

Semion implements a transformation method aiming at triplifying heterogeneous content sources without such limitations. It aims at obtaining high-quality (in the sense of task-based, relevant, and semantically well-founded) data and ontologies, through a transformation process with explicit, customizable, and incremental steps.

3 Semion method

Figure 1 depicts the two key processes composing the Semion method. The first process i.e. the reengineering process, performs a syntactic transformation of the data source to RDF without making any choice with respect to neither the formal semantics to be used for expressing the domain knowledge (encoded by the datasource), nor the formal language to be used for encoding the (final) resulting dataset. RDF at this stage is only used as a serialization format. The second process i.e. the refactoring process, performs semantic-based transformations of the RDF dataset. During this process it is possible to choose a specific formal semantics e.g. model theory, and a specific formal language e.g. OWL2-EL, to be used for modeling the dataset (and associated ontology), which will be the result of the whole Semion triplification procedure. Finally, the refactoring process can be iterated once more in order to align the resulting ontology to existing ones that the user might want to use for his/her specific application.

3.1 The reengineering process

The reengineering process aims at producing a RDF dataset starting from a content source encoded in any possible format[1]. The goal of this process is to express, through RDF triples, the same (or a selection of) data that are stored in the data source. The approach followed by this process is mainly syntactic. In order to achieve such goal Semion requires, as input, a RDF vocabulary describing the data source meta-model e.g. for a relational database, a vocabulary containing concepts like table, field, query, result set, etc[2]. Additionally, it requires, as input, a set of rules mapping the vocabulary entities to the data source entities e.g. DB tables map to `rdfs:Resource` with `rdf:type` `mydb:Table`. The result of the reengineering process is a RDF dataset including both the data and the schema of the data source, encoded according to the adopted database vocabulary[3]. The domain semantics of the dataset is at this point implicit.
Depending on the amount of data to be reengineered, this process can be performed with incremental iterations. At the moment, once the RDF dataset is produced, it is independent on the original data source, i.e. any change applied to the original source after the reengineering step is not reflected (automatically) into the RDF dataset.

[1] Although Semion's current implementation supports relational database and XML sources, the theoretical method here described is designed in order to be applicable to any possible data source.

[2] An example of a relational database vocabulary used by Semion implementation can be downloaded at http://ontologydesignpatterns.org/ont/iks/dbs_l1.owl

[3] http://stlab.istc.cnr.it/software/semion/tool/samples/customerProductSchema.rdf and http://stlab.istc.cnr.it/software/semion/tool/samples/customerProductData.rdf are RDF datasets resulting from the reengineering of a sample database for storing and managing data about customers and products. Respectively, they express the schema and the data of the sample database according to the Semion default database vocabulary available at http://ontologydesignpatterns.org/ont/iks/dbs_l1.owl

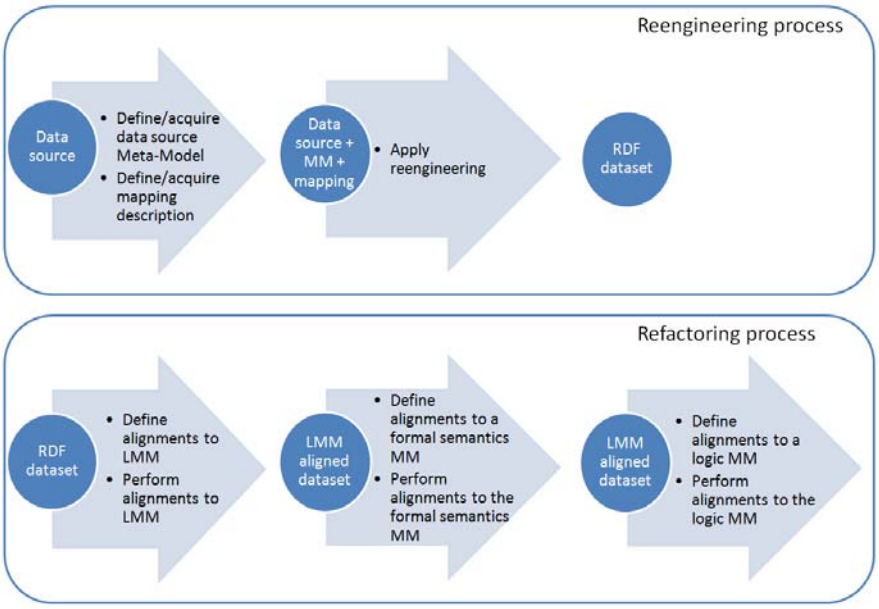

Figure 1. Tranforming method: key concepts.

3.2 The refactoring process

The final goal of the whole Semion triplification method is to obtain a RDF dataset modeled according to a certain formal semantics, encoded in a specific logic language, with its knowledge expressed according to axioms defined in a domain ontology. The reengineering process, explained above, produces a dataset containing the data of the original data source encoded in RDF format. The refactoring process allows us to further transform such dataset, by introducing formal and domain semantics through a number of steps in which formal and domain design choices are made explicit. The first step of the refactoring process, as depicted in Figure 1, consists of defining a set of rules that map the available RDF dataset to a so called "mediator" ontology such as the Linguistic-Meta Model (LMM) [18]. LMM is an OWL ontology describing the entities of the informal semantics that we use for organizing our knowledge in a semiotic-cognitive way. The core of LMM represents the main concepts of semiotics according to [17], namely:

– the **lmm:Expression**[4] class includes social objects produced by agents in the context of communication acts. They are natural language terms, symbols in formal languages, icons, and whatever can be used as a vehicle for

[4] lmm: is the prefix for http://www.ontologydesignpatterns.org/ont/lmm/LMM_L2.owl

communication. Expressions have a content (or meaning) and possibly a reference. For example, the word *dog* can have the meaning of a collection of animals in the utterance: *dogs are usually friendly to children*, and can refer to a specific dog in the utterance: *Viggo dog has played with my daughter this morning*. In this case, an additional semiotic relation holds, i.e. that Viggo dog is *interpreted by* the meaning *dog*;
- the **lmm:Meaning** class includes any entity that is supposed to be the content of an expression in a communication act, e.g. other expressions that explain an expression as in dictionaries, the cognitive processes corresponding to the use of an expression in context, the concepts in a classification scheme, are all examples of meanings;
- the **lmm:Reference** class includes any possible individual that is referred to by an expression, under a certain meaning;
- the **lmm:LinguisticAct** class includes the actual communication events, in which some agents use expressions to either convey meanings or refer to entities.

Now consider a relational database as an example data source. In this case, the refactoring to LMM could be performed according to the following mapping assertions:

Table(?x) → `lmm:Meaning`(?x)
Record(?x) → `lmm:Reference`(?x)

The previous assertions state that, starting from the database meta-model, and in the scope of the refactoring rules applied to a database *db1*,

- a table t_1 is the meaning of the table structure and vocabulary (that are expressions) defined for that table in *db1*
- a record r_1 in t_1 is the reference of the record structure and vocabulary (that are expressions) defined for that record within the table t_1. r_1 is the reference of the same expression that has t_1 as meaning, so that r_1 is *interpreted by* t_1.

Although Semion uses LMM as built-in mediator ontology, it allows the user to customize such choice. For example, the Simple Knowledge Organization System (SKOS) [16] can be used as a mediator. SKOS is a common data model for knowledge organization systems such as thesauri, classification schemes, subject heading systems and taxonomies. SKOS describes the typical informal (semiotic) semantics used by communities of practices familiar with classification schemes, for organizing their knowledge.
This refactoring step can be seen as a reverse engineering action that gets the knowledge out of the constraints of a specific data structure: the obtained RDF dataset is expressed in terms of semiotic entities.
The next refactoring step consists in aligning such dataset to a formal semantics i.e. semiotic entities are mapped to formal entities complying to a specific reference formal semantics e.g. model theory. In the previous example,

`lmm:Meaning` can be mapped to e.g. `forsem:Class`[5] or `forsem:Relation`, while `lmm:Reference` can be mapped to e.g. `forsem:SetElement` or (one or more) `forsem:Proposition`. The next iteration allows to choose a specific logic language, which complies to the specified formal semantics e.g. OWL (e.g. `owl:Class`, `owl:ObjectProperty`, `owl:NamedIndividual`, `owl:PropertyAssertion`, etc.). The user might want to merge the last two steps into one if the need is only to choose a logical language without making it explicit the formal semantics behind it. Nevertheless, Semion method allows a higher degree of customization in order to express the same formal semantics into different logical languages.

4 Semion tool

The method described in the previous section is implemented in a tool called Semion[6]. Semion is available both as reusable components organized in two main Java libraries (called SemionCore), and as a standalone graphical tool based on the Standard Widget Toolkit(SWT) [5] and JFace [4]. The tool has been designed according to the Model-View-Controller pattern, in which view and controller are part of the user interface, while the models are provided by the SemionCore libraries. Currently the tool provides support, and has been tested for transforming relational databases to RDF, however it has been designed in order to be easily extensible for supporting transformations of any kind of data source to RDF. Figure 2 shows the reengineering perspective of the Semion tool, according to the method terminology. In this perspective, the user is supported to transform a database to a RDF dataset according to a vocabulary describing the database meta-model; Semion provides a built-in vocabulary for such reengineering process[7]. The user can choose whether to reengineer the whole database by generating a single dump of RDF triples, or to reengineer only a selection of database entities e.g. a subset of tables and their records. Additionally, the interface provides a SPARQL [20] view, which allows to query the resulting RDF dataset.
The refactoring perspective supports the user in performing the refactoring process. Semion performs refactoring transformations according to a set of user-defined rules. Such rules can be expressed in terms of a simple syntax of the form:

$$antecedent \rightarrow consequent$$

Technically, such rules are interpreted and executed as SPARQL CONSTRUCT queries. For example, the following rule cause the transformation of resources of type *dbs:Table* to instances of the class *dul:Concept* of DOLCE Ultra Light [12]:

$$dbs : Table(?x) \rightarrow DUL : Concept(?x)$$

This rule is interpreted and executed as the SPARQL query:

[5] forsem: is the prefix for http://www.ontologydesignpatterns.org/ont/dul/FormalSemantics.owl
[6] http://stlab.istc.cnr.it/software/semion/tool
[7] http://ontologydesignpatterns.org/ont/iks/dbs_l1.owl

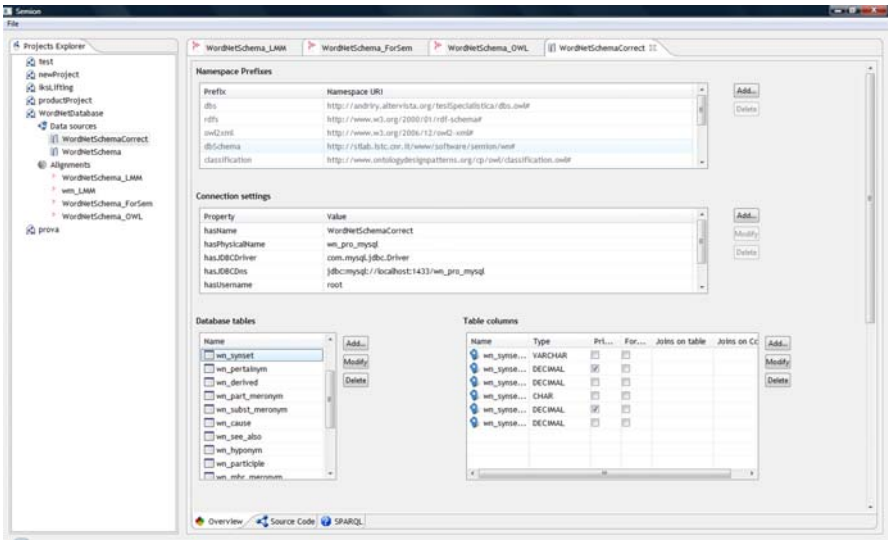

Figure 2. Semion tool: view of the reengineering interface.

CONSTRUCT { ?x rdf:type DUL:Concept. }
WHERE { ?x rdf:type dbs:Table. }

Semion tool has been applied for triplifying the WordNet database [8]. According to the Semion method, WordNet has been first transformed to RDF triples according to a defined vocabulary for RDB. The resulting dataset has been then transformed to a new one based on LMM by defining specific refactoring rules. Next, a refactoring step for fixing the formal semantics has been performed. Figures 3 and 4 show two screenshots of the tool interface during refactoring steps performed for the WordNet use case. Figure 3 deals with mapping WordNet LMM-based dataset to a vocabulary expressing a formal semantics, Figure 4 shows how the resulting dataset is mapped to the OWL vocabulary for obtaining an OWL ontology (including its individuals) for WordNet. Finally Figure 4 shows the screenshot of the tool while performing the last refactoring step, which transforms the dataset according to the OWL vocabulary.

5 Evaluation

Semion aims at maximizing flexibility of the transformation process: it wants to support both a user who wants to customize reengineering and refactoring of data sources, and a user who wants to reuse "good practices" such as recipes to convert databases, thesauri, etc., and does not want to know anything about the reengineering and refactoring clockwork. In addition to the WordNet use

[8] MySQL version of the WordNet database.

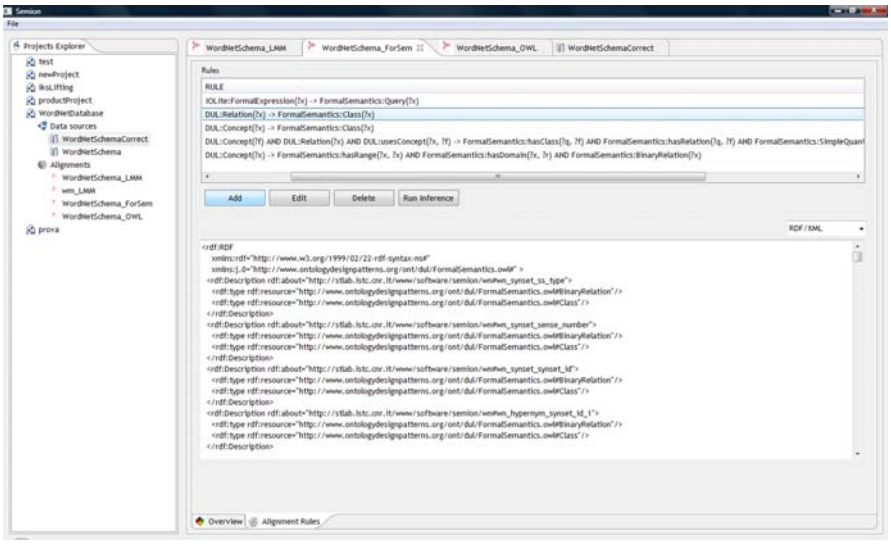

Figure 3. Alignment to the FormalSemantics vocabulary.

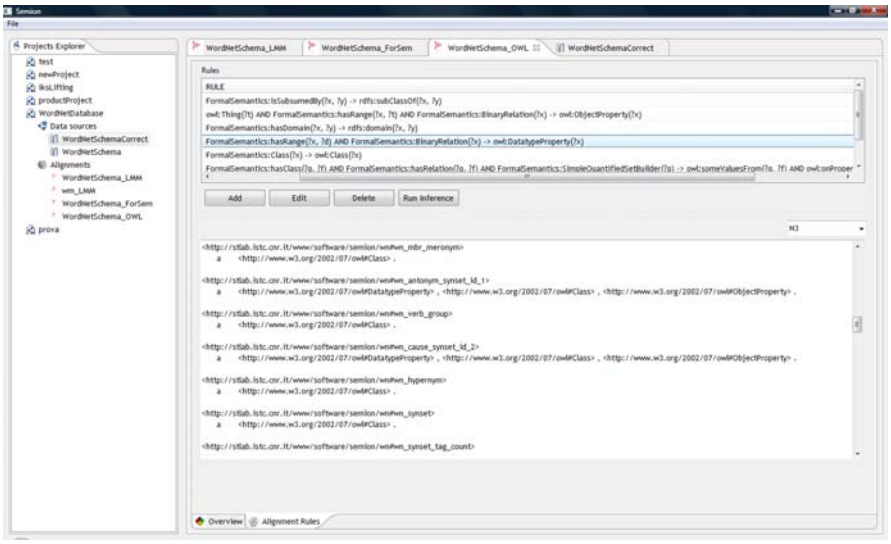

Figure 4. Alignment to the OWL vocabulary.

case presented in the previous section, we have performed a comparison of the Semion tool to other related tools according to a defined set of functionalities. Such functionalities are not meant to constitute an evaluation framework, they have been selected for emphasizing the characteristics that distinguish triplification methods, based on the issues that we have addressed in this paper i.e.

customization of the transformation process, adoption of good practices for data reengineering and domain modeling, exploitation of OWL expressivity. In the future, Semion will be subject of a more rigorous evaluation. The tools involved in such comparison are D2R [9], Triplify [7] and METAMorphoses [21]. The other tools mentioned in Section 2 have not been included because of the lack of availability of details about their design. We are also missing many other tools that provide support for reengineering data sources to ontologies, e.g. from XML databases, for HTML scraping, etc. This is because we could evaluate only the current implementation of Semion, which supports transformation of only relational database, hence we leave this aspect to future work.

D2R transforms relational databases in a different way with respect to Semion, since it allows to access the source as it was an RDF graph translating, through a mapping file, SPARQL queries into SQL queries. The mapping file can be configured, but the transforming choices are made implicit in the "bridges" that it realizes. Triplify reveals the semantic structures encoded in relational databases, but the transformation and the domain semantics is not configurable as it is in Semion. METAMorphoses has some similarity to Semion, since it allows to configure the mapping from the data based on a metamodel, and to map the result to another ontology. However, it is not clear if the mapping can be made by means of any (even customized) metamodel for the data source and if the second mapping can be applied iteratively to other ontologies.

Table 1 shows the functionalities selected for comparing Semion performances with the other tools'. The first two functionalities deal with the core business of triplifiers i.e. transforming legacy content to RDF datasets. Although the approaches are different, all tools are able to tranform non-RDF data sources to RDF. Nevertheless, it must be noticed that while Semion is transforms also the schema (and produces an ontology) of the source, the other tools do not. Alternatively, they keep a reference to the original schema e.g. a mapping, in order to access and extract data.

The transformation process is highly customizable in Semion, while is fixed in Triplify. D2R allows some degree of customization when defining the mapping rules, and METAMorphoses allows only customization of the domain ontology the transformation has to comply with.

Extensibility to support other source type is a Semion feature, it is implemented by providing the source metamodel ontology. The other tools, at the moment, do not consider support for other types of legacy sources but relational databases.

