UNIVERSITÄT KOBLENZ · LANDAU
Institut WeST

FB 4 Informatik

LOVER: Support for Modeling Data Using Linked Open Vocabularies

Thomas Gottron
Johann Schaible
Stefan Scheglmann
Ansgar Scherp

Nr. 2/2013

Arbeitsberichte aus dem Fachbereich Informatik

Universität Koblenz-Landau • Campus Koblenz • Universitätsstraße 1 • 56070 Koblenz
Telefon +49 261 287-0 • http://www.uni-koblenz.de

Arbeitsberichte des Fachbereichs Informatik

ISSN (Print): 1864-0346
ISSN (Online): 1864-0850

Kontaktdaten der Verfasser

Thomas Gottron, Johann Schaible, Stefan Scheglmann, Ansgar Scherp
Institut für WeST
Fachbereich Informatik
Universität Koblenz-Landau
Universitätsstraße 1
D-56070 Koblenz
E-Mail: gottron@uni-koblenz.de, johann.schaible@gesis.org, schegi@uni-koblenz.org, ansgar@informatik.uni-mannheim.de

LOVER: Support for Modeling Data Using Linked Open Vocabularies

(Technical Report)

Johann Schaible
GESIS - Leibniz Institute for the Social Sciences
Unter Sachsenhausen 6-8
50667 Cologne, Germany
johann.schaible@gesis.org

Thomas Gottron
WeST - Institute for Web Science and Technologies
Universitätsstraße 1
56070 Koblenz, Germany
gottron@uni-koblenz.org

Stefan Scheglmann
WeST - Institute for Web Science and Technologies
Universitätsstraße 1
56070 Koblenz, Germany
schegi@uni-koblenz.org

Ansgar Scherp
Institute for Business Informatics and Mathematics
B6, 26
68131 Mannheim, Germany
ansgar@informatik.uni-mannheim.de

ABSTRACT
Various best practices and principles guide an ontology engineer when modeling Linked Data. The choice of appropriate vocabularies is one essential aspect in the guidelines, as it leads to better interpretation, querying, and consumption of the data by Linked Data applications and users. In this paper, we present the various types of support features for an ontology engineer to model a Linked Data dataset, discuss existing tools and services with respect to these support features, and propose LOVER: a novel approach to support the ontology engineer in modeling a Linked Data dataset. We demonstrate that none of the existing tools and services incorporate all types of supporting features and illustrate the concept of LOVER, which supports the engineer by recommending appropriate classes and properties from existing and actively used vocabularies. Hereby, the recommendations are made on the basis of an iterative multimodal search. LOVER uses different, orthogonal information sources for finding terms, e.g. based on a best string match or schema information on other datasets published in the Linked Open Data cloud. We describe LOVER's recommendation mechanism in general and illustrate it along a real-life example from the social sciences domain.

Categories and Subject Descriptors
E.2 [**Data Storage Representations**]: Linked representations; H.3.3 [**Information Search and Retrieval**]: Search process, Selection process

General Terms
Design, Measurement

Keywords
Linked Data Modeling, Vocabulary Mapping, Support System

1. INTRODUCTION
The Linked Open Data (LOD) cloud comprises data from diverse domains, which is represented in RDF. To publish Linked Data, Bizer et al. provided a set of Linked Data guidelines [6], which were updated a few years later by Heath and Bizer [13]. These guidelines can be categorized into best practices regarding the naming of resources, linking and describing resources as well as a dereferenced representation, as it was done by Hogan et al. [14]. When modeling Linked Data the focus is on *describing resources*. This includes that the ontology engineer should rather re-use classes and properties from existing vocabularies than re-invent them, and mix several vocabularies where appropriate. Hogan et al. [14] consider the conformance of data providers with respect to these two guidelines, i.e., reusing vocabularies as "non-trivial" and as "very common practice" with respect to mixing vocabularies. To improve the re-use of more classes and properties, it seems that the following is required: (i) methods for the proposal and promotion of not only the most popular but also new, domain specific, and highly relevant vocabularies, in order to expand coverage; (ii) tools and search engines that support publishers to find the correct, most widely-adopted classes and properties for their needs. Furthermore, expressing data in RDF has analogies to creating an ontology. In general, ontology engineering is the field that studies the methodologies for building ontologies in a systematic way [11]. Thus, to improve creating a

5-star Linked Data dataset, the ontology engineer has to be supported with respect to the ontology engineering guidelines such as [18].