Triplification of the original data structure is a Semion' specific feature. It refers to the result of the reengineering process, which do not impose any semantic assumptions at domain level when triplifying the datasource i.e. using a metamodel approach such as SKOS' one, for any possible data source. The other tools include always some implicit semantic choices at domain level during the transformation e.g. the bridge approach described in Section 3.

Another Semion' specific feature is the support for incremental, iterative mapping of the dataset to custom ontologies. The definition and reuse of existing practices for reengineering i.e. transformation recipes, is supported by both

Semion and METAMorphoses. D2R partially supports this functionality because it is possible to specify references to a target ontology in the mapping definition. Furthermore Semion is compliant with the good reengineering practices as the Ontology Design Patterns (ODP) [13].
Finally, there are two Semion's special features. Semion is integrated with an inference engine i.e. it provides OWL reasoning support, and its implementation relies completely on semantic web standards and technologies. In other words Semion is itself a semantic web-based tool, everything that could be done by using standard technologies was implemented with such approach e.g. SPARQL, SWRL, etc., in order to minimize the amount of developed ad-hoc code for reasoning purposes.

Table 1. Functionalities implemented by Semion and comparison to other tools

Functionality	**Semion**	**D2R**	**Triplify**	**METAMorphoses**
Transform non-RDF data to RDF	yes	yes	yes	yes
Transform non-RDF data source schemata to RDF	yes	partly	partly	partly
Customization of the transformation process	yes	partly	no	partly
Extensibility to support other source types	yes	no	no	no
Triplification of original data structure	yes	no	no	no
Incremental, iterative mapping to custom ontologies	yes	no	no	no
Support for translation recipes definition and reuse	yes	partly	no	yes
OWL(2) and reasoning support	yes	no	no	no
Based on semantic web standards and technologies	yes	no	no	no

6 Conclusion and future work

We have presented Semion, a method and a smart triplification tool that is designed for transforming any non-RDF sources to RDF datasets. At the moment Semion supports only transformation of relational databases. The Semion method is divided into two processes. The first one allows a syntactic transformation of the datasource to RDF without making assumptions on the domain semantics. The second is an iterative process, and allows for a highly customizable transformation of the dataset based on formal and domain semantics. We have also applied Semion to a use case, and shown a feature-based comparison of it with related tools. Current and future effort is focused on extending the types

of legacy sources supported e.g. XML, latex, PNG, etc., on making it available under various forms e.g. restful services, on both experimental and user-based evaluation, and its usage in large use cases.

Acknowledgements

This work has been part-funded by the European Commission under grant agreement FP7-ICT-2007-3/ No. 231527 (IKS - Interactive Knowledge Stack).

References

1. OpenLink Software. Virtuoso Universal Server 4.5 Data Management and Integration Reviewer's Guide. http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/. Visited on January 2010.
2. Simile. RDFizers. http://simile.mit.edu/wiki/RDFizers. Visited on August 2010.
3. Talis Platform Data Connected. http://www.talis.com/platform/developers/. Visited on January 2010.
4. Eclipse. JFace. http://wiki.eclipse.org/index.php/JFace, Visited on January 2010.
5. Eclipse. SWT: The Standard Widget Toolkit. http://www.eclipse.org/swt/, Visited on January 2010.
6. A. Acciarri, D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, M. Palmieri, and R. Rosati. QuOnto: Querying Ontologies. In M. M. Veloso and S. Kambhampati, editors, *AAAI*, pages 1670–1671. AAAI Press / The MIT Press, 2005.
7. S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller. Triplify: Light-Weight Linked Data Publication from Relational Databases. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 621–630. ACM, 2009.
8. T. Berners-Lee. Linked Data. World wide web design issues, July 2006. Visited on August 2010.
9. C. Bizer and R. Cyganiak. D2RQ - Lessons Learned. *W3C Workshop on RDF Access to Relational Databases*, October 2007.
10. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
11. A. Gangemi. What's in a Schema? In C.-R. Huang, N. Calzolari, A. Gangemi, A. Lenci, A. Oltramari, and L. Prevot, editors, *Ontology and the Lexicon: A Natural Language Processing Perspective*, pages 144–182. Cambridge University Press, Cambridge, UK, 2010.
12. A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. Sweetening Ontologies with DOLCE. In *Proceedings of 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, volume 2473 of Lecture Notes in Computer Science, page 166 ff, Sigünza, Spain, Oct. 1–4 2002.
13. A. Gangemi and V. Presutti. Ontology Design Pattern portal. Available at: http://www.ontologydesign-patterns.org, 2008.
14. F. v. Harmelen and D. L. McGuinness. OWL Web Ontology Language Overview. W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210/.

15. C. Lange. Krextor An Extensible XML to RDF Extraction Framework. In S. Auer, C. Bizer, and G. A. Grimnes, editors, *Proc. of 5th Workshop on Scripting and Development for the Semantic Web at ESWC 2009*, volume 449 of *CEUR Workshop Proceedings ISSN 1613-0073*, June 2009.
16. A. Miles and S. Bechhofer. SKOS Simple Knowledge Organization System Reference. W3C working draft, W3C, June 2008. http://www.w3.org/TR/2008/WD-skos-reference-20080609/.
17. C. S. Peirce. *Hartshorne (Eds.) Collected Papers of Charles Sanders Peirce*. Harvard University Press, 1931.
18. D. Picca, A. M. Gliozzo, and A. Gangemi. LMM: an OWL-DL MetaModel to Represent Heterogeneous Lexical Knowledge. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.
19. V. Presutti and A. Gangemi. Content ontology design patterns as practical building blocks for web ontologies. In *ER '08: Proceedings of the 27th International Conference on Conceptual Modeling*, pages 128–141, Berlin, Heidelberg, 2008. Springer-Verlag.
20. E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. W3C recommendation, W3C, Jan. 2008. http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/.
21. M. Svihla and I. Jelinek. Benchmarking RDF Production Tools. In R. Wagner, N. Revell, and G. Pernul, editors, *DEXA*, volume 4653 of *Lecture Notes in Computer Science*, pages 700–709. Springer, 2007.

Ontological Analysis of Human Relations for Semantically Consistent Transformations of FOAF Data

Miroslav Vacura and Vojtěch Svátek

Faculty of Informatics and Statistics,
University of Economics
W. Churchill Sq.4, 130 67 Prague 3,
Czech Republic
vacuram|svatek@vse.cz

Abstract. The FOAF project has prominent importance for capturing human relations in Linked Data. We analyze the FOAF data structures and their extensions from the point of view of formal ontology and discuss problems inherent in its design. We also point out necessary considerations for transforming the FOAF data structures by supplying additional knowledge into them, while achieving/maintaining semantic consistency.

1 Introduction

The Linked Data initiative was started by Tim Berners-Lee as an architectural vision for the Semantic Web. It explores the idea of Semantic Web as puting emphasis on making links so both people and machines can explore the interconnected web of data. If the data are linked then "when you have some of it, you can find other, related, data" [1]. Just like in HTML where there are relationships and hypertext links between documents, the Linked Data initiative wants to encourage a similar approach in the case of general data content, described by RDF. The key requirements for Linked Data are quite simple:

1. Use URIs as names for things.
2. Use HTTP URIs so people can look up those names.
3. When someone looks up a URI, provide useful information, using standards (RDF*, SPARQL).
4. Include links to other URIs, so that they can discover more things.

Guidance provided by these general points was later extended by technical documents like [3] and [12], as well as conference overview papers like [5] and [4]. Linked Data can be now crawled with an appropriate browser following RDF links; a search engine can also search these information sources similarly to conventional relational databases. However, unlike HTML, which only provides a generic linking capability, links in Linked Data environment can have different

types: we can e.g. specify that one person is author of a paper, or that this person *knows* another.

In our paper we focus on the problem of 'injecting' additional knowledge into Linked Data. We provide a case study based on one of the key projects in Linked Data – FOAF [6]. We analyze this 'standard' from the point of view of ontological engineering, and provide guidelines for injecting knowledge while maintaining semantic consistency.

The rest of the paper is organized as follows: The next section brings the basic characterization of the FOAF project, its history and extensions. It also points out some basic issues and complexities. Section 3 analyses the formal ontological structure of the *relationship vocabulary* extension of FOAF, and Section 4 proceeds with a detailed analysis of properties this vocabulary defines. Section 5 investigates the possibilities of leveraging on the previous analysis for supplying additional structures to FOAF data while maintaining semantic consistency, and presents some transformations patterns that can be utilized for such a purpose. Finally, Sections 6 provides some conclusions acknowledgments.

2 Relation knows in FOAF

In this section we will discuss some problems related to the FOAF project and the 'knows' relation. The FOAF project is well known in the Linked Data community and the 'knows' relation is an intuitive relation well understood by everyone. Since 2004 there were more than 1 million FOAF documents and 79% of them utilized the knows property [7].

The Friend of a Friend (FOAF) project was started with the ambition of creating a Web of machine-readable pages describing people, the links between them and the things they do, work on, create and like.[1]

For us the most important property of FOAF is knows, defined as "a person known by this person (indicating some level of reciprocated interaction between the parties)." It is understood as property of a person, however it is defined clearly as *symmetric* relation, because the specification requires "some form of reciprocated interaction" and stresses that "if someone knows a person, it would be usual for the relation to be reciprocated" [6].

The word "knows" is vague, and the FOAF specification doesn't resolve this vagueness in any formal way. It is described in natural language in the basic FOAF specification, and any explication or formalisation is lacking. For at least partial disambiguation of what this relationship means we have to turn to the *relationship FOAF module* developed in 2002 by E. Vitiello.[2] The RDF schema of this module defines several subproperties of property knows: friendOf, acquaintanceOf, parentOf, siblingOf, childOf, grandchildOf, spouseOf, enemyOf, antagonistOf, and ambivalentOf.

Inclusion of some of these properties seems debatable. For example, if a person describes someone as his/her enemy, then the person surely knows this

[1] `http://www.foaf-project.org`

[2] `http://www.perceive.net/schemas/20021119/relationship/`

enemy; however, the opposite may not be true – one may not know his/her enemy. Also inclusion of such subproperty in a *friend* of a *friend* vocabulary seems counterintuitive, because the general intuition may be that knows is in the semantic context of FOAF a positive (or at least neutral) relation between people. Another problem may arise if we in an application formally define the knows relation as symmetric as suggested in the FOAF specification. The property enemyOf is clearly not symetric (it is asymetric). Properties like childOf are obviously antisymmetric, and defining an antisymmetric property as subproperty of a symmetric one is logically inconsistent. This just emphasizes the problem of vagueness of the term knows.

Since 2004 the relationship module has been modified to a more general *relationship vocabulary* and is continually maintained and enhanced.[3] The following subproperties were added: ancestorOf, apprenticeTo, closeFriendOf, collaboratesWith, colleagueOf, descendantOf, employedBy, employerOf, engagedTo, friendOf, grandparentOf, hasMet, influencedBy, knowsByReputation, knowsInPassing, knowsOf, lifePartnerOf, livesWith, lostContactWith, mentorOf, neighborOf, participant, participantIn, Relationship, worksWith, and wouldLikeToKnow. The *relationship vocabulary* is now based on OWL, as some of the properties are explicitly declared with regard to the OWL standard. Still, however, the *relationship vocabulary* has not become too popular. A quick survey using the Swoogle[4] search engine revealed that only less than 0.1% indexed FOAF documents use this extension.

We have not been able to find out whether any particular methodology was used for choosing these subproperties or they were added just ad hoc or based on suggestions by participants of FOAF-DEV mailing list.[5] The main limitation of the *relationship FOAF module* is that it consists of a fixed and very limited set of subproperties. This has been partially overcome by extending it and turning it into a generic vocabulary. The description of extended properties now includes some semantics with a more complex subproperty structure. Still, probably for backward compatibility, properties like childOf are considered to be subproperties of knows. The new property knowsOf, which is not symmetric, was introduced, although only recently (February 2010) its semantics was changed such that the asymmetric knowsOf is no longer subproperty of the symmetric FOAF property knows; now, correctly, knows is subproperty of knowsOf.

It could be also noted that x childOf y does not imply x knows y. Such consideration was not important when we thought of *relationship* as a module or extension of FOAF, but when considered as a generic vocabulary that could be possibly included in any complex knowledge or reasoning system then it is important that it should not lead to logically incorrect conclusions.

Similarly, a recent (February 2010) revision of the *relationship vocabulary* acknowledged that for distant descendants it may not be possible to know reciprocally each other, so the property descendantOf is no longer subproperty of

[3] `http://purl.org/vocab/relationship`
[4] `http://swoogle.umbc.edu`
[5] `http://lists.foaf-project.org/mailman/listinfo/foaf-dev`

knows. However, the editors failed to notice that we run into exactly the same problem when we consider the property grandparentOf and even parentOf. For a grandparent (and even parent) of a person could die before the person was born, so there are real-world cases where people could not know their children or grandchildren. This can also happen in some other special circumstances.

Another problem of the *relationship vocabulary* is the definition of domains and ranges of the described properties. The problem becomes visible in the case of properties like employedBy – both its domain and range are set to class Person. In a real-world scenario an entity that employs other persons is usually a legal entity – company, institute or some other type of organisation. This may be called *legal person*; however, in the FOAF vocabulary such kind of entity is represented by class Organisation, which is by explicit semantic statement disjoint with class Person. If we use this formalization then we cannot express that a (physical) person is employed by an organisation without introducing logical inconsistency into our system. The employment relation is in FOAF usually expressed by the property workplaceHomepage with range Document. Then this document can be related to an Organisation using the property homepage. It would be intuitive to say that if the workplaceHomepage of a Person is a Document that is the homepage of an Organisation that it implies that the Person is employedBy an Organisation. But this is impossible in the scope of *relationship vocabulary* semantics, which defines employedBy as relation between two (physical) persons.

The reason why we go into such depth with the analysis of the FOAF project and the related *relationship* vocabulary is to see what difficulties are there when we are to link these data to some even very little different semantic system. Logical relations between properties and subproperties are important because they are used even in the most simple reasoners and information aggregators like e.g. Tabulator [2].

3 Formal Ontological Structure of *Relationship Vocabulary*

The previous section comprised informal discussion of some problems identified in the FOAF relationship module and in its more recent *relationship vocabulary* extension. This section will only focus on the latter, and will provide a more detailed analysis of its ontological (and logical) structure.

If we are to process knowledge captured in the FOAF format, we have to properly understand its ontological structure. A failure to do so may result in introducing semantic inconsistency to the knowledge thus transferred to an application.

If we go through the list of terms defined in the *relationship vocabulary*, there is one thing that immediately catches one's attention. The majority of terms describes standard properties that have the class Person as both domain and range. An example is the property livesWith – a relation between two persons, in this case symmetric. But in the *relationship vocabulary* there are also three terms that don't fit within this description: participant, participantIn and Relationship.

The term Relationship designates a class rather than a property. The terms participant and participantIn designate two properties, in turn. The domain of property participantIn is class Person, but its range is class Relationship. In the case of property participant it is the other way around. Intuitively we would expect these two properties to be inverse of each other, however, this is not formally declared in the *relationship vocabulary*.

An interesting fact we observe is that *relationship vocabulary* does not include one ontology pattern for human relations but actually *two* of them.

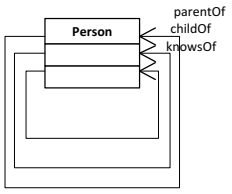

Fig. 1. Ontology pattern of *relationship vocabulary* 1

The first pattern that follows the legacy of the original FOAF is depicted in Figure 1. Human relations are defined as properties that have the class Person as both domain and range. This is formally just an extension of the original property knows – based on an idea that the new properties will be just subproperties of this property, thus maintaining "backward compatibility".

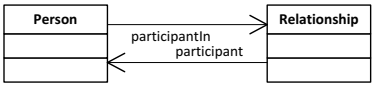

Fig. 2. Ontology pattern of *relationship vocabulary* 2

The second pattern introduces class Relationship that is described as a "class whose members are a particular type of connection existing between people related to or having dealings with each other." Based on this description it would seem that members of this class are reifications of *types* of relations – so we have one member per type of relation (note that we do not mean RDF reification here, but individuals representing types, i.e., indirectly, sets of other individuals). We have one member representing the relation "FriendOf", another representing the relation "livesWith", and so on. Let's take for example the relation "FriendOf" and we will call its reification FriendOf – it will be an instance of class Relationship. Now let's say that Petr and John (members of class Person) are friends:

$$\mathsf{participantIn}(\textsc{Peter}, \textsc{FriendOf})$$
$$\mathsf{participantIn}(\textsc{John}, \textsc{FriendOf})$$

Now let's say that Mary and Jane are also friends. So we can again add two assertions:

$$\mathsf{participantIn}(\textsc{Mary}, \textsc{FriendOf})$$
$$\mathsf{participantIn}(\textsc{Jane}, \textsc{FriendOf})$$

Now we have in our knowledge base four assertions (RDF triplets), but how no information on who is friend of who, see Figure 3. Such a structure only provides information about who is in any friendship relation at all, and seems therefore semantically inadequate.

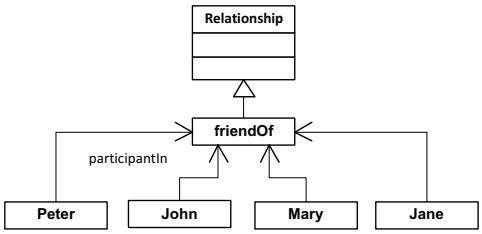

Fig. 3. Instance FriendOf of class Relationship and other instances.

We must conclude that for class Relationship to be of realistic use it must have members that are not reifications of *types* of relations but reifications of actual relations. This subtle ontological difference means that we have an instance of class Relationship for every individual relation. So relation FriendOf between Peter and John would be reified to instance FriendOf1 and relation FriendOf between Mary and Jane would be reified to instance FriendOf2.

$$\mathsf{participantIn}(\textsc{Peter}, \textsc{FriendOf1})$$
$$\mathsf{participantIn}(\textsc{John}, \textsc{FriendOf1})$$
$$\mathsf{participantIn}(\textsc{Mary}, \textsc{FriendOf2})$$
$$\mathsf{participantIn}(\textsc{Jane}, \textsc{FriendOf2})$$

The resulting ontological structure is in Figure 4. We can see that now we can still recognize who is friend of who. FriendOf1 can be easily recognized as reification of individual relationship of Peter and John, and similarly FriendOf2 can be recognized as reification of relationship of Mary and Jane.

Still it is hard to see how we can model asymmetric properties using this ontology pattern. Let's take for example fanOf – how can we describe that Peter is fan of Beethoven but Beethoven is not fan of Peter? We can perhaps use negative property assertions of OWL 2 (or the pattern-based approach for OWL 1 described in [9]), but this would go against the intended simplicity of FOAF, and the definition of *relationship vocabulary* never mentions such need for higher languages.

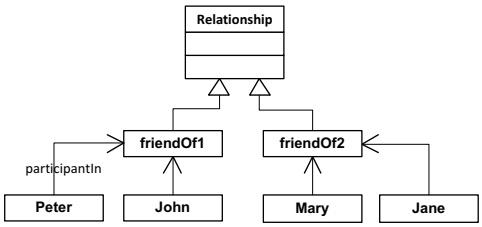

Fig. 4. Instances FriendOf1 and FriendOf2 of class Relationship and other instances.

It seems that the unclear and confusing definition of these class and properties belong to the reasons why they aren't more generally utilized. Using the Swoogle search engine we were not able to find any document, except various cached versions and copies of the original *relationship vocabulary* RDF, that would use these constructs. We believe that under such circumstances the maintainers of the *relationship vocabulary* should either review and rework these constructs and the relevant documentation or drop them completely.

4 Analysis of properties

The core of some of the problems we identified can be found in confusing the epistemic and ontological state of affairs.

- **Epistemic** state of affairs concerns with what is *known* to conscious agents. We may ask e.g. "Does x know that y is his/her child (enemy, neighbor, ancestor etc.)?" In none of these cases the answer is obvious and it may require further empiric investigation, which in this case consists in *questioning* of person x.
- **Ontological** state of affairs deals with what is matter of fact independently of knowledge (epistemic state) of particular conscious agents. We therefore ask: "Is matter of fact that y is child (enemy, neighbor, ancestor etc.) of x?" Again, in the case of such questions the answer may not be obvious and it may require empiric research, but usually *not* questioning but e.g. DNA test to find out if y is child of x. Then it may be found that "y is child of x" is true even if there is no knowledge (epistemic state) of this fact in either x nor y.

Principles of epistemic reasoning are usually formalized by *epistemic logic*, see e.g. [10]. Standard epistemic logic is based on introduction of notational convention Kxp, which we read as "x knows p", where x is a "knower" (i.e. conscious agent) and p is a proposition. The relation *knows* is problematic because of its vagueness – what exactly do we mean if we say "Person x knows person y"? What exactly does the person x know? What is the proposition p that s/he knows? We may use the approach inspired by [10, p. 6] and say $(\exists p)$ (p identifies person $y \wedge Kxp$) or in short form $(\exists p)$ $(p@I(y) \wedge Kxp)$ where $p@Q$

abbreviates "p answers question Q" and $I(y)$ is the question for identity of y. Using another approach we may conclude that the best way to formally model the vague *knows* relation is to model it by standard first-order predicate without epistemic extension and consider it normal empiric relation between two people.

Still these considerations about epistemic and ontological level of reasoning may provide us some help. For every predicate P in the *relationship vocabulary* we may ask whether the following proposition holds:

$$(\forall x)(\forall y)(P(x,y) \leftrightarrow Kx(P(x,y))) \tag{1}$$

That means that the relation P between persons x and y holds if and only if person x *knows* that relation P holds between persons x and y. This is a non-trivial assertion because while $(Kxp \rightarrow p)$ is the most general principle of epistemic logic, our proposition also says the reverse: that for a predicate P holds:

$$(\forall x)(\forall y)P(x,y) \rightarrow Kx(P(x,y)) \tag{2}$$

It is thus never the case that the assertion $P(x,y)$ evaluates to true without person x also knowing that it evaluates to true. Such feature of assertions may be true for some predicates but not for others. It means that such predicates are in a sense equivalent on epistemic and ontological level. Because of this we can refer to such a metaproperty as to 'being an *ontoepistemic* predicate'.

The predicates that do have such a feature are in many cases those describing our mental state. Such properties are usually called *mental properties* [11]. It might be true that ontoepistemic predicates are only mental predicates, however we are puzzled by properties such us apprenticeOf, which we believe are not pure mental (they may have social or institutional content) but still it seems unlikely that a person could be other person's apprentice without knowing it. We will postpone the solution of this theoretical problem to further investigation.

If we consider an ontoepistemic predicate then the domain of such a predicate is that of "knowers", and when the situation that $P(x,y)$ is true occurs then the knower x also knows it. Formally:

$$Oe(P) \equiv (\forall x)(\forall y)(P(x,y) \leftrightarrow Kx(P(x,y))) \tag{3}$$

E.g. the predicate *hates* is ontoepistemic because if x *hates* y then also x always knows that s/he hates y (this is an easy example because the *hates* property is mental). On the other hand the predicate *isFatherOf* is not ontoepistemic because there can be situations when x *isFatherOf* y but x does not know this.

We can now determine which of the predicates defined in the *relationship vocabulary* are ontoepistemic. We have also performed a detailed analysis of these predicates from the point of view of formal ontology. These results are presented along the *relationship vocabulary* definitions in Table 1. We also independently determined which of these properties are symmetric, asymmetric and antisymmetric, and compared the results with the *relationship vocabulary* definitions. In the first column there is the name of the property, the second presents what

superproperties this property has in the *relationship vocabulary* (we omitted differentFrom because it is defined as superproperty for all properties). The third column presents superproperties as based on our analysis. The column *Ontoepistemic* defines whether the property has this metaproperty. Column *RV sym.* contains information about symmetricity as defined in the *relationship vocabulary*, and the last column *Sym.* includes our results for symmetricity.

Property	**RV Super-prop.**	**Super-prop.**	**Ontoepistemic**	**RV Sym.**	**Sym.**
acquaintanceOf	k, kO	k, kO	yes	sym.	sym.
ambivalentOf	-	kO	yes[6]	-	asym.
ancestorOf	-	-	no	-	antisym.
antagonistOf	k, kO	kO	yes	-	asym.
apprenticeTo	k, kO	k, kO	yes	-	antisym.
childOf	k, kO	-	no	-	antisym.
closeFriendOf	k, kO	k, kO	yes	sym.	sym.
collaboratesWith	k, kO	k, kO	yes	sym.	sym.
colleagueOf	k, kO	-	no[7]	sym.	sym.
descendantOf	-	-	no	-	antisym.
employedBy	k, kO	kO	yes	-	asym.
employerOf	k, kO	-[8]	no	-	asym.
enemyOf	k, kO	kO	yes	-	asym.
engagedTo	k, kO	k, kO	yes	sym.	sym.
friendOf	k, kO	k, kO	yes	sym.	sym.
grandchildOf	k, kO	-	no	-	antisym.
grandparentOf	k, kO	-	no	-	antisym.
hasMet	k, kO	k, kO	yes[9]	sym.	sym.
influencedBy	-	-	no[10]	-	asym.
knowsByReputation	-	kO	yes	-	asym.
knowsInPassing	k, kO	kO	yes	-	asym.
knowsOf	-	kO	yes	-	asym.
lifePartnerof	k, kO	k, kO	yes	sym.	sym.
livesWith	k, kO	k, kO	yes[11]	sym.	sym.
lostContactWith	k, kO	kO	yes	sym.	asym.
mentorOf	k, kO	k, kO	yes	-	antisym.
neighborOf	k, kO	-	no	sym.	sym.
parentOf	k, kO	-	no	-	antisym.
siblingOf	k, kO	-	no	sym.	sym.
spouseOf	k, kO	k, kO	yes	sym.	sym.
worksWith	k, kO	-[12]	no	sym.	asym.
wouldLikeToKnow	-	kO	yes	-	asym.

Table 1. Properties of *relationship vocabulary*

[6] The definition says that x "has mixed feelings or emotions" towards y. We suppose that a conscious agent is aware of his/her feelings or emotions. Therefore s/he also knowsOf the person towards whom s/he has these emotions.

We have seen in Section 2 that according to recent update of *relationship vocabulary* property knows is now subproperty of knowsOf. We also know that property knows is symmetric while property knowsOf is asymmetric. The brief look at Table 1 reveals that these refinements were not reflected in the descriptions of properties. Properties that are asymmetric cannot be subproperties of symmetric property knows. This is an easy conclusion. More importantly – properties that are not ontoepistemic are, from point of view of formal ontology, not subproperties of property knowsOf. If a property P is not ontoepistemic then $P(x, y)$ does not imply that x knowsOf y. Again an example of such property P may be parentOf. However it is not necessary to think of such an issue as of mistake. We will show how to deal with it in the next section.

5 Supplying Additional Structures to Descriptions of Human Relations

A practical scenario for applying the previous ontological analysis is that of designing an application that would exploit FOAF data beyond their typical context (such as navigational browsing or social network visualisation). As we pointed out, due to problematic assumptions and implicit knowledge, such data could become semantically inconsistent when linked to other data; they should therefore undergo transformations. We are interested in transformations of linked data on human relations, mainly consisting in enriching them with additional information, which is implicitly present in the vocabularies.

We want to maintain *semantic consistency*, i.e. assure that the semantics of the data before and after transformation remains the same. These issues related to consistency may be classified to several categories based on their characteristics.

There are less important issues that we may characterize as typos or mere omission. These probably didn't have any impact on data created so far and are of merely of formal importance. In the definition of properties there is no differentiation between properties that are not symmetric and those that are antisymmetric. Also a more precise natural language description of relationship (maybe with examples) should be sometimes useful for general users to differentiate between such properties as collaboratesWith, worksWith, colleagueOf and

[7] The definition says: "A property representing a person who is a member of the same profession as this person." We suppose that usually people don't necessary know all people who are members of the same profession. It is also different from relation collaboratesWith, which requires symmetric knowledge of both persons involved.

[8] An employer who has thousands of employees usually does not know each of them.

[9] We understand this 'has met' as at least 'having been introduced to', i.e. not just 'having occurred at the same place in the same time'.

[10] A person doesn't necessarily know that s/he was (in his/her work etc.) been influenced by someone else.

[11] We understand it as a social relation, so it is ontoepistemic.

[12] This relation is defined as "a property representing person who works for the same employer as this person". This does not imply that they know each other.

similar. Our research using Swoogle revealed that some users are confused by these properties and use them incorrectly. Using inappropriate property simply because of confusion may introduce unnecessary semantic inconsistency to data.

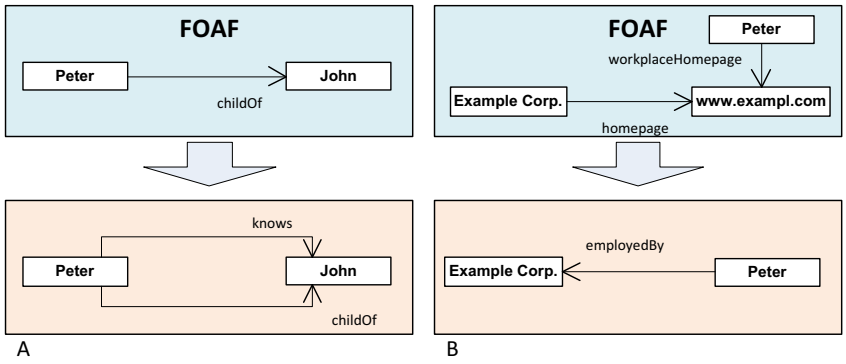

Fig. 5. Transformation patterns.

When considering FOAF data in the context of a different, semantically sounder ontology, it is not necessary to e.g. understand FOAF's internal declaring of properties like childOf subproperty of knows to be an ontological engineering mistake. Rather we could understand it as stating some additional knowledge. While from the point of view of formal ontology the relation childOf does not imply the relation knows, we propose that we should approach FOAF formal property definitions as stating a specific kind of *prior* knowledge: we should understand the statement describing childOf as subproperty of knows as declaring that whenever we have a FOAF statement that (x childOf y) we implicitly assert that (x knows y). If we accept such understanding then we could use our Table 1 as basis for developing transformation patterns that can be used to supply additional structures to FOAF data, without committing to FOAF modelling in general (for data coming from other namespaces). The differences between columns *RV Super-prop.* and *Super-prop.* may help us identify implicit *a priori* knowledge that has to be taken care of when performing such transformation. An example of a simple transformation pattern that makes implicit knowledge explicit is in Figure 5A. We can also say that according to our analysis properties that are ontoepistemic are subproperties of property knowsOf, so when using FOAF data in an application we should use a similar appropriate pattern or at least check whether the target data structure reflects such a priori constraints.

Another transformation pattern can be easily designed to overcome some semantic limitations of FOAF mentioned in Section 2. An example of such pattern is in Figure 5B. Here we concatenate two properties into another one, i.e. infer the 'employment' relationship from the 'mediating' webpage.

Finally, we may also proceed somewhat the other way around: 'unfold' a complex relationship from a simple FOAF relationship. For example, many

FOAF relationships, such as knowing a person by having met him/her, or being someones collaborator on a project, can be modelled by an *event-participation pattern*. Such an 'unfolding' transformation, relying on additional hints, may be used to disambiguate or enrich the semantic content of relations, based on transformation-based reference to complex content pattern reflecting the internal structure and semantics of the relation. Similarly as suggested above, the formal characteristics in Table 1 may be used for extended checking of semantic consistency during knowledge transformation or injection.

6 Conclusions

We have analyzed FOAF and its extensions for describing human relations from point of view of formal ontology. We focused on the property `knows` and pointed out some important issues. We also analyzed the ontological structure of the *relationship vocabulary* extension of FOAF, and identified some confusing ontological definitions. Detailed analysis of its properties revealed some interesting characteristics and assumptions that we believe are not generally valid. We pointed out that these could be understood as *a priori* knowledge and when exploiting FOAF data in an external context we must use appropriate transformation patterns. We have also presented examples of such transformation patterns and formulated directions of following research.

Acknowledgments

This work has been partially partially supported by the IGS 4/2010 and by the CSF grant no. P202/10/1825 (PatOMat - Automation of Ontology Pattern Detection and Exploitation).

References

1. Tim Berners-Lee. LinkedData. `http://www.w3.org/DesignIssues/LinkedData.html`, 2009.
2. Tim Berners-Lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd International Semantic Web User Interaction*, 2006.
3. Christian Bizer, Richard Cyganiak, and Tom Heath. How to publish linked data on the web, 2007.
4. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.
5. Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked data on the web (ldow2008). In Huai et al. [8], pages 1265–1266.
6. Dan Brickley and Libby Miller. FOAF Vocabulary Specification 0.97. `http://xmlns.com/foaf/spec/`, 1 2010.

7. Li Ding, Lina Zhou, Tim Finin, and Anupam Joshi. How the semantic web is being used: An analysis of foaf documents. In *In Proceedings of the 38th International Conference on System Sciences*, 2005.
8. Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang, editors. *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*. ACM, 2008.
9. Olaf Noppens. Negative property assertion pattern (npas). In *Proc. 1st ISWC 2009 workshop on Ontology pattern (WOP), Chantilly (VA US)*, 2009.
10. Nicholas Rescher. *Epistemic logic: a survey of the logic of knowledge*. Univ of Pittsburgh Press, 2005.
11. David Robb and John Heil. Mental Causation. In Edward N. Zalta, editor, *Stanford Encyclopedia of Philosophy*. 2008.
12. Leo Sauermann, Richard Cyganiak, and Max Völkel. Cool uris for the semantic web. Technical Memo TM-07-01, DFKI GmbH, Deutsches Forschungszentrum f ür Künstliche Intelligenz GmbH, February 2007. Written by 29.11.2006.
13. Vojtech Svatek, Ondrej Svab-Zamazal, and Valentina Presutti. Ontology naming pattern sauce for (human and computer) gourmets. In *Proc. 1st ISWC 2009 workshop on Ontology pattern (WOP), Chantilly (VA US)*, 2009.

GuessWhat?!

Human Intelligence for Mining Linked Data

Thomas Markotschi and Johanna Völker

KR & KM Research Group
University of Mannheim, B6 26, 68159 Mannheim, Germany

Abstract. Ontologies are an important prerequisite for an increasing number of knowledge-intensive applications, not to mention the great vision of the Semantic Web. However, despite the obvious need of such formal and explicit representations of knowledge, many people refrain from investing into the tedious and time-consuming task of ontology engineering. At the same time, purely automatic means for ontology construction so far have failed to meet our expectations in terms of quality and expressivity. In this paper we describe *GuessWhat?!*, a multi-player online game in the tradition of *semantic games with a purpose*. By leveraging people's play instinct it motivates them to contribute to the creation of formal domain ontologies from Linked Open Data. We detail on the implementation of the game and present the results of an initial user study.

1 Introduction

In 2001, Tim Berners-Lee [1] introduced the term *semantic web* in order to refer to what is now perceived as the future of the internet: a web of machine-interpretable content that can be processed by automatic agents in a meaningful way. Since achieving this ambitious goal requires both an explication and formalization of relevant domain knowledge, ontology languages such as RDFS [3] and OWL [11] have emerged as a means for unambiguous knowledge specification. However, the realization of the semantic web as envisioned by Tim Berners-Lee and the wide-spread use of intelligent, reasoning-based applications is still hampered by the lack of ontological resources.

The vast amount of *linked data*[1] in the form of RDF triples which is out there on the internet can be considered an important step forward on the way to the semantic web. In fact, a huge number of mashups and applications already benefit from billions of triples in the repositories of DBpedia[2], Freebase[3] or the like. At the same time, several applications, especially in the complex domains of medicine or bioinformatics, demand for more formal and expressive knowledge representations which are highly accurate in terms of syntax and semantics – a crucial prerequisite for logical inference yielding non-obvious conclusions. Constructing such representations, i.e. ontologies, of sufficient quality, size and expressivity is a very challenging endeavor. Making high

[1] `http://linkeddata.org`
[2] `http://dbpedia.org`
[3] `http://www.freebase.com`

demands on scarce human resources and the expertise of ontology engineers it is extremely expensive and time-consuming. While sooner or later automatic approaches to ontology construction (*ontology learning* [5]) could help to overcome this knowledge acquisition bottleneck, these approaches so far have failed to meet the expectations of people who argue in favor of powerful knowledge-intensive applications.

Semi-automatic approaches leveraging human intelligence and the *wisdom of the crowds* seem a particularly promising way to increase the efficiency and effectiveness of knowledge acquisition. A pioneer in the field of crowdsourcing [12] was Luis von Ahn who suggested to exploit the play instinct of humans for computationally difficult tasks by so-called *games with a purpose* [20]. His ideas were later taken up by Siorpaes and Hepp [17] who created the first *semantic* games with purpose: multi-player online games as incentives for human participation in the acquisition of formal and explicit representations of knowledge.