Modeling Linked Data generally requires an ontology engineer and a domain expert. The domain expert has to verify the semantic correctness of the properties and classes used to represent the data. The ontology engineer on the other hand has to verify the quality of the dataset with respect to the Linked Data guidelines, in order to make it as easy as possible for client applications to process the data. Regarding the aspects mentioned above under (i) and (ii) there are tools and services which promote diverse existing vocabularies. However, the ontology engineer still has to do a huge part of the modeling process manually. This is either very time consuming or it affects the semantic richness of the dataset. As a result, it might decrease the interoperability of the data, i.e., it makes it difficult for Linked Data applications to consume the data. This is due to the syntax of SPARQL queries, as it requires the user to specify the precise details of the structure of the RDF graph being queried. Thus, the user has to be familiar with the dataset, and if several datasets from the same domain are modeled differently, the user or the Linked Data application has to deal with the problem of schema heterogeneity [15].

Existing transformation tools like D2RQ [5], Triplify [3], Open Refine[1], Karma [16], and others transform input data into RDF. They provide a first step towards supporting the ontology engineer to model a Linked Data dataset. Hereby the hierarchical structure of the RDF is inherited from the input data, but the mapping to existing classes and properties and their search has to be done manually. Vocabulary search engines like Swoogle[8] or LOV[2] represent a further step, as they provide a possibility to automatically assess and retrieve vocabularies and their terms. However, to re-use such classes and properties from existing vocabularies still demands additional efforts, such as measuring the correctness of a mapping and incorporating the terms into the transformation system. The measurement of the semantic correctness of a mapping needs to be performed semi-automatically as it requires a human verification, preferably from a domain expert. A support feature which calculates such a measurement is not included in the existing tools. Only Karma [16] integrates a recommendation service for a mapping based on machine learning techniques. However, it uses only a single vocabulary and the user has to specify the vocabulary used for the recommendation manually.

To alleviate this situation and to take another step towards supporting an ontology engineer to model Linked Data, we present LOVER (short for: Linked Open Vocabulary EngineeRing). LOVER is a generic approach, which uses an iterative multimodal search mechanism to recommend the re-use of classes and properties from existing and actively used vocabularies in the LOD cloud. Hereby, LOVER incorporates different information sources such as Swoogle for string match based search and the SchemEX index [17], which contains a comprehensive directory of the use of all properties and concepts that appear in some dataset from the Billion Triple Challenge data from 2012. Hereby, LOVER is not restricted to these information sources and can be extended to incorporate additional sources such as LOV or Wordnet[3]. A multimodal search includes specific contextual information, such as the vocabularies already used in the model, which is used to obtain better fitting results. Using this information, LOVER iterates through all schema elements. During each iteration, it recommends a set of terms for every schema element, adapts, and updates the recommendations using the context information. This way, LOVER is most likely to achieve an optimized mapping. By this, it supports increasing the re-use of existing vocabularies and mixing an appropriate amount of different vocabularies. Furthermore, the search for vocabularies is integrated in the modeling system, which implies that the ontology engineer does not have to incorporate the terms manually.

The remainder of the paper is structured as follows: Section 2 provides a detailed discussion of the various types of support features for an ontology engineer to model a 5-star Linked Data dataset. In Section 3, we present existing tools and services which support an ontology engineer in modeling Linked Data and tabulate these tools and services with respect to the previously provided various types of support features. LOVER is described in Section 4, where we illustrate the general concept of our approach and demonstrate it in Section 5 on a real-life example from the social sciences domain. We provide a preliminary proof of concept of the LOVER approach based on the results from the example. In Section 6, we conclude our work and discuss the purpose of LOVER in comparison to existing approaches.