In this paper, we present *GuessWhat?!*, a novel semantic game with a purpose which leverages both human intelligence and collaboratively created data for bootstrapping the semantic web. *GuessWhat?!* motivates people to contribute to the creation of a domain ontology: Presented with class expressions such as `fruit AND yellow AND grows on tree` automatically generated from Linked Open Data the players have to invent as quickly as possible a suitable class name (`banana` or `lemon`, for example). This can be quite challenging as the generated descriptions, which are fairly general in the beginning (e.g. `fruit`), become more and more specific as the game proceeds (e.g. `fruit AND yellow`). As soon as a player cannot think of a suitable label anymore, he or she has lost the round, and finally, the player who after multiple rounds, has come up with the highest number of plausible class labels wins the game. Note that the rules of this game are inspired by a well-known card game.[4] We modified them in order to enable the verification and labeling of automatically created class expressions by people who do not even need to be ontology experts. Initial user studies give raise to the hope that *GuessWhat?!* will make *semantic web mining* [18] a lot more fun in the future.

The remainder of this paper is structured as follows. Section 2 gives an overview of related work in the field of automatic and semi-automatic knowledge acquisition. In Section 3, we outline the rules and implementation of our semantic game with a purpose, *GuessWhat?!*. Section 4 describes the results of our evaluation experiments, and finally, we conclude with a summary and an outlook to future work (cf. Section 5).

2 Related Work

Aiming at the semi-automatic acquisition of terminological knowledge from linked data, we find our approach related to a considerable amount of work on *ontology learning* [5], i.e. the automatic or semi-automatic generation of ontologies by machine learning or natural language processing techniques. The vast majority of existing methods have been developed to facilitate the extraction of ontologies from unstructured text [4], but only few of them support the acquisition of logically complex class expressions [19]. This also holds for early attempts to generate ontologies from linked data, e.g.,

[4] *Ein solches Ding* by Urs Hostettler (1989)

by means of systematic generalization [7], clustering [13] of RDF data, or more recent work on selective ontology reuse [15].

Other logical approaches based on *Inductive Logic Programming* (ILP) [6, 8] combine machine learning and logic programming techniques in order to derive class expressions from positive and negative examples (e.g. individuals known to instantiate the target class). Although ILP-based methods have already been shown to yield good results when applied to linked data [9, 10], most implementations are inferior to statistical approaches in terms of scalability and robustness. Moreover, ILP is not per se an interactive approach – a fact that makes it very difficult for these techniques to handle incomplete or incorrect knowledge at runtime. An alternative to the automatic generation of class expressions are natural language interfaces allowing users to interact with an ontology editor by means of controlled natural language (e.g. [2]). The drawback of these approaches is that people have to invest into learning a syntactically and lexically restricted language. Therefore, strong incentives might still be required in order to motivate people to formalize knowledge.

One of the strongest incentives is money or any type of financial benefit, as witnessed by *crowdsourcing* applications such as *Amazon Mechanical Turk*.[5] This service provides programmers with the opportunity to create so-called *Human Intelligence Tasks* (HITs), i.e. tasks that are not yet solvable by purely computational means. Such HITs can be anything from choosing the best category for a specific product over validating addresses to a fun quiz about celebrities. Other applications use "cheaper" incentives like fun and entertainment to attract people. *Games with a purpose* first introduced by Luis von Ahn [20] have been invented in order to leverage the play instinct of humans for tasks such as image labeling, solving captchas or the tagging of audio or video files. The ideas of von Ahn were picked up by Siorpaes and Hepp [16] who pioneered the field of *semantic games with a purpose* by suggesting to turn ontology acquisition into a fun game. One of their games, *OntoPronto*, motivates people to link Wikipedia articles to concepts of an upper-level ontology, while another one has been invented to facilitate the annotation of YouTube videos with respect to their genre or language.[6] Even more games are currently being developed in the EU project *Insemtives*.[7]

3 GuessWhat?!

In the following we will elaborate on the design and implementation of *GuessWhat?!*, a novel game with a purpose that leverages human intelligence for mining linked data. After introducing the rules of the game (cf. Section 3.1), we will turn to the software architecture and describe in detail the algorithms underlying the computational intelligence of *GuessWhat?!* (see Section 3.2).

3.1 Rules of the Game

GuessWhat?! can be played with a minimum of at least two players and currently has no limitation on how many users are allowed to participate. Each gaming session consists

[5] `http://www.mturk.com`
[6] `http://www.ontogame.org`
[7] `http://www.insemtives.eu`

of one or more *rounds* in the course of which the players have to guess the name of an unknown concept partially described by the game engine. Note that in most cases there will not be *one* correct answer, but many possible solutions – namely all the concepts which match the given description. When a round has ended, the players evaluate each other's answers in terms of plausibility, before starting with a new round and a new concept description.[8] More specifically:

Guessing: At startup, the players are presented with a partial description of a concept like, for example, `tangible`, `animal` or `used for transporting people`. Now, each player is asked to think of a "fitting object", i.e. a concept which matches the description, and to enter its name into the user interface. Alternatively, a player may choose "pass" in order to indicate that he or she does not know what the description might refer to. When every player has given an answer, the initial description is extended in a way that it becomes more specific (e.g. `animal AND carnivore`), and again each participant in the game needs to come up with a plausible label for the class denoted by the description. A *round* ends when either every player passed on the same description (e.g. nobody can imagine something that is `animal AND carnivore AND NOT dangerous AND poisonous`) or a previously defined maximum description length has been reached.

Evaluation: In the subsequent evaluation phase the players are asked to judge the final answers of their opponents. In particular, a player has to decide for each concept name entered by an opponent whether or not it fits the class expression that has been generated by *GuessWhat?!* until the moment when the round ended. The possible choices are accept ("OK"), reject ("Not OK") or abstention ("I don't know"). If he or she decides to reject an answer, the evaluator has to specify which part of the class expression conflicts with the given answer (see Figure 1). After the evaluation phase, a new round with a fresh description begins. To hold up a certain game flow, the last player who has not finished his task (i.e. answering or evaluating) is faced with a ten second timeout. If he fails to beat the clock he automatically "passes" or chooses "I don't know" in the evaluation phase.

The development of *GuessWhat?!* was motivated by the lack of formal terminological knowledge on the semantic web. The players' answers during the various game rounds and subsequent evaluation phases give us the opportunity to not only obtain valuable feedback with respect to the meaningfulness of the generated class expressions (as we will see in Section 3.2, these are automatically generated from linked data), but they also enable us to link complex descriptions to atomic concepts in an ontology. Note that the expressivity of the class descriptions generated by *GuessWhat?!* is not limited to conjunctions. Imagine, for instance, that during one of the rounds, three definition fragments `tangible`, `fruit` and `yellow` have been presented to the participants of *GuessWhat?!* altogether forming the class expression `tangible AND fruit AND yellow`. Further let us assume that the final answers of the players are `banana`, `lemon` and `cherry`. Now, during the evaluation phase that follows, the first two of these answers could be accepted as both bananas and lemons match the proposed description. The last answer, `cherry`, should rather be rejected as it is

[8] In the remainder of this paper, we will occasionally use the OWL terminology and refer to these (semi-formal) descriptions of concepts as *class expressions*.

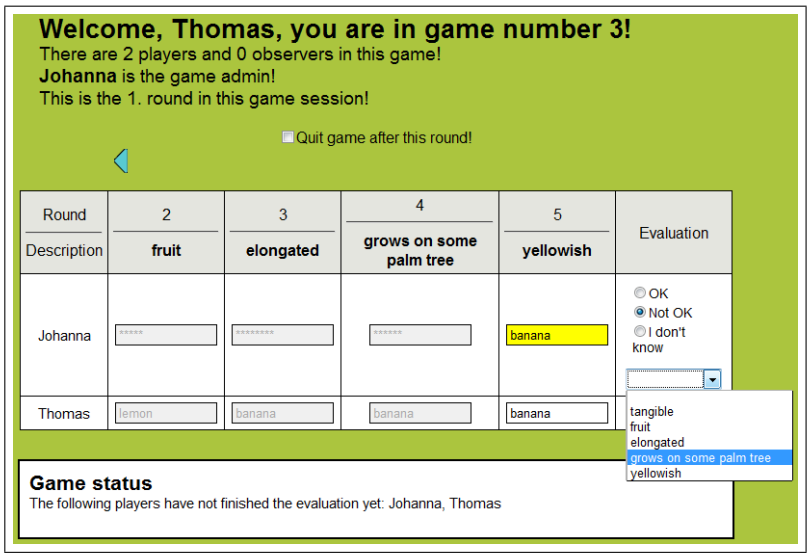

Fig. 1. A screenshot of *GuessWhat?!* taken during an evaluation phase.

not `yellow`. If one of the players notices this mismatch between something being both yellow and a cherry, and explains his judgement accordingly (i.e. by selecting one or more[9] parts of the description which contradict the other player's answer) we can not only conclude that `banana` and `lemon` are `tangible AND fruit AND yellow`, but also that `cherry` must belong to the class `tangible AND fruit AND NOT yellow`. Further screenshots as well as detailed instructions concerning the user interface of the game can be found online.[10]

3.2 Implementation

We developed and implemented the game in Java. Figure 2 shows the layered architecture of the game that runs on an Apache Tomcat 6.0 web server. The *data layer* consists of a Sesame RDF Store and a MySQL database. The connectors for these data stores can be found in the *data access layer* which also contains several components for gathering RDF triples from external semantic resources. The definition mining implementation in the *business logic layer* accesses the collected data, stores it in internal repositories and generates a class expression. The two beans which also belong to this layer are responsible for handling the user inputs which are made via the graphical user interface. In the following, the most essential components are discussed in more detail. For further details, see the extended version of this paper [14].

Data access and storage. For the creation of each class expressions we use a "seed concept" that serves as a starting point of the data gathering process.[11] This way we make

[9] The next version of the user interface will allow for multiple selection.

[10] `http://nitemaster.de/guesswhat/manual.html`

[11] In our experiments, these concepts were picked by hand but they could also be chosen randomly from a dictionary or an existing ontology.

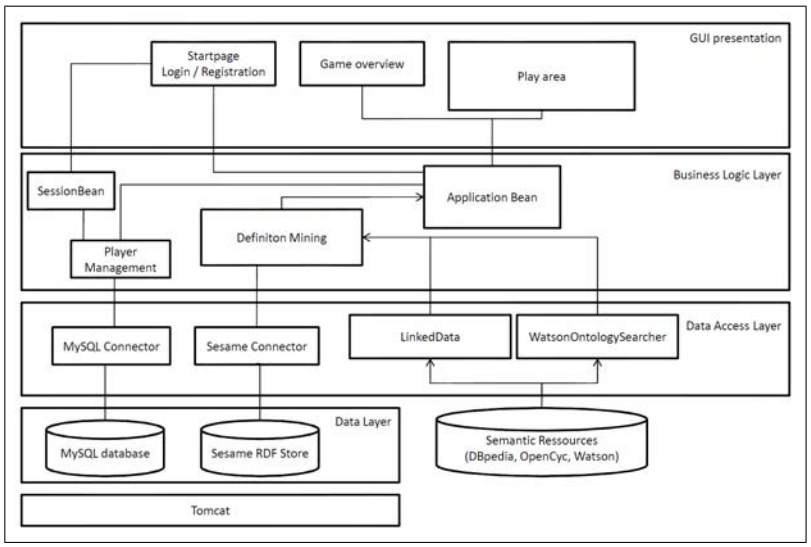

Fig. 2. The architecture of *GuessWhat?!*.

sure that the generated class expressions are mostly meaningful as otherwise people might be bored with a lot of nonsense descriptions. The collection of data from external resources such as *Linked Open Data* is mainly done via a local SPARQL endpoint and consists of several steps:

1. For each of the seed concepts (e.g. `banana`), try to find a matching URI in *DBpedia*, *Freebase* and *OpenCyc*.
2. Gather as much information as possible (e.g. superclasses, object properties) about the concept by querying related (i.e. linked) RDF repositories.
3. Store the gathered information it in a repository for faster access.

Definition mining. This component is responsible for generating class expressions from the individual pieces of information collected by the data access component described further above. The procedure of assembling the various bits and pieces collected from the various sources into a coherent description requires several steps:

1. Analyze the labels and URIs of the superclasses and properties that were retrieved before by means of simple natural language processing. The purpose of this step is to identify expressions which can be translated into logical operators (e.g. negation or disjunction), as well as to break down complex descriptions (e.g. long class labels found in OpenCyc) into smaller fragments. This way, we can avoid redundancy when assembling the overall class expression and compute more meaningful statistics for ranking the various aspects of a concept's description.
2. Judge the smaller fragments with respect to *generality* and *confidence* (i.e. relevance as to the seed concept). This information is required to ensure that the individual parts of a description become more specific as a round goes on and players are not presented with overly specific descriptions (e.g. `prepared from some`

tea leaves) in the beginning or very high-level descriptions (e.g. tangible) at the end of a round.

For example, consider the superclass an elongated yellowish fruit which was found during the search for information about the seed concept banana. Using the LExO [19] approach, we can split the class label into elongated, yellowish and fruit. In order to compute the confidence and generality scores of these fragments, the extracted data is joined in one big tree structure. The graph mining algorithms applied to this structure take into account the following aspects:

- How often was a class (e.g. an elongated yellowish fruit) or property found during the search for information about a concept?
- How often is every single fragment of its description (e.g. elongated, yellowish, fruit) present in the result set?
- How many paths from the seed concept to the root node (i.e. owl:Thing) does a class or property lie on?
- What is the distance of a class or property to the seed concept?

The first three factors are expressed as values between 0 and 1. By averaging them we obtain a value which we refer to as "confidence". The higher this value is, the more certain it is that the class or property it belongs to is a good description of the seed concept. The forth aspect in the enumeration above is referred to as "generality" and also ranges between 0 and 1. The higher this value is, the more specific is the respective fragment of the concept description. From both confidence and generality we compute, for each fragment of a description, an overall score which changes as the game proceeds: Imagine for example the seed concept banana and the two fragments tangible and fruit. The confidence of tangible is rather high (as it was found quite frequently) while its generality score is comparatively low (i.e. it is not very specific). For fruit it is the opposite. Now, in the beginning of the round ($step$ = 0), tangible is favored over fruit, that comes into play later when $step$ approaches $step_{max}$. This way of balancing confidence and generality is expressed by the following formula:

$$score(c, step) = (step_{max} - step) * confidence(c) + step * generality(c) \quad (1)$$

User interface. The user interface has been fully designed in XHTML and uses some components of the *ICEfaces*[12] framework. The latter also includes an AJAX Push implementation which is used to exchange data with the server in near real-time and to update the graphical user interface on the client-side. The execution of the game logic is handled by two independent types of beans. The application bean implements the Singleton pattern and is only initialized at its first access. It coordinates everything concerned with the game execution such as managing players, game creation and handling inputs. Additional session beans, which are initialized for every connected user, are responsible for the login and registration.

[12] http://www.icefaces.org

4 Evaluation

We will now summarize the user feedback which we gathered throughout the implementation and testing process (cf. Section 4.1), before taking a closer look at the results of these test sessions (see Section 4.2). The complete data set acquired during the evaluation of *GuessWhat?!*, including the automatically generated class expressions as well as the players' answers and the ontology constructed thereof is available online.[13]

4.1 Gaming Experience

In order to evaluate the gaming experience and the incentives created by *GuessWhat?!*, we scheduled several test sessions with different groups of people – ontology experts as well as users without any prior knowledge about semantic technologies.

First, two "beta tests" were conducted with 5 players participating in each of them. Afterwards, we asked all of the participants for their experiences throughout the game. They complained about the description fragments in the game being too complex and they told us that many of those did not make much sense, and we were surprised to see that the players of the first round were not enthusiastic about the game. However, as the players suggested several improvements to make the game more appealing, we learned a lot from their feedback and re-designed some parts of *GuessWhat?!* right after this first test session. In particular, the description extraction mechanism has been greatly improved to generate much more simple fragments which are presented to the players. Additionally, game components such as a timeout to prevent dead-locks or a chat function for communication has been added. When we had finished the implementation of the revised, second version of the game, we conducted two more test sessions, each of them with 6 participants. In order to help us evaluate the gaming experience, the players were asked to fill out the questionnaire presented in Table 1.

In total, 10 players filled out the questionnaire. The most striking findings of this survey are summarized below. While the number of answers was too small for generating meaningful statistics, the feedback was mostly positive:

- We found no correlation between the players' prior knowledge about ontologies and their understanding of the game rules.
- None of the players disliked the game concept per se.
- A few people found that the game got boring after a while, but most of them were willing to play again soon.
- The generated descriptions made sense to the users in most of the rounds.
- The majority of players found that the others judged their answers in a fair manner.

4.2 Acquired Knowledge

During the various test sessions which we conducted in order to evaluate the "fun factor" of the game, an overall number of 59 class expressions was generated and labeled by the players. Table 2 shows a subset of these descriptions along with their corresponding seed concepts as well as the labels guessed by the players. For example, given the

[13] `http://nitemaster.de/guesswhat/data.html`

1. What is your experience with ontologies?
   Well experienced / *No expert* / *No knowledge about ontologies*
2. Are the game idea and the rules comprehensible?
   Yes / *Learned by doing* / *No*
3. How many rounds did you play?
4. How many players participated in your game (including yourself)?
5. Did you enjoy playing the game?
   Yes / *Only in the beginning* / *No*
6. Would you like to play the game again?
   Yes / *No*
7. Do you think that the order of the definition fragments did
   make sense? (i.e. getting more and more specific over time)
   Yes / *Sometimes yes, sometimes no* / *Mostly not*
8. Did you find it hard to answer?
   Yes / *Sometimes* / *No*
9. Do you think the other players' evaluation was fair?
   ○ *Yes* / *Sometimes not* / *No*
10. Please point out problems that you experienced while
    playing. (e.g. technical problems)
11. Please point out what could be improved, especially
    if you did not enjoy playing the game.

Table 1. The questionnaire for evaluating the second version of *GuessWhat?!*.

description that was generated for the seed concept `photo`, one participant of the game thought of a *picture of water*, while the other players said *image*, *poster* or *map* respectively. All of these answers are plausible and thus can be used to extend the ontology. For example, given the description that was generated for the seed concept `horse`, one participant of the game thought of a *mule*, while the other players all said *horse*. Note that not every fragment of the class expression makes perfect sense from a formal point of view. Some of the errors were introduced by misleading class labels, the extraction of contradictory facts or by the false classification of words during the natural language processing. However, as the players are asked to find fitting answers, they are held to recognize such malformed expressions and react by passing and ending the round.

In some cases the concept names provided by the players seem to denote concrete individuals rather than classes (e.g. *Focus*, a German magazine). Ideally those should be recognized and handled appropriately. Several of the other concept names do not really match the original description, like *milky way*, for example, which is not a type of `plasma`). This fragment of the class expression generated for the seed concept `star` has been extracted from OpenCyc, according to which a star is a kind of plasma.[14] Finally, not every fragment of the class expressions suggested by *GuessWhat?!* makes perfect sense from a formal point of view. Several errors were apparently introduced when long class labels were split into their semantic constituents (e.g. `containing stories AND articles`). Despite the above mentioned problems, many of the generated descriptions can be represented by means of OWL in a relatively straightforward way. For example, the class expression `device AND solid`

[14] `http://sw.opencyc.org/concept/Mx4rvVi80ZwpEbGdrcN5Y29ycA`

`AND tangible AND user_guided AND (egg_shaped OR round)` which was assigned to the concept `ball` by the players could be formalized as follows:

```
<owl:Class rdf:about="ball">
 <rdfs:subClassOf rdf:resource="device"/>
 <rdfs:subClassOf rdf:resource="solid"/>
 <rdfs:subClassOf rdf:resource="tangible"/>
 <rdfs:subClassOf rdf:resource="user_guided"/>
 <rdfs:subClassOf>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
    <rdf:Description rdf:about="egg_shaped"/>
    <rdf:Description rdf:about="round"/>
   </unionOf>
  </owl:Class>
 </rdfs:subClassOf>
</owl:Class>
```

5 Conclusion

As noticed by Buitelaar and Cimiano [4], the implementation of appropriate user interaction paradigms is among the greatest challenges for today's ontology learning approaches – let it be ontology learning from text or from structured resources. This is partly because automatically approaches are still far from achieving the accuracy that humans have in any knowledge modeling task. Also, the realization of the semantic web vision is such an ambitious goal that it seems indispensable to involve more people than just a handful of professional knowledge engineers. Especially domain experts without any prior knowledge about formal semantics and ontology representation languages must be enabled to contribute to the construction of ontologies.

In this paper, we presented *GuessWhat?!*, a semantic game with a purpose which has been developed in order to facilitate the construction of ontologies by people without profound knowledge in the field of semantic technologies. By hiding the complex syntax of ontology representation languages under the surface of an entertaining multiplayer online game, it makes knowledge acquisition easier and a lot more fun. In our opinion, this way of combining the wisdom of the crowds with semantic web mining is a very promising paradigm for future knowledge acquisition. Initial user studies indicate that a game like *GuessWhat?!* can be a lot of fun and that it might even raise awareness for semantic technologies among people who have never thought about problems such as the knowledge acquisition bottleneck or the semantic web.

Still, many technical and conceptual enhancements are left for future work. For example, we plan to redesign the current scoring system in order to improve the longterm motivation of the game, and to reduce the temptation of cheating, e.g., by an unfair evaluation of the rivals' answers. This is quite important as the overall success of the game with respect to the purpose of knowledge acquisition crucially hinges on the reliability of the information that can be obtained during the game. Moreover, we would like to conduct another user study, as we hope that more data (i.e. collected from a lot more users or within a longer timeframe) will enable the investigation of new methods for

photo	`resource AND depiction AND source AND tangible AND solid AND spatially continuous AND graphic`
Players:	*picture of water*, *poster*, *map*, *image*
bed	`physical object AND intentionally made AND furniture AND object within room AND four legged flat frame AND mattress AND used for sleeping on AND NOT natural AND NOT animate AND used on everyday basis`
Players:	*bed*, *steel bed*, *nail bed*, *ferric bed with matress*, *cocaine*
cloths	`woven AND sheet of some substance AND medium amount of bio deterioration resistance AND spatial AND topic AND generic`
Players:	*nylon bedsheet*, *cloth*, *jack wolfskin jacket set*
star	`heavenly body AND any of luminous celestial object AND seen on some sky AND astronomical AND spatially bounded AND plasma`
Players:	*proxima centauri*, *milky way*, *plasma rocket disguised as angle*
kitchen	`area AND set off walls within building AND room AND food preparation AND (home OR restaurant) AND indoor location`
Players:	*kitchen*, *garden house room*
toilet	`tangible AND disposal AND apparatus AND consisting of bowl AND fitted AND hinged AND seat`
Players:	*full garbage can*, *single-use camera*, *trash can*
magazine	`periodical publications AND containing stories AND articles AND often published AND (monthly OR bimonthly) AND journal AND institution AND publisher`
Players:	*PM*, *Focus*, *Bravo*, *paper*, *comic*

Table 2. Examples of class expressions and the seed concepts they were generated from. The following rows, starting with "Players", list the labels assigned by the players.

mining semantics from the players' behavior (e.g. considering answer times). We are confident that such a bigger user study will also provide us with additional arguments for many of the conclusions we have drawn from our preliminary experiments.

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

References

1. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
2. Abraham Bernstein and Esther Kaufmann. GINO – a guided input natural language ontology editor. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *Proceedings of the 5th International Semantic Web Conference (ISWC)*, volume 4273 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 2006.
3. Dan Brickley and R.V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, World Wide Web Consortium, February 2004.
4. Paul Buitelaar and Philipp Cimiano, editors. *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, volume 167 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, Amsterdam, 2008.

5. Philipp Cimiano, Alexander Mädche, Steffen Staab, and Johanna Völker. Ontology Learning. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 245–267. Springer, 2009.
6. William W. Cohen and Haym Hirsh. Learning the classic description logic: Theoretical and experimental results. In Jon Doyle, Erik Sandewall, and Pietro Torasso, editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 121–133. Morgan Kaufmann, 1994.
7. Alexandre Delteil, Catherine Faron-Zucker, and Rose Dieng. Learning ontologies from RDF annotations. In Alexander Maedche, Steffen Staab, Claire Nedellec, and Eduard H. Hovy, editors, *Proceedings of the 2nd Workshop on Ontology Learning (OL) at IJCAI*, volume 38 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2001.
8. N. Fanizzi, L. Iannone, I. Palmisano, and G. Semeraro. Concept formation in expressive description logics. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Proceedings of the 15th European Conference on Machine Learning (ECML)*, volume 3201 of *Lecture Notes in Artificial Intelligence*. Springer, 2004.
9. Gunnar Aastrand Grimnes, Peter Edwards, and Alun D. Preece. Learning meta-descriptions of the FOAF network. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, volume 3298 of *Lecture Notes in Computer Science*, pages 152–165. Springer, November 2004.
10. Sebastian Hellmann, Jens Lehmann, and Sören Auer. Learning of OWL class descriptions on very large knowledge bases. *International Journal on Semantic Web and Information Systems*, 5(2):25–48, 2009.
11. Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. OWL 2 Web Ontology Language Primer. Recommendation, W3C, October 2009.
12. Jeff Howe. The rise of crowdsourcing. *Wired*, 14(6), 2006.
13. Alexander Maedche and Valentin Zacharias. Clustering ontology-based metadata in the semantic web. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, volume 2431 of *LNCS*, pages 348–360. Springer, 2002.
14. Thomas Markotschi and Johanna Völker. GuessWhat?! – Human Intelligence for Mining Linked Data. Technical report, KR & KM Research Group, University of Mannheim, B6 26, 68159 Mannheim, Germany, 2010. available at http://ki.informatik.uni-mannheim.de.
15. Marta Sabou, Mathieu d'Aquin, and Enrico Motta. SCARLET: SemantiC RelAtion DiscoveRy by Harvesting OnLinE OnTologies. In Sean Bechhofer, Manfred Hauswirth, Jörg Hoffmann, and Manolis Koubarakis, editors, *Proceedings of the 5th European Semantic Web Conference (ESWC)*, volume 5021 of *LNCS*, pages 854–858. Springer, 2008.
16. Katharina Siorpaes and Martin Hepp. Ontogame: Towards overcoming the incentive bottleneck in ontology building. In *Proceedings of the 3rd International Federation for Information Processing Workshop On Semantic Web & Web Semantics (SWWS '07)*, volume 4806 of *Lecture Notes in Computer Science*, pages 1222–1232. Springer, 2007.
17. Katharina Siorpaes and Martin Hepp. Games with a purpose for the semantic web. *IEEE Intelligent Systems*, 23(3):50–60, 2008.
18. Gerd Stumme, Andreas Hotho, and Bettina Berendt. Semantic web mining: State of the art and future directions. *Journal of Web Semantics*, 4(2):124–143, 2006.
19. Johanna Völker, Pascal Hitzler, and Philipp Cimiano. Acquisition of OWL DL axioms from lexical resources. In *Proceedings of the 4th European Semantic Web Conference (ESWC'07)*, volume 4519 of *Lecture Notes in Computer Science*, pages 670–685. Springer, Juni 2007.
20. Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the Special Interest Group on ComputerHuman Interaction conference on Human factors in computing systems*, pages 319–326. Association for computer machinery, 2004.

Linking Semantic Personal Notes*

Laura Drăgan, Alexandre Passant, Tudor Groza, and Siegfried Handschuh

DERI, National University of Ireland, Galway,
IDA Business Park, Lower Dangan, Galway, Ireland
firstname.lastname@deri.org
http://www.deri.ie/

Abstract. Semantic Web technologies are available and gain popularity both on the Web and on the desktop. However, in spite of common representation formats, personal and online data is still difficult to interlink, notably because of the different vocabularies used to describe it, as well as the lack of common identifiers between desktop and Web-based applications. In this paper, we describe a process for easily publishing and sharing of personal notes as Linked Data. Our approach can be used to publish any kind of information from the desktop to the Web, enabling integration of small chunks of personal knowledge into the Web of Data and focusing on a user-driven approach of knowledge management.

1 Introduction

Semantic Web technologies are now deployed in various domains and applications. Among the different sub-domains of the broader Semantic Web vision, two relevant fields are the Linked Data initiative, focusing on global interlinking on the Web, and the Semantic Desktop, focusing on personal information integration. While these two domains share compatible representation models (RDF(S)/OWL), there is still a gap between data from the Web and the desktop. Among others, vocabularies that they use are generally not well integrated and identifiers (URIs) are generally distinct. Such gap can be explained as the Semantic Desktop focused on using local identifiers and desktop-related ontologies, while the Linking Open Data (LOD) initiative focused on the global reuse of identifiers and ontologies.

In this paper, we tackle a particular issue regarding the integration of data from these two environments, offering an approach for publishing personal notes from the desktop (using Semantic Desktop technologies) to the Web (using the Linked Data principles). Especially, our need is to publish this data online without losing the personal context established on the desktop. Our approach consists of two main steps: (i) preparing the desktop data for sharing, and (ii) publishing it online. In addition, it requires two prerequisite steps, which are not the focus of this paper: (i) the note-taking process and annotation of the note (adding

* This work is supported by the Líon-2 project funded by Science Foundation Ireland under Grant No. SFI/08/CE/I1380.

the context), and (ii) the identification of Web URIs which represent the same real-world thing as the desktop resources that belong to the context of a note. We will however describe (in less detail) these two initial steps to give the entire view of the workflow.

Transferring personal desktop data online requires some issues to be properly addressed. To achieve this goal, our contributions include: (i) mappings between the relatively small number of desktop vocabularies and the most popular Web vocabularies. The mappings are used in the transformation of the desktop data, represented with the desktop ontologies, to data represented with the Web vocabularies, ready to be published online; (ii) a process for publishing of desktop information on the Web using the Linked Data principles, while protecting the sensitive private data from being shared unwillingly, and (iii) a system implementation that allows sharing of semantic personal notes as semantic blog posts, interlinked with existing information within the LOD cloud.

The remainder of the paper is structured as follows. We first describe a motivating use case, from which we identified the main requirements of our system (Section 2). In Section 3, we continue with the background work on which our approach is built. Section 4 details the process and its realisation, focusing on the ontology mappings and the software architecture, and Section 5 evaluates the conformance of the system with the initial requirements. We then discuss related work and some challenges and lesson learnt we have found when implementing the system, before concluding the paper.

2 From Note-Taking to Weblogging: Use Case and Requirements

Two relevant characteristics of blog posts are: (i) their topics are of interest to the author and thus are very likely to have references to things present on the desktop (e.g. people, events); (ii) they belong to a context consisting of the references made in their content, such as places, projects, or other blog posts However, not all blog posts start by being a blog post. Some are just ideas or impressions jotted down for later, in one's preferred desktop note-taking application. Nevertheless, some of these notes do become posts after polishing and refining.

Tools from the Semantic Desktop [1] provide means to enhance these notes locally, by interlinking them with other desktop data — the contacts in the address book, the events from the calendar application, the projects worked on, the music listened to. Semantic note-taking tools like SemNotes[1] automatically generate relations between the notes and the desktop things mentioned in their content. For example, it allows to link one note about an upcoming concert to the performing artist which is in turn linked to the music files of that artist and pictures from earlier shows stored in a desktop photo application. Such annotations give context to the note and should be preserved when the note is published as a blog post on the Web, since it enables serendipitous browsing and information discovery, through the relevant additional links they contain.

[1] `http://smile.deri.ie/projects/semn`

Currently, personal notes, even the ones semantically enriched using Semantic Desktop applications, must be published as blog posts by being manually copied into a blogging tool. In this way, any additional semantic information available on the desktop becomes lost or, if copied, leads to broken references as they point to the local resources which are not accessible outside of the desktop. The note-taking to publishing process is sometimes shortcut by using the drafting functionality that some systems like WordPress or Blogger offer, so that users can directly take the notes in the blogging tool, usually online, thus replacing the desktop note-taking application. Using online tools deprives the user from having the personal context automatically added to the blog post, since desktop information cannot be easily integrated in Web-based interfaces.

In order to enable a better translation from personal notes to blog posts, or simply to Web-based information available to others (for example, meeting notes published in a company intranet or lecture notes shared between students of a same class), we defined a list of requirements that a system for publishing semantic personal data online should fulfil:

R1 Publish the complete desktop data on the Web without losing any relevant information, including metadata and context (e.g. tags, relations, identifiers);
R2 Protect any machine readable and private data that might be unwillingly be included in the context being transferred;
R3 Publish the note according to the Linked Data principles and describe it use popular ontologies;
R4 Enable object-centred sociality by establishing connections between data published by different users.

3 Overview of the Approach

3.1 Background

In order to enable our approach for publishing notes from the desktop to the Web, we reused previous work and software components already available. In this section, we present them briefly and explain why we chose them and how they contribute to the global picture that our architecture provides.

Semantic Desktop. Extensive research has been done in the area of the Semantic Desktop. Systems like Haystack [2], IRIS [3] or NEPOMUK [1] bring Semantic Web technologies to the desktop. The vision of the Semantic Desktop is to create a space of interconnected resources, where applications encourage linking between new and existing resources and provide new and easy ways of browsing, searching and organising the data.

Our solution builds on the NEPOMUK realisation of the Semantic Desktop, more precisely Nepomuk-KDE[2]. It extracts metadata from the desktop (*i.e.* from files, address book, calendar, task manager, etc.) and integrates it into a central repository, making it available to all applications. The data is described using

[2] `http://nepomuk.kde.org`

a common representation – Nepomuk Representation Language (NRL)[3], and a set of ontologies[4], known as "desktop ontologies". They describe the desktop data, at different levels of abstraction, and can be complemented by additional ontologies, like Xesam[5].

SemNotes. SemNotes is a note-taking application for the NEPOMUK Semantic Desktop, which uses semantics to save the context of each note by linking it to the relevant desktop resources mentioned, such as people, events, projects, etc. It uses the "desktop ontologies" to describe its data structure and the relations between the notes and other resources from the desktop. We decided to add our Linked Data publishing functionalities to an existing note-taking application as SemNotes for two reasons: (i) usually blog posts or online articles start as personal notes that are refined until ready to be published, as we discussed earlier, and (ii) a familiar application such as SemNotes is more likely to be used than a new one, notably as users will not have to learn a new systems but keep to their existing note-taking habits.

Linked Data. The term Linked Data was first introduced by Berners-Lee in 2006 to define a set of best practices for publishing data on the Web [4]. In addition to these principles, the recent Linking Open Data[6] initiatives enables the creation of a huge amount of interlinked RDF data on the Web, from various datasets, ranging from HCLS information to the BBC programmes. Our system takes advantage of this increasing amount of structured data, about various kinds of entities available online [5], for defining and using identifiers so that desktop information and Web information can be related.

3.2 Overall approach

We propose an approach that enables the publishing and sharing of personal notes by extending the functionality provided by SemNotes. The process consists of two steps: (i) transformation and (ii) publication. In the first step, the note is transformed locally for publication, and private local data is replaced with public server references. In the second step, the transformed note is published online on a dedicated server, where the resources referenced and the tags assigned, are shared between the notes of all users. As we mentioned above, there are also two prerequisite steps: (i) the note-taking process and semi-automatic annotation of the note, which is the usual note-taking approach, and (ii) the identification of Web aliases for the desktop resources related to a note, where URIs are mined from the Web for locally defined resources, such as people, events or projects. These steps are required in the workflow, but will not be detailed in this paper.

The first prerequisite step — note-taking and annotation of the note with the relevant desktop resources — must be performed before any actual sharing of

[3] NRL is an extension of RDF which provides named graphs and a closed world assumption more suitable to the desktop environment.

[4] `http://www.semanticdesktop.org/ontologies/`

[5] `http://xesam.org/main/XesamOntology`

[6] `http://linkeddata.org`

notes can be done. The annotation is done semi-automatically and is an existing feature in SemNotes. For each note, the user is offered a list of possible related desktop resources from which he can choose the relevant ones. When a resource is chosen, a link (*i.e.* an RDF triple) is created in the local repository between it and the note.

The second prerequisite step consists in finding Web resource for each of the desktop entity linked to the note that is about to be published. This step is currently executed by a desktop service that relies several Semantic Web indices (*i.e.* Sindice[7]) and public SPARQL endpoints (*i.e.* DBpedia, Semantic Web Dog Food Server) to retrieve results. The matching process is based on the one described in [6], which we developed further, to include more types of desktop resources. It is based on a combination of methods: type and property mapping and filtering and a combination of string matching algorithms. The service has access to, and uses all the information available on the desktop about a resource to identify only exact matches for it.

4 System Implementation Details

Based on the process described in the previous section, we engineered a system for publishing personal notes on the Web. The system is divided between its local part and its remote part, as shown Figure 1. The local part handles local *private* data, while the remote one handles online *public* data. The separation between them extends over 3 layers: ontology, data and application. On the *ontology* level, the NEPOMUK desktop ontologies are used locally while popular Web vocabularies are used on the server-side. These ontologies are used to describe the *data* exchanged between the applications. Desktop data is stored in the local NEPOMUK repository, which is provided with any NEPOMUK installation, while Web data is distributed in the Linked Data cloud. Finally, on the *application* level, the local component is an extension to SemNotes that provides publishing functionality for notes, and the remote component is a server that hosts and publishes online the notes received.

The first step of the process is executed on the local side, by an extension of the SemNotes application. Then, the publication step is done by the server, which receives information from the desktop and publishes the note, as we will describe next. These two application components, the communication between them, and the data translation process are described in detail below.