2. TYPES OF SUPPORT FOR MODELING LINKED DATA

If an ontology engineer re-uses classes and properties from existing vocabularies, it increases the probability that the resulting Linked Data dataset can be consumed by Linked Data applications without requiring further pre-processing of the data or modification of the application [13]. This applies especially if the applications are tuned to well-known vocabularies. But, how to support an ontology engineer to re-use existing classes and properties to generate a 5-star Linked Data dataset for general purpose?

(1) To re-use a "vocabulary" implies that the ontology engineer faces the problem of finding an appropriate term from a specific vocabulary. To support the ontology engineer to find a specific vocabulary, a system should provide a vocabulary search mechanism. The index of such a search engine should comprise established vocabularies as well as newly published and actively used vocabularies, in order to expand the general coverage.

(2) Furthermore, the search engine should provide meta information on promoted vocabularies such as the usage frequency in the LOD cloud, the degree of novelty of a vocabulary, and provenance information if available. These information help the ontology engineer to decide what vocabulary to use to describe the data.

[1] `https://github.com/OpenRefine` accessed 2012/12/14
[2] `http://lov.okfn.org/dataset/lov/` accessed 2012/12/14
[3] `http://wordnet.princeton.edu/` accessed 2012/12/14

(3) Subsequently, the ontology engineer has to decide whether he is going to use a "class" or a "property" from a specific vocabulary. For this, he requires information on the semantics of every property and class. For example, the ontology engineer intends to express the data element "publication". Re-using the Semantic Web for Research Communities (SWRC[4]) ontology, he can utilize the property `swrc:publication` or the class `swrc:Publication`. Information on the semantics of these terms, e.g. `swrc:publication` denotes that an author has a publication and `swrc:Publication` denotes a document which is published, would help the engineer to make the right decision. This also applies for deciding between multiple classes and properties, which are "perfect" for describing a data element. For example, using `dcterms:date` is appropriate to express temporal information, but it might be semantically more appropriate to use `dcterms:issued` for expressing the temporal information on a formal issuance of the resource. Therefore, to support the ontology engineer in making such decisions, a system should provide information on the semantics of a term along with the search of a term if possible.

(4) Furthermore, the engineer requires meta information on terms to re-use a specific class or property. Such meta information would include the popularity of a specific term, the most common type of usage, and providing datasets which utilize this term. The popularity of a term indicates whether it is used by many or no data providers. The most common type of usage shows the engineer how the term is used by other data providers, e.g. the property `akt:has-Author` from the Aktors ontology[5] is used 15 times as an object property and 2 times as a datatype property.[6] In addition, the most common kind of usage is to apply this object property between the classes `akt:Book-Section-Reference` and `akt:Person`. Another additional information on the kind of usage would be to provide all possible ranges of an object property. To provide datasets which utilize a specific term, the ontology engineer is able to choose a class or property used by a data provider he intends to link to. This kind of meta information helps the engineer to decide the correct usage of the term increasing the probability of consumption of the dataset. Therefore, to support the engineer, a system should also provide meta information on terms.

(5) A term can be re-used to express a data element or it can be re-used as a parent class/property of a newly defined class or property. For example, the ontology engineer needs to specify a class "Very Important Person" which is a subclass of a Person. The engineer defines a class `ex:Very-Important-Person` but he must also specify this class in a hierarchical structure. He re-uses the class `foaf:Person` and specifies `ex:Very-Important-Person rdfs:subClassOf foaf:Person`. To support the ontology engineer to implement this modeling aspect is a crucial step forward to the interlinking of different vocabularies.