4.1 Ontologies

Although both the Semantic Desktop and the Semantic Web use the same representation languages, *i.e.* RDF(S)/OWL, they use different vocabularies to describe their data. This vocabulary gap makes data integration difficult. The NEPOMUK project uses "desktop ontologies" to describe its data. The central ontology here is the Personal Information Model[8] (PIMO). SemNotes represents

[7] `http://sindice.com/`

[8] `http://www.semanticdesktop.org/ontologies/pimo/`

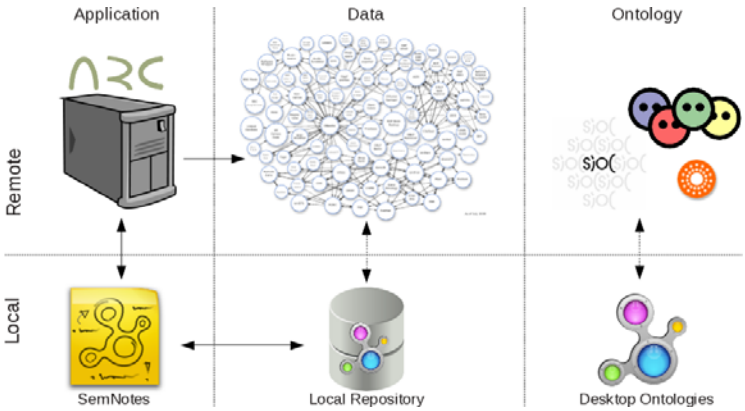

Fig. 1. Overview of the system.

personal notes as instances of `pimo:Note` and are linked to the `pimo:Things` they mention by the relation `pimo:isRelated`. When a desktop resource is found to represent the same real world entity as a Web resource, the relation is stored on the desktop as `pimo:hasOtherRepresentation`. This property is recommended by the PIMO specification as desktop equivalent to the `owl:sameAs` relation, although without the formal semantics that the latter provides. We also use the property `pimo:hasOtherRepresentation` to store the remote URL of a note when it is published. The property is replaced with `pimo:hasDeprecatedRepresentation` if the note changes on the desktop after publication.

While well-suited to represent desktop information, these ontologies are not used, so far, on the Web. However, numerous vocabularies have emerged for describing semantic data published online. Among them, a limited number have gained wide-spread adoption, including: (i) FOAF for describing people and their social relations; (ii) SIOC for describing communities and their interactions; (iii) DOAP[9] for software projects; (iv) GeoNames[10] for geographic information; (v) the Music Ontology for music-related information; and (vi) models such as Dublin Core for general metadata or SKOS to represent lightweight controlled vocabularies. Such ontologies have now been widely adopted and are recommended as best practices when publishing data on the Web [7].

Consequently, while representing similar objects, the two sets of vocabularies must be aligned so that on the one hand, desktop information can be moved to the Web and understood by usual SW applications (that rely on the aforementioned vocabularies) and on the other hand, Web information could be understood and imported by SD applications. In order to enable interoperability between the desktop and the Web, we defined mappings between the sets of

[9] `http://trac.usefulinc.com/doap`
[10] `http://www.geonames.org/ontology/`

Class	Subclass of
pimo:Note	sioc:Post
nao:Tag	sioct:Tag
pimo:Person	foaf:Person
pimo:Project	doap:Project
pimo:Event	ical:Vevent

Property	Subproperty of
nao:prefLabel	rdfs:label
nao:created	dcterms:created
nao:lastModified	dcterms:modified
nao:hasTag	sioc:topic
pimo:isRelated	sioc:related_to

Table 1. Sample of the mapping between (i) classes, and (ii) properties.

ontologies. The mappings create appropriate subclasses or subproperties of the relevant concepts from the chosen vocabularies.

SIOC is probably the most widely used vocabulary for interlinking social media within the Linked Data cloud. There are already many tools for creating and using SIOC data [8]. This is why we chose to represent the `pimo:Note`s as `sioc:Post`s when they are published online with our system. The rest of the desktop resources are also transformed into concepts from the vocabularies listed above (see Table 1 (i)), the mappings being published at `http://rdfs.org/sioc/nepomuk`. The note's properties, like title, creation and last modification time, are translated to the appropriate Dublin Core properties: `dcterm:created`, `dcterms:modified` and `dcterms:title`. The tags associated locally to the notes are transformed into `sioct:Tag`s associated with the post using the `sioc:topic` property. Table 1(ii) lists the proposed mappings for properties[11].

4.2 Server Schema

In order to publish the resources with a consistent URI scheme, we defined patterns for naming of the various objects published from the desktop on the Web. In the schema definition, we apply several Linked Data patterns described in [9]: (i) *patterned URIs* for all the entities, to make them more human readable; (ii) *proxy URIs*, (iii) *annotation* and (iv) *equivalence links* for the resources related to the notes, to unify various sources; (v) *natural keys* in the tag URIs.

For each note the server generates a new unique identifier `id` which is used to create the note's URI in the form: `http://semnotes.deri.ie/notes/note/id`.

According to the *proxy URIs* identifier creation pattern, we generate new URIs for the resources related to the notes. This ensures that the publishing process is consistent and avoids having to choose among several Web aliases a resource could have. Like the notes, each resource has a unique identifier on the server, which is used to create the resource URI according to the following format: `http://semnotes.deri.ie/notes/resource/id`. Each resource is shared by all the notes that link to it, which increases the interlinking and the consistency of the data. For each resource, the server keeps internally a list of Web aliases (*i.e.* Web URIs that were found to represent the exact same real world thing) using `owl:sameAs` links.

[11] Although `nao:lastModified` and `dcterms:modified` do not have the same semantics, defining subproperty relations between them is acceptable.

Tags are considered a particular type of resources, and are also shared on the server. The specific format for the URI differentiates them from regular resources: `http://semnotes.deri.ie/notes/tag/label`. The label of the tag acts as a unique identifier, and is case sensitive. They are created on the fly, and are persisted when they are used for the first time.

Non-information resources[12] also got their own URI, and we distinguish URI of the resources and URIs of the pages describing them.

4.3 Transformation of the Note for Sharing

The first step of the process consists in the preparation of the note for publishing. This phase consists of including all the relevant information about the note in the content, specifically the title, creation and last modification time, the tags and the referenced resources. This transformation is necessary, so that less only the HTML content of the note is sent to the server, and not the entire RDF graph describing the note. The content is already stored as HTML, but to include in it all the metadata about the note, it has to be enriched with RDFa before it is posted to the server.

The preparation step is done on the desktop side, by the extension to the note-taking tool, but still requires to communicate with the publishing server to retrieve Web URIs for the note and the linked resources. In case the note has already been published, the user can overwrite the old post (on the Web) or create a new one. Depending on this choice, the server is requested a new URI or the existing one is used (that was saved in the local repository when the note was published the previous time). The referenced resources are shared by all the published notes, therefore the server must create the URI for a resource only if it has not been created before. To decide if a local resource already has a server URI created, the list of Web aliases found for it — in the second prerequisite step of the process — is sent to the server (see Fig. 2). If a resource with the same type and a similar list of aliases exists, the server reuses it, otherwise it creates a new one and saves the information about it in its own RDF repository. On the server, the URI aliases are saved as `owl:sameAs` as it is customary for Linked Data. The server URIs for the note and the resources are also stored on the desktop for reuse, as `pimo:hasOtherRepresentation`.

The communication between SemNotes and the server is done with a single REST call, in order to minimise network delays. The reply contains the newly created URI for the note, if one was required, as well as a list of server URIs for the resources (see Figure 3).

Using the information received from the server, the note content is enriched with RDFa. The metadata about the note, like type, creation and last modification times and the tags, is added in `meta` tags in the `head` of the HTML page. RDFa is added to the `title` tag and in the `body`, to the links. Figure 4 shows the content of a note prepared for publishing.

[12] For a discussion about *information resources* and *non-information* resources, we refer the reader to `http://www.w3.org/TR/webarch#id-resources`

```
{
    "id" : "",
    "resources": [
    {
        "id": "nepomuk:/res/bfcdcd1a-4898-492f-940b-4cc4c67799a7",
        "type": "mo:MusicArtist",
        "uris": [
            "http://dbpedia.org/resource/Scorpions_(band)",
            "http://musicbrainz.org/artist/c3cceeed-3332-4cf0-8c4c-bbde425147b6"
        ]
    }
    ]
}
```

Fig. 2. JSON formatted message sent to the server.

```
<note uri="http://semnotes.deri.ie/notes/note/4baccab834e20">
    <resource local="nepomuk:/res/bfcdcd1a-4898-492f-940b-4cc4c67799a7"
                uri="http://semnotes.deri.ie/notes/resource/4bacca84ca8bb"/>
</note>
```

Fig. 3. Server reply with the server URIs for the resource aliases sent.

4.4 Publication Step

After preparation step, which takes place on the desktop side, the RDFa enriched content is sent to the server via another REST call. The publication step of the process only handles public data. When the content is received it is parsed and the server extracts the contained RDF triples and stores them in its repository. The content (as it is received) is also stored.

The server implementation uses ARC2[13], as it provides out of the box RDFa parsing and an RDF repository. It is easily deployable due its minimal setup requirements (a PHP enabled Web server and a MySQL database), thus making our system easily deployable as well.

[13] `http://arc.semsol.org`

```
<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML RDFa 1.0//EN'
                        'http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd'>
<html about="http://semnotes.deri.ie/notes/note/4baccab834e20">
    <head>
        <meta content="sioc:Post" property="rdf:type"/>
        <meta rel="sioc:topic" href="http://semnotes.deri.ie/notes/tag/concert"/>
        <title property="dc:title">concert sunday</title>
    </head>
    <body> ...
        <a rel="sioc:is_related"
             href="http://semnotes.deri.ie/notes/resource/4bacca84ca8bb">Scorpions</a> ...
    </body>
</html>
```

Fig. 4. RDFa-annotated XHTML content of note.

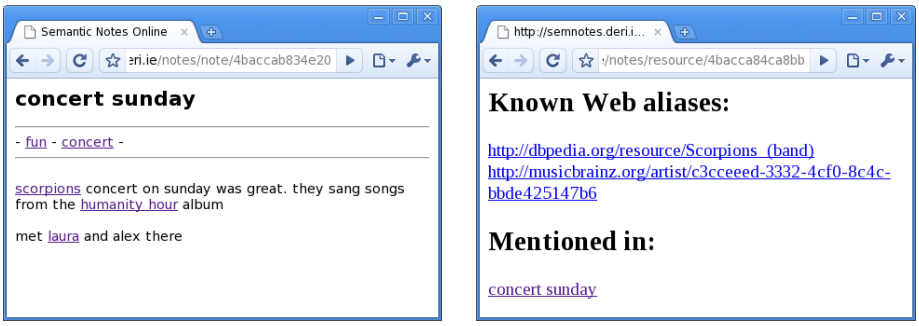

Fig. 5. Online view of a note (i) and a resource (ii).

All server URIs are dereferenceable, as required by the Linked Data principles. For notes, the URI redirects to the RDFa annotated HTML page containing the note itself (as shown in Figure 5 (i)), the URI of the note being the URL of this page. For the linked resources, the URI is also dereferenceable and provides RDFa information about itself, linking to the known existing Web aliases of the same resource. The description also includes a list of backlinks to all the notes that reference the resource (see Figure 5 (ii)). The page for a tag will contain backlinks to all the notes tagged with it.

The RDFa annotated page for the note is generated on the user's desktop by the SemNotes plugin, as we have seen in the previous step, while the one describing each resource and tag is generated on the fly, by the server, when the URI is requested.

5 Conformance with the Initial Requirements

When establishing the specifications of the framework, we identified four main requirements (Section 2). Our proposal conforms with them as follows.

***R1**: Publish the complete desktop data on the Web without losing any relevant information, including metadata and context (e.g. tags, relations, identifiers).*
By translating existing desktop data in RDF an putting it online, available as RDFa, the whole information available on the desktop side is made available on the Web for further reuse. In addition, all information from the original note-taking tool, including title, tags, etc. is publicly made available on the Web.

***R2**: Protect any machine readable and private data that might be unwillingly be included in the context being transferred;.*
By replacing the private desktop data with equivalent public Web data, we protect the former. On the desktop there is much private personal information stored about the resources, like the email address or telephone number for people, or the list of attendees of an event. When the person or event linked to by a note that is afterwards published online, such private information is not exported, because the reference to the local resource is replaced by a reference to already

public Web data representing the same thing. In this manner, the context of the note being published is preserved, but the private details are not exposed.

***R3**: Publish the note according to the Linked Data principles and describe it use popular ontologies.*
Our system publishes notes on the Web using the Linked Data principles. Each note has its own URI, as well as resources, and these URIs are made dereferenceable, while distinguishing information resources and non-information resources. In addition, while original desktop data is provided using "desktop ontologies", the published information is made available using FOAF, SIOC, Dublin Core, etc. and the mappings have been validated through Vapour[14].

***R4**: Enable object-centred sociality by establishing connections between data published by different users.*
Since resources and tags are shared between users, notes can be browsed serendipitously through shared topics, or tags. This enables "object-centred sociality" [10], since people can interact around these shared tags and topics, such as projects or people that they know in common.

6 Related Work

Semantic blogging has received much interest since it was introduced by Cayzer and Shabajee in [11], and later when Karger and Quan described semantic blogging in the context of the Semantic Web with the Haystack browser [12]. So far, existing systems for semantic blogging fall into two categories: (i) desktop applications that involve publishing the actual local resource information together with the blog post, or (ii) online application that does not have access to desktop data relevant to the user.

The main benefit of the first category, represented by tools like SemiBlog [13] or SemBlog [14], is the fact that the user has better access to the relevant data from the desktop. However, both tools require that the resources that contain sensitive private information are published together with the blog post, which might lead to privacy issues. The SemBlog project allows users to add data from personal ontologies to their blogs. SemiBlog, allows integration of personal data in the posts by drag in drop from various desktop applications like the address book. They are used for exchange of personal information in the blog posts, which differs from our approach of using already published web data as to protect the privacy of the personal information. The process described implies manually adding the metadata, while our approach relies on automatic export. Both tools comply with our first requirement, but not with the last three.

Online services like BlogAccord [15] for music information or Zemanta[15] blogging assistant, belong to the second category. They have access to various online resources to create the context of a blog post and enhance the blogging experience, but not to the personal context of the user.

[14] `http://vapour.sourceforge.net/`
[15] `http://www.zemanta.com`

7 Conclusion

In this paper we presented an approach for publishing personal notes as Linked Data on the Web. The aim of our work was to provide a way for publishing and sharing complete information by preserving the personal context of the notes without compromising privacy. Our solution makes a step towards bridging the gap between Semantic Desktop data and Linked Data.

We defined a publishing process that comprises two steps: (i) preparation – the note is transformed into a SIOC-based Web representation; and (ii) publication / sharing – the note is published online following the Linked Data principles. In addition, we provided a related implementation and tested it against a set of requirements regarding publishing personal content from the desktop to the Web as Linked Data. While we do not address security issues in this current release, we consider SW-compliant authentication systems such as FOAF+SSL [16] for the upcoming version of our application.

References

1. Bernardi, A., Decker, S., van Elst, L., Grimnes, G., Groza, T., Jazayeri, S.H.M., Mesnage, C., Moeller, K., Reif, G., Sintek, M. In: The Social Semantic Desktop: A New Paradigm Towards Deploying the Semantic Web on the Desktop. (2008)
2. Quan, D., Huynh, D., Karger, D.R.: Haystack: A Platform for Authoring End User Semantic Web Applications. In: Proc. of the 2nd ISWC. (2003) 738–753
3. Cheyer, A., Park, J., Giuli, R.: IRIS: Integrate. Relate. Infer. Share. In: Proc. of the Semantic Desktop and Social Semantic Collaboration Workshop. (2006)
4. Berners-Lee, T.: Linked Data. Technical report, W3C (July 2006)
5. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. Int. J. Semantic Web Inf. Syst. **5**(3) (2009) 1–22
6. Groza, T., Dragan, L., Handschuh, S., Decker, S.: Bridging the gap between linked data and the semantic desktop. In: Proc. of the 8th ISWC). (2009)
7. Bizer, C., Cyganiak, R., Heath, T.: How to Publish Linked Data on the Web (2007)
8. Bojars, U., Passant, A., Cyganiak, R., Breslin, J.: Weaving SIOC into the Web of Linked Data. In: Proc. of LDOW2008. (April 2008)
9. Dodds, L., Davis, I.: Linked Data Patterns. (2010) `http://patterns.dataincubator.org`.
10. Knorr-Cetina, K.: Sociality with Objects: Social Relations in Postsocial Knowledge Societies. Theory, Culture & Society **14**(4) (1997) 1–30
11. Cayzer, S., Shabajee, P.: Semantic blogging and bibliography management. In: BlogTalk Proc. (2003) 101–108
12. Karger, D.R., Quan, D.: What Would It Mean to Blog on the Semantic Web? In: Proc. of the 3rd ISWC, Hiroshima, Japan, Springer (2004) 214–228
13. Möller, K., Decker, S.: Harvesting Desktop Data for Semantic Blogging. In: Proc. of Semantic Desktop Workshop, ISWC, Galway, Ireland. (2005)
14. Takeda, H., Ohmukai, I.: Semblog Project. In: Activities on Semantic Web Technologies in Japan, A WWW2005 Workshop. (2005)
15. Cayzer, S.: What next for Semantic Blogging? In: Proc. of the SEMANTICS 2006 conference, Vienna, Austria (November 2006) 71–81
16. Story, H., Harbulot, B., Jacobi, I., Jones, M.: FOAF+SSL: RESTful Authentication for the Social Web. In: Proc of the First SPOT workshop, ESWC 2010. (2009)

LOTED: Exploiting Linked Data in Analyzing European Procurement Notices

Francesco Valle[1], Mathieu d'Aquin[2], Tommaso Di Noia[1] and Enrico Motta[2]

1. Technical University of Bari, Electrical and Electronics Engineering Department
Information Systems Research Group
francescovalle84@gmail.com, t.dinoia@poliba.it
2. Knowledge Media Institute, The Open University, Milton Keynes, UK
{m.daquin, e.motta}@open.ac.uk

Abstract. The world of procurements and eProcurement generates daily large amounts of data, that represent knowledge of great economical value both for individual companies and for public organisations wishing to achieve a better understanding of a given market. However, such data remains difficult to explore and analyze as it is being kept isolated from other sources of knowledge, in dedicated systems. In this paper, we present an ongoing work on extracting and linking data from the European 'Tenders Electronic Daily' system, which publishes approximately 1,500 tenders five times a week. We specifically show how such information is dynamically extracted and linked to external datasets, and how the created links enrich the original data, introducing new perspectives to its analysis. We show tools we developed to support such 'linked data-based' analysis of data, and report on the lessons learnt from our experience in building a linked data application with potential for real-life use in knowledge extraction.

1 Introduction

The Tenders Electronic Daily (TED[1]) website is a portal maintained by the European Commission and dedicated to the public procurement in the countries of the European Union. As the name suggests, it is updated daily (5 days a week) with newly published tenders in 14 different sectors (e.g., Education, Technology and Equipment, Agriculture and Food). Each tender is an official document, containing information related to the public organisation it originates from (e.g., a town council, a public administration), its location, the type of activity it is related to, etc.

As such, this portal represents a rich source of information from which crucial strategic and economical knowledge could be extracted. Services exist that provide mail alerts and other mechanisms for companies on the basis of TED, but these tend to simply redirect the information from the original system to the user, without making any further analysis.

In this paper, we demonstrate how the principles of linked data can be used on the information exposed by the TED system, on the one hand to improve access

[1] `http://ted.europa.eu/`

to this data, allowing new applications to be built on top of it, and on the other hand, to investigate how links to external datasets can bring valuable additional perspectives into the data, enriching it with new dimensions and making possible new forms of analysis that exploit data and links to reach a better understanding of the global public market. In addition, we derive from our experience in building such a concrete linked data application lessons regarding the current limitations of linked data and of the supporting tools.

In the next section, we detail how we developed a platform realising a workflow from obtaining the data in the original TED portal, to exposing it as linked data and providing interfaces to it. In Section 3, we detail a prototype application of this platform to build visualisations combining dimensions from the original data and from external datasets, demonstrating how such an approach can provide endless possibilities for new perspectives on previously isolated data. Section 4 reports on our concrete experience in building such an application, showing in particular how additional work should be realised in the area of linked data to make applications such as ours easier to build and more efficient. Finally, we conclude the paper pointing out directions for future work in Section 5.

2 LOTED: Anatomy of a Linked Data Application

The TED portal updates its subscriber on newly published tenders daily through a set of RSS feeds, with an RSS feed for each combination of country (27 countries in total) and sector (14 sectors in total). Each RSS feed is updated at most once a day, generally five days a week, and is every time entirely replaced by the new tenders. For example, `feed://ted.europa.eu/TED/rss/en/RSS_tran_UK.xml` is the current URL of the RSS feed to the sector "Transport and related services" in United Kingdom.

A tender on TED is presented by a document, available in its original languages and in translated form. For example, `http://ted.europa.eu/udl?uri=TED:NOTICE:202572-2010:TEXT:EN:HTML&tabId=1` is a tender document (Contract Notice), for photocopying and offset printing equipment in Stuttgart, Germany. Such a document contains general, common information about the tender such as its type, requested products or services, the location, originating organisation, award criteria, etc. A summary of this data is available for each document, presented in a tabular format (see Figure 1) with normalised fields and values for these fields (for the previous tender, see `http://ted.europa.eu/udl?uri=TED:NOTICE:202572-2010:DATA:EN:HTML`). The availability of such a semi-structured summary of the document greatly facilitates the task of extracting data from the TED system, as shown in the following sections.

2.1 Overview

In this section, we give an overview of the platform for the publication and use of a linked data version of the information provided by the TED system, which we called LOTED[2] (Linked-Open Tenders Electronic Daily).

[2] `http://loted.eu`

TI	Title	D-Stuttgart: photocopying and offset printing equipment
ND	Document number	202572-2010
PD	Publication date	10/07/2010
OJ	OJ S	132
TW	Place	STUTTGART
AU	Authority name	Ministerium für Umwelt, Naturschutz und Verkehr Baden-Württemberg
OL	Original language	DE
HD	Heading	Member states - Supply contract - Contract notice - Open procedure
CY	Country	DE
AA	Type of authority	1 - Ministry or any other national or federal authority
DS	Document sent	07/07/2010
DD	Deadline for the request of documents	23/08/2010
DT	Deadline	23/08/2010
NC	Contract	2 - Supply contract
PR	Procedure	1 - Open procedure
TD	Document	3 - Contract notice
RP	Regulation	5 - European Communities, with participation by GPA countries
TY	Type of bid	1 - Global tender
AC	Award criteria	2 - The most economic tender
PC	CPV code	30120000 - Photocopying and offset printing equipment
OC	Original CPV code	30120000 - Photocopying and offset printing equipment
RC	NUTS code	DE11

Fig. 1. Example of tabular summary of a tender on the TED portal.

LOTED essentially relies on a triple store[3] which is being updated daily with information extracted as linked data from the RSS feeds of the TED system, and exposed through a SPARQL endpoint (see Figure 2). The way RDF data is extracted from the original tender documents, and how such data is linked to external datasets (currently geonames[4] and DBpedia[5]), is explained in the next section.

As can be seen from Figure 2, at the heart of the system is the *LOTED Ontology*[6], which has been specifically developed for the needs of the platform. It is a lightweight ontology, that matches directly the semi-structured representation of the tenders from the TED system, while introducing an additional level of structure. It is worth also noticing that the labels in this ontology are available in three different languages. It can be argued that reusing existing ontologies would have better encouraged interoperability. However, we found that extracting existing information was made easier and less error-prone if realised in a target structure that matched the original data closely. Mapping and integrating this ontology with others, as well as evolving it towards a more expressive modelisation of the domain of procurement is planned as part of our future work.

[3] After a few tests, we found that the Jena system (`http://openjena.org/`) with a TDB persistent store (`http://openjena.org/TDB`) offered the best compromise between flexibility, robustness and performance for our scenario.

[4] `http://www.geonames.org/`

[5] `http://dbpedia.org`

[6] `http://loted.eu/ontology`

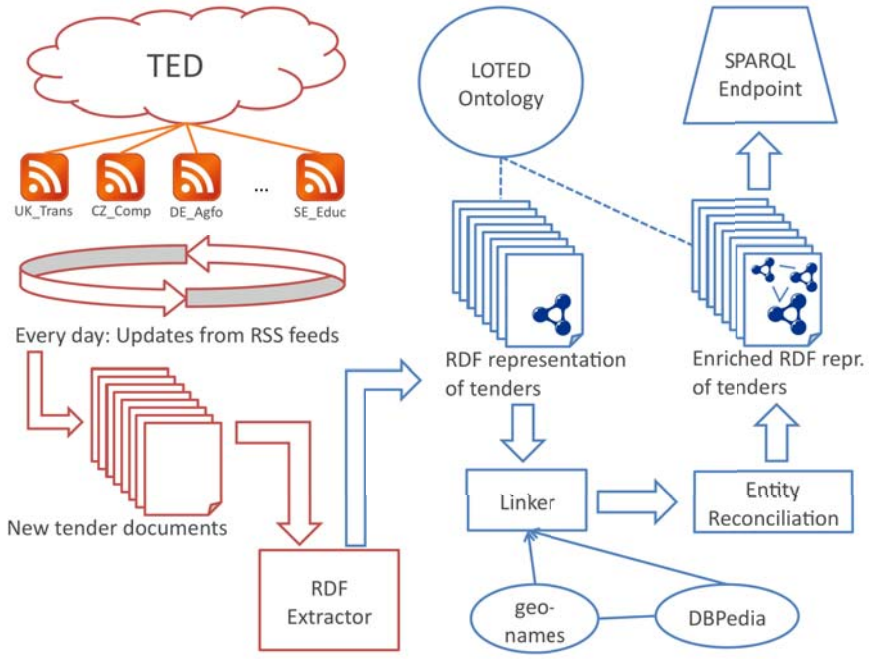

Fig. 2. Workflow for the daily update of linked data from the TED system to the LOTED endpoint. RDF representations based on the LOTED ontology are extracted from the TED RSS feeds, and enriched through automatically discovered links to geonames and DBpedia.

2.2 Extracting Information and Creating Links

The extraction of structured information is realised as a scheduled task, happening every day. It starts by checking whether any of the RSS feeds from the TED system had changed and downloading the English, tabular version of any new tender published on the portal. As explained above, extraction from these documents is facilitated on the one hand by the fact that they are formatted in a semi-structured way, and on the other hand, as the LOTED ontology has been explicitly designed to match this structure. We therefore developed a custom made RDF extractor which parses the table structure of the original document and transforms it into a structured RDF representation. Amongst the difficulties that are common to such processes is the issue of having to deal with special characters and unicode strings that have to be included in URIs of entities. We dealt with this problem by replacing any of such characters by its equivalent HTML entity.

Creating links is obviously a more challenging issue. Geographical information is well covered by available linked datasets and can provide useful informa-

tion to associate to the tender documents, including the data about the places where the originating organisation is. Within the data, the information available is a string representing the name of the city where the organisation is located, and a two letter country code. Geonames is a data set of geographical places around the world. It provides simple identifiers for places, and information about their locations (coordinates) and their names in various languages. Coordinates can be very useful, as we will see later, simply to be able to place the city on a map. Another large source of information about cities is DBpedia. Indeed, DBpedia being extracted from Wikipedia[7], it can contain a large variety of interesting data about a given city, from its population to the region it is in or the current mayor.

To link the places found in the tenders to geonames, we make use of the search engine provided on the geonames website[8]. Having both the name of the city and the country code (being in Europe) makes the query sufficiently unambiguous, so that the first result from the search engine is in the large majority of the cases the one we are looking for. Actually, we never found any error in this linking process in the time the system has been running (more than 2 months). Another advantage of relying on geonames is that it is already well connected to other datasets. In particular, DBpedia includes in most of the resources it contains about cities a *sameAs* link to the corresponding URI identifying the place in geonames. Therefore, finding links to DBpedia is realised straightforwardly through a SPARQL query requesting resources that are the same as the discovered geonames objects.

When building a linked data based platform relying on links to external datasets, different choices can be made on the way to integrate and use these links. Indeed, a natural choice would be that any query sent to the system would automatically look up for any link involved and retrieve dynamically, at run-time, the corresponding information to be included in the results. However, while this appears to be the kind of process linked data applications would normally rely on, the tool support for realising it appears to be very weak. Only a few existing systems are currently able to realise live look-ups of external entities [1, 2], under specific conditions. In addition, these systems tend to ignore the links such as *sameAs* which, while having well defined semantics, are considered in the same way as any other relation by SPARQL engines.

For this reason, in LOTED, we made the choice of including in the workflow an offline step of *entity reconciliation* (also called materialisation in [1]). The basic idea for this step is to aggregate locally, under one identifier, all the information from entities related with each other by *sameAs*. In our case, we decided to use the geonames URI to represent the location of the organisation in a tender, retrieving any information related to this URI from the geonames system. We then retrieve the URI of entities in DBpedia linking to the geonames object, and import all the information obtained by resolving this URI as

[7] `http://www.wikipedia.org/`
[8] `http://www.geonames.org/search.html`

attached to the geonames URI. In this way, all the information about a given place, from both geonames and DBpedia, ends up being aggregated in our triple store, under the corresponding geonames URI.

The creation and linking process described above has been running for more than 2 months at the time of writing (since the 12th May 2010). It collected around 55,000 procurement notices unequally distributed in the 14 sectors and 22 countries. These tenders relate to 5,000 places that are being linked to geonames and DBpedia.

2.3 Access Interfaces

The primary goal of building a linked data platform such as LOTED is to make available data in a machine readable and connected way, so that this data can be further linked to and exploited in applications. Therefore, all the URIs used in tender descriptions in LOTED resolve, and provide a complete representation of the corresponding objects in RDF. For example, `http://loted.eu/data/tender/204339-2010` can be accessed to obtain the complete representation of the tender number *204339-2010*. Similarly, `http://loted.eu/data/authorityName/Ville_de_Nice` is the URI for the particular organisation (Nice City Council) that created the tender and the same pattern applies to other types of objects in the data. The same kind of information is also available through a SPARQL endpoint, located at `http://loted.eu/Sparql` and for which a basic interface is available linking from the front page of the LOTED portal.

The portal also implements a straightforward Web user interface, from which people can find, retrieve and obtain information about tenders (see Figure 3). A specific country, sector and range of dates can be chosen, that will make appear the selection of tenders on a map, focusing on the selecting country. In practice, this selection is translated into a SPARQL query to obtain the precise coordinates of the cities that are related to tenders in the given sector, country and range of dates. Clicking on the marker on the map corresponding to a given city makes appear the list of tenders available in this city for the given sector and dates, which are further linked to their original documents on the TED portal, and to their RDF representations. It also displays information about that city as described in DBpedia. This can in particular include data about existing companies in this location. The RDF description of the current selection of tenders can also be obtained, as well as the SPARQL query to generate this RDF description, which can be further edited by the user to obtain a more precise selection.

In the next section, we show how we can use the links to DBpedia and geonames to create the global visualisations of the tenders which are available under the *Charts* button of the interface.

3 Analyzing the Data: Visualization of 'Tender Profiles'

As a way to demonstrate how linked data can benefit the analysis of data, and obtaining a better understanding of a global domain where large data is involved,

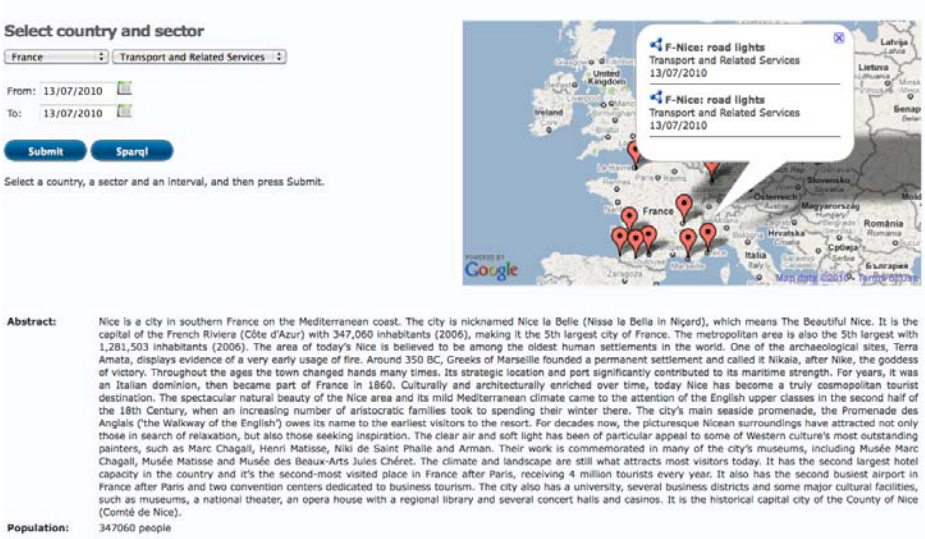

Fig. 3. Web user interface of the LOTED portal.

we included in the LOTED portal visualisations of *tender profiles* according to various dimensions. The idea of a tender profile is that it corresponds to the proportion of each sector in terms of the number of tenders being published in a particular place or by a particular group of organisations.

To illustrate this idea, we consider the most straightforward of these charts, shown in Figure 4. This chart shows the tender profiles using the country of origin as the main clustering dimension over the entire period starting at the beginning of the LOTED system (12th May 2010). As can be seen, different countries tend to have similar profiles, with however some variations in the focus on some of the sectors. Therefore, from there, it is possible to focus on a specific sector, ordering automatically the different countries according to their contribution to this sector. Doing so, we can then realise that one of the greatest discrepancies between countries is on the sector of "financial and related services", with Belgium having a larger proportion of tenders in this area, and countries such as Malta and Slovakia being almost absent from such a market. In addition, a table is presented associated with the chart that shows the number of tenders for each country. This is useful to assess whether some countries have a sufficient amount of tenders for the results to be significant. For example, we can notice that Malta only has had 66 tenders during this period, while several other countries had thousands. For some reasons, France produces significantly more tenders than any other countries (more than 14,000, compared to 6,500 for the second country, Germany). Finally, it is also possible to manually order the country in the chart, to try to find correlations with other dimensions than the sector (e.g., ordering the countries from East to West, to see if any regularity appears).

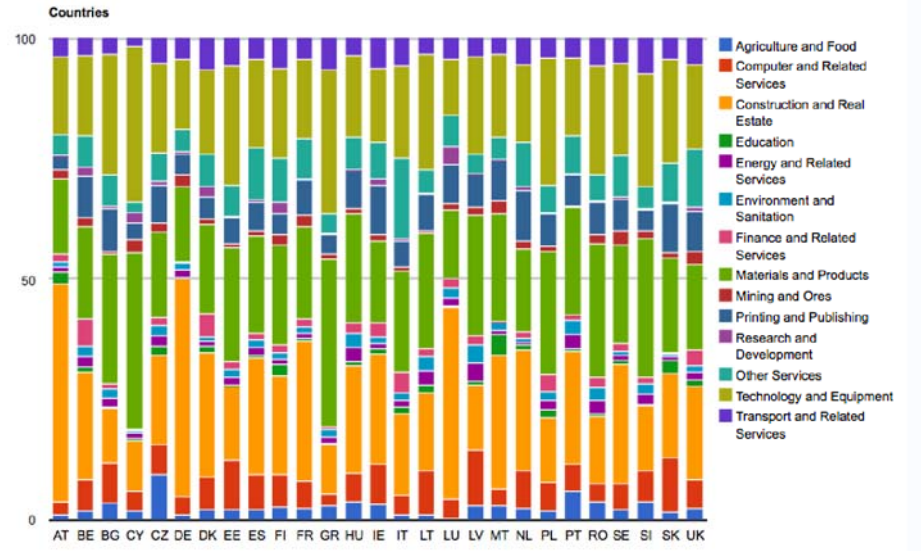

Fig. 4. Chart representing the tender profiles of the 22 countries in the LOTED system.

The chart relating tender profiles with country is very useful to obtain a general idea of the distribution of tenders across the entire set. However, what we want to show here is how the links to other datasets can provide additional, and more granular ways of analysis. Another chart included in the LOTED portal concerns specific regions within countries (see Figure 5 for the chart of the tender profiles by region in Italy). Indeed, for some countries, DBpedia provides the information on which sub-division a city is in. For Italy, we can identify 20 different regions with, like for countries, different amounts of tenders being published and different numbers of cities represented (this information is available in the table attached to the chart on the website).