(6) When re-using and mixing existing vocabularies, a data provider is able to describe information from different domains e.g. re-use of FOAF for information about people and Dublin Core for metadata about documents. The re-use of existing vocabularies therefore also implies the guideline of mixing a reasonable amount of different vocabularies. If an ontology engineer re-uses only a few vocabularies and if many data elements have been mapped to terms that do not express the full semantic richness of the data element, it might result in a semantically poor dataset. If the engineer uses too many vocabularies, consuming the data will become inefficient as this levels up the expressiveness of the dataset making it less performant for reasoning techniques and consumption [2]. In average, 8.6 namespaces are used per pay-level domain as a recent investigation of Hogan et al. [14] shows. Therefore, if an engineer re-uses a vocabulary, he should examine whether other data elements can be mapped to terms from the same vocabulary with little to no loss of the semantic richness. For example, if a property is modeled with `dcterms:creator` and the FOAF vocabulary is not used yet, it might be more reasonable to use `dcterms:Actor` as range instead of `foaf:Person`. Supporting the engineer in this effort would alleviate the entire process of making decisions regarding the amount of re-used vocabularies.

(7) As the process of modeling a Linked Data dataset is generally comparable to the engineering of an ontology, another type of support is to follow general ontology engineering principles. In this work, we focus on the "Ontology Development 101" which is provided by [18], where Noy et al. discuss the general issues considering one possible process for developing an ontology. The authors describe an iterative approach to ontology engineering, which starts with a rough sketch of the ontology and evolves by refining it and filling in the details. Throughout this process, the engineer has to make a lot of modeling decisions. To support the ontology engineer in this process would enable him to produce a Linked Data dataset in an easier way.

(8) While modeling a Linked Data dataset, an ontology engineer makes decisions in cooperation with a domain expert. Therefore, to support the ontology engineer to find appropriate terms, we consider all types of support to be semi-automatic, i.e., they are based on calculating a relevance metric and the results have to be verified by the engineer. Otherwise, a fully automatic support might permit too many false modeling decisions.

3. EXISTING SUPPORT SYSTEMS FOR MODELING LINKED DATA

In Section 3.1, we present existing tools and services which can be used to express a dataset in RDF. To this end, we have selected the most "prominent" tools and services which support the ontology engineer to model a Linked Data dataset in some way. By "prominent" we mean, that the selected tools and services were mentioned by many references which outline the process of generating a Linked Data dataset. To the best of our knowledge, there is no tool which provides more support for the ontology engineer than the ones we mention in the following subsection. Subsequently, in Section 3.2 we provide a tabular comparison of these tools and services with respect to the types of support mentioned in the previous section and discuss whether the tools and services provide any of these types or not.

[4] `http://ontoware.org/swrc/` accessed 2012/12/15

[5] `http://www.aktors.org/publications/ontology/` accessed 2012/12/15

[6] Results are obtained from the Semantic Web search engine Swoogle on Dec 21, 2012

3.1 Existing Tools and Services

Existing tools and services used to publish Linked Data or model a dataset in RDF can be divided into three categories:

(a) Ontology development tools which allow creating RDF classes and properties from scratch and by using templates,

(b) transformation tools which transform data, like CSV, into a RDF representation, and

(c) existing ontology recommendation systems used to recommend properties and classes from established domain specific vocabularies.

However, the search for vocabularies is not included in any of the categories and has to be done manually. Thus, we mention the vocabulary search engines as well and include them into our tabular comparison.

3.1.1 Vocabulary Search Engines

Vocabulary search engines can be categorized by their approach and support for finding terms and by the results they provide. Basically, a search can be performed by utilizing a keyword-based or a SPARQL-query-based approach. The keyword-based approach uses the keywords to execute syntactic and semantic similarity analyses on the search index and provides a ranked list of results. Such results might be vocabularies, terms from vocabularies including meta information, or links to the vocabulary providers. The SPARQL-query-based approach utilizes a specific SPARQL query to gather aggregated information and provides single resources or whole triples as a result containing specific information on the usage of the terms and their vocabularies.