In this case as well, tender profiles can be ordered by sector and manually to try to extract correlations between the regions and the sector in which they tend to publish tenders. This can be used to find the best place for different sectors, as the ones from which the most tenders originate, or the ones where a particular market has not developed yet. Using such a process, we can in particular devise the following table (Table 1) showing which region is the most and the least represented in each sector[9] (ignoring regions with less than 15 tenders):

Of course, one would need to be an expert in the economy of Italy to interpret these results appropriately. However, it appears unlikely for it to be a coincidence for example that Emilia-Romagna is first in both sectors of "education" and "research and development", while Umbria is last in these two, as

[9] In the spirit of `http://www.informationisbeautiful.net/visualizations/because-every-country-is-the-best-at-something/`

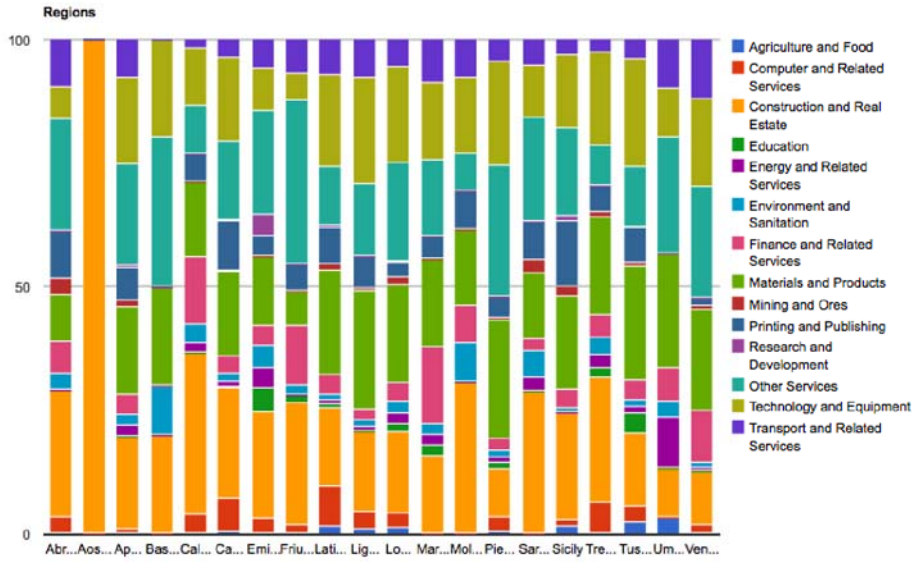

Fig. 5. Chart representing the tender profiles of 20 regions of Italy in the LOTED system.

Table 1. Most represented and least represented regions of Italy in each sector of tenders in LOTED.

Sector	*First region*	*Last region*
Agriculture and Food	Umbria	Abruzzo
Computer and Related Services	Latium	Marche
Construction and Real Estate	Calabria	Piedmont
Education	Emilia-Romagna	Umbria
Energy and Related Services	Umbria	Sicily
Environment and Sanitation	Sardinia	Sicily
Finance and Related Services	Marche	Liguria
Materials and Products	Piedmont	Friuli-Venezia Giulia
Mining and Ores	Abruzzo	Friuli-Venezia Giulia
Printing and Publishing	Sicily	Umbria
Research and Development	Emilia-Romagna	Umbria
Other Services	Friuli-Venezia Giulia	Trentino-Alto Adige/Sdtirol
Technology and Equipment	Tuscany	Friuli-Venezia Giulia

well as in "printing and publishing".

To illustrate further how additional data brought into the initial dataset of tenders can be used to add originally unintended dimensions for data analysis, we also computed the chart of the tender profile in a given country, depending on the political affiliation of the cities the tender originates from (see Figure 6 for the chart for France). The political affiliation of a city can be found in DBpedia, either directly attached to the city (under the property `party`), or as a characteristic of the mayor (through the property `mayorParty`). Such information is however very heterogeneously present across countries and cities.

The basic idea here is that it would appear natural that different parties would have a different focus when it comes to public spending and that making emerge these different profiles can show users the influence of local politics. However, as can be seen from Figure 6, the tender profiles of the 2 major political parties in France are surprisingly similar. This is valid also for other countries where sufficient data can be obtained. This consistency within a country is even more surprising considering that, as shown previously, it is a lot less apparent across different countries.

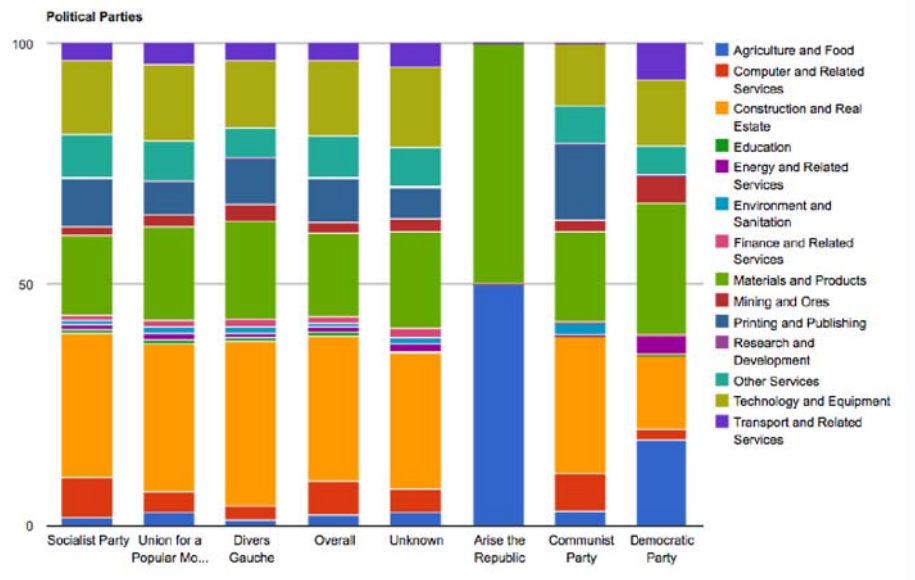

Fig. 6. Chart representing the tender profiles of political parties in France.

It is worth noticing here two additional columns that have been added to the chart: "Overall" and "Unknown". Overall corresponds to the average tender profiles of all the cities for which the political party is known, while Unknown correspond to the ones for which the political party is not provided by DBpedia.

This allows us to measure the bias introduced by incomplete information, by looking at the dissimilarities between these two profiles.

4 Lessons Learnt

Generally, one of the most important lessons learnt from building an application such as LOTED is that there are obvious challenges in trying to *extract high level knowledge from interlinked datasets*. First of all, the incompleteness of the linked data cloud, and the general uncertainty regarding this incompleteness, appear clearly as major problems in analysing tender data using the visualisations presented above. Indeed, information as basic as the region in which a particular city is cannot be assumed to be always present. Heterogeneity also appears as a major issue, with many different properties used to represent the same information in DBpedia, as well as redundancies and variations that have to be manually cleaned up. Of course, it can be argued that such issues are specific to the considered dataset, DBpedia, and that geonames for example does not suffer from such issues. Indeed there seem to be two distinct sorts of linked datasets currently available, of which DBpedia and geonames seem to be the stereotypes: general purpose, heterogeneous, incomplete datasets, and focused, homogeneous and clean datasets. Having said that, ways for an application developer to realise in which category a dataset is and what can be expected from it seem crucially needed. This element appears especially critical in application such as LOTED which intend to provide some form of linked-data based data analysis, where the discrepancies in the representation of different resources can introduce a bias rendering the results of the analysis impossible to interpret.

At a lower level of granularity, our experience also made emerge the lack of support for the lifecycle of linked data applications. Indeed, we already discussed the need for us to realise the task of entity reconciliation offline and in an ad-hoc manner, due to the unavailability of tools to exploit links between datasets at run-time. This has obvious disadvantages, but also shows a need for generic frameworks to support common tasks in linked data applications, including data cleaning, URI creation (generating valid, consistent URIs from arbitrary unicode strings), data discovery, linking, link storage, link exploitation, etc. There have not yet been many applications exploiting linked data to a significant level. One of the reasons might be that, as we experienced, a lot of efforts need to be spent on setting up the basic underlying infrastructure, with every application starting almost from scratch.

The availability of such a generic framework for linked data applications, including reusable components implementing common tasks such as the ones listed above beyond the simple query engine/triple store, would then make possible the development of data analysis framework for linked data, able to find relevant links, and new dimensions to enrich an existing dataset, making possible the discovery of new knowledge from the created connections.

5 Conclusion

In this paper, we have presented the LOTED linked data application for European public procurement. There have not been many applications of linked data until now that where able to really exploit links to external dataset to provide additional functionalities. We can for example mention DBRec [3], a music recommender system exploiting DBpedia to help users in finding music, or simpler applications such as described for example in [4, 5]. While relatively simple, LOTED demonstrates how data analysis can be supported by linked data, including external, originally unintended perspectives into a dataset to potentially make emerge new high level knowledge about the considered domain. While this application and the general idea are still at an early stage, the obtained results have highlighted the new possibilities associated with the approach, showing the potential of being able to create data visualisations seamlessly combining dimensions from various datasets on the Web of Data.

We also report in this paper on how the current state of linked data and of the corresponding tools is hampering the development of such applications, leading to a need for a generic framework, a platform or toolkit to support the developers in realising the common, necessary tasks. Such a framework would allow applications such as LOTED to evolve from 'proofs-of-concept' of what the Web of data can help achieve, to concrete, real-life applications. In relation to this, we are currently exploring new analysis mechanisms on top of the data, extracting and explaining trends in the various sectors covered by LOTED tenders, in order to support the real needs of the potential users of the data, in relation with existing work in the area of business intelligence.

Acknowledgement

The authors would like to thank Stefano Bertolo, Aldo Gangemi, Jose Manuel Gomez Perez and Jeff Pan for their ongoing contributions to the work on LOTED.

References

1. Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K.U., Umbrich, J.: Data summaries for on-demand queries over linked data. In: 19th International World Wide Web Conference (WWW2010). (April 2010)
2. Hartig, O., Bizer, C., Freytag, J.C.: Executing SPARQL queries over the web of linked data. In: ISWC 2009: Proceedings of the 8th International Semantic Web Conference, Chantilly, VA, USA. (2009) 293–309
3. Passant, A., Decker, S.: Hey! ho! let?s go! explanatory music recommendations with dbrec. In: 7th Extended Semantic Web Conference, ESWC Demo session. (2010)
4. Hausenblas, M.: Exploiting linked data to build web applications. IEEE Internet Computing **13** (2009) 68–73
5. Bizer, C., Heath, T., Berners-Lee, T.: Linked data – the story so far. International Journal on Semantic Web and Information Systems **5**(3) (2009) 1–22

Ontology of Ontology Patterns as Linked Data Integration Tool

Miroslav Vacura and Vojtěch Svátek

Faculty of Informatics and Statistics,
University of Economics
W. Churchill Sq.4, 130 67 Prague 3,
Czech Republic
vacuram|svatek@vse.cz

Abstract. The paper present preview of ontology of ontology design patterns and transformation patterns being developed as support tool for emerging ontology design techniques and methodologies.

The Linked Data initiative was started by Tim Berners-Lee as an architectural vision for the Semantic Web. It explores the idea of Semantic Web as puting emphasis on making links so both people and machines can explore the interconnected web of data [1].

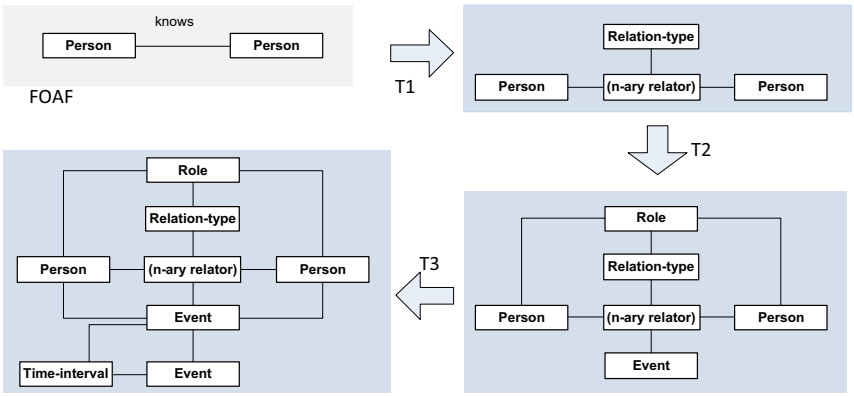

Fig. 1. FOAF Transformations

As an use-case we choose the FOAF project and it's 'knows' relation. Since 2004 there was more than 1 million FOAF documents and 79% of them utilized the knows property [2]. The typical needs of ontology engineer working on top of some Linked Data source comprises of transformation or aligning data to some more complex ontology either newly designed or already existing. This is also case of knows relation that in its FOAF implementation is very simple and doesn't allow expressing more complex relations among individuals.

In the case of newly designed ontology use of ontology design patterns (ODPs) proved the most effective and the least time consuming way of doing it. In context of our use-case we can think of several iterations of ODPs that represent more or less complex or expanded view on 'knows' relation as is depicted on Fig. 1. These ODPs can be connected together using predefined transformation patterns (OPPL), It seems to us that having library of such predefined pattern transformations at hand could make such design tasks easier and much faster. Our ongoing work proposes development of such library on top of existing portal `OntologyDesignPatterns.org`, but 1) in form of ontology, 2) with explicitly stated relations, 3) that are formally defined and 4) with appropriate transformations (OPPL) between related patterns, that enable automatic transformation from one pattern to another. We also focus on providing more fine-grained analysis of relations (like specialization/generalization) between ontology ODPs.

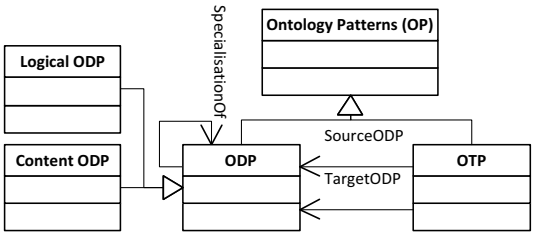

Fig. 2. Design Patterns Transformations Ontology

Such a library would could be easily integrated with methodologies like *Extreme Design* (XD) [3] and respective development tools like NeON or Protegé. This extended abstracts presents early preview of architecture of ontology being developed on Fig. 2. For additional information see `http://keg.vse.cz`.

Acknowledgments

This work has been partially partially supported by the IGS 4/2010 and by the CSF grant no. P202/10/1825 (PatOMat - Automation of Ontology Pattern Detection and Exploitation).

References

1. Tim Berners-Lee. LinkedData. `http://www.w3.org/DesignIssues/LinkedData.html`, 2009.
2. Li Ding, Lina Zhou, Tim Finin, and Anupam Joshi. How the semantic web is being used: An analysis of foaf documents. In *Proc. of the 38th ICSS*, 2005.
3. Valentina Presutti et al. eXtreme Design with Content Ontology Design Patterns. In Eva Blomqvist et. al., editor, *Proc. of the WOP 2009, coll. with ISWC-2009*, volume 516. CEUR Workshop Proceedings, 2009.

Potentials of enriching the Web of Documents with Linked Data by generating RDFa markup

Benjamin Adrian

[1] Knowledge Management Department, DFKI
Kaiserslautern, Germany
[2] CS Department, University of Kaiserslautern
Kaiserslautern, Germany
benjamin.adrian@dfki.de

Abstract. Linked Data is out there. It consists of data about various topics in a range from human health care to Pop Music or product information. While on a web search users like Sarah, Pete, and Tom would like to see this data while reading and browsing the web of documents.
The position of this paper figures out the potentials of enriching web pages with linked data content according to the specific information demand the user has in his current situation.
Tools that automatically enrich web pages with RDFa content from Linked Data knowledge bases are considered to be the next step to really use the web of data while browsing the web of documents.

1 Motivation

Please consider Sarah, a PHD student. Sarah wants to buy a laptop and has a budget of 600$. Sarah has a tight schedule, so her plan is to search online to quickly compare different offers regarding to product properties and existing reviews from customers. Fortunately, Sarah knows Linked Data and she was happy to see the Openlink Virtuoso Sponger that would give her access to Amazon.com data in RDF format. She knows Amazon to have a large knowledge base about products and vendors, their offers, and user generated ratings and experience reports about using these products. Sarah likes Amazon and its marketplace, but wants to give all online vendors a chance. Finally she wants the cheapest offer for the best laptop she can get with 600$. Therefore she wants a projection of Amazon's product data to products mentioned in web pages of online shops she found while searching Google products or Yahoo's Search Monkey.

Imagine Pete, a high school student. Pete is reading an exciting thriller on his iPad. Pete loves to have a more colorful imagination about concrete sets and locations where actions take place in. Therefore he installed a fancy App that uses data from DBpedia and LinkedGeoData to produce a mashup on Google map and Streetview consisting of scenery pictures, satellite images, and links to additional background information about heritages, famous buildings, battlegrounds, etc.

Tom is a young entrepreneur. He really knows about the power of social platforms like Twitter or Facebook. Tom knows many experts, friends and customers inside these platforms and he likes to know about their opinion about some new technologies and products Tom is reading about in blogs or other web pages. Tom installed a browser plugin that uses the RDF data published by Twitter and Facebook to get the latest tweets, comments and blog entries from twitter and Facebook from his contacts about topics mentioned in these web pages.

2 Position

The tools Sarah, Pete, and Tom use have one in common: they use RDF data published as Linked Data to enrich existing instance mentions (textual phrases that refer to existing instances in the RDF data) with RDFa markup. This markup explicates the reference between a phrase in text and the instance within a data set. The tools of Sarah, Pete, and Tom depend on different Linked Data sets but enrich the document content with additional information that:

- helps Sarah to face more product offers and reviews about a certain Laptop she found during her web search,
- helps Pete in stimulating his imagination while looking at real pictures and maps about the primeval forests near the small town Folks in Washington, USA which is the set of the latest book by Stephanie Meyer about vampires, Native Americans, and werewolves, Pete is currently reading.
- helps Tom in getting the opinions and comments of well regarded experts from his Twitter and Facebook account about topics he is reading about.

Behind the scenes the tools start with analyzing text passages in original web pages ...

From Port Angeles I carried on towards Forks on highway 101.

... then use Linked Data to explicate mentions in text with RDFa markup ...

From Port Angeles I carried on towards Forks on highway 101.

... which is then used by applications to request more information ...

```
dbpedia:Folks\%2C\_Washington  rdfs:label  "Folks" ;
   geo:lat          "47.950980"^xsd:double ;
   geo:long         "-124.384749"^xsd:double ;
   foaf:depiction   <http://upload.wikimedia.org/wikipedia/commons/1/14/Forks_WA.jpg>
   rdfs:seeAlso     <http://maps.google.de/maps?ll=47.951111,-124.384722&spn=0.25,0.25> .
```

... which is finally visualized in useful, inspiring, and interesting mashups.

Conclusion: We recommend to build more browser or proxy based RDFa generators that automatically enrich web pages with Linked Data that helps the user in his current situation. Tools such as Epiphany `http://projects.dfki.uni-kl.de/epiphany` might be a first step to a usable web of data.