Swoogle [8] and LOV[7] are the most popular directories of ontologies that allow a keyword-based search for classes and properties. Swoogle contains over 10,000 ontologies and the LOV index comprises over 300 established vocabulary spaces in the Linked Open Data cloud. Both provide filter functions for the result set as well as a brief and detailed presentation of results. Furthermore, Swoogle offers an API for Linked Data applications to use the search functionality. There is also a more sophisticated ranking of the concepts and properties in Swoogle pursued by Ding et al. [9]. Other directories of vocabularies on the LOD cloud are vocab.cc[8] providing lists of the top 100 classes and properties in the Billion Triple Challenge 2011 data set and prefix.cc[9] which is a service for looking up URI prefixes. The results of both services implicate a link to the original vocabulary where the engineer can gather additional information. Alani et al. [1] present a different approach for searching ontologies from different domains. When looking for ontologies on a particular topic, they retrieve a collection of terms that represent the given domain from the Web and then use these terms to expand the user query. Other types of search engines such as the SchemEX index [17] and LODatio [12] use SPARQL queries to gather information about datasets published in the LOD cloud. Such information can be derived on schema level as well as on instance level. The SchemEX index contains a comprehensive directory of all properties and concepts that appear in data observed on the LOD cloud. This directory also includes the original location of the data enriched with additional meta data. Thus, SchemEX can be used to gather relevant information on vocabularies, like its actual popularity in the LOD cloud on the level of data sources as well as instances, and information on schema usage. For example, SchemEX can provide the most common object property between to concepts, i.e., respond to information needs such as "provide me a ranked list of concepts that are connecting some other concept by property p". LODatio makes use of this schema index and displays examples, estimating the size of the result set and allows for advanced functions, such as ranking, query generalization (Did you mean?) and query specification (Related queries).

3.1.2 Ontology Development Tools

Traditional ontology engineering methods and tools aim at creating an ontological representation of a specific part of the world, i.e., the domain that is under consideration. They typically do not explicitly consider the modeling and publishing of Linked Data and thus do not foresee a mapping of the created ontology to some schema. However, they allow the ontology engineer to develop a model according to the Ontology Engineering Principles, as they were mentioned in Section 2, and also to complement existing ontologies.

Protégé[10], the TopBraid Composer[11], the NeOn Toolkit[12], and Neologism [4] are the most commonly used ontology development tools from a list of ontology tools[13]. Protégé is an ontology editor and knowledge-based framework, which supports modeling ontologies and provides a plug-and-play environment that makes it a flexible basis for rapid prototyping and application development. Using Protégé, the ontology engineer is able to develop an ontology according to [18]. The TopBraid Composer and the NeOn toolkit are both an ontology development environment and also allow the engineer to develop an ontology with respect to the Ontology Engineering Principles. Both are implemented based on the Eclipse platform, which makes the tools highly customizable as well. The web-based vocabulary editor Neologism includes promoting vocabulary development and maintenance. Neologism is no ontology editor like Protégé or TopBraid Composer as it does not support to build large and complex domain ontologies, nor to express the full power of OWL, and it does not provide sophisticated reasoning services. Its primary goal is to reduce the time required to create, publish, and modify vocabularies for the web of data.

3.1.3 Data Transformation Tools

Data transformation tools provide an automatic and semi-automatic support for transforming data from one representation into another, in our case into RDF. Such tools can be categorized by the input data they transform, e.g., is the data stored in a relational database or as a flat file, like CSV or XML.

[7] `http://lov.okfn.org/dataset/lov/` accessed 2012/12/14
[8] `http://www.vocab.cc/` accessed 2012/12/15
[9] `http://prefix.cc/` accessed 2012/12/15

[10] `http://protege.stanford.edu/` accessed 2012/12/15
[11] `http://www.topquadrant.com/index.html` accessed 2012/12/15
[12] `http://neon-toolkit.org/wiki/Main_Page` accessed 2012/12/20
[13] `http://techwiki.openstructs.org/index.php/Ontology_Tools` accessed 2012/12/15

An overview of different tools for transforming data from relational databases to RDF is provided in the survey by Sahoo et al. [20]. Their approaches are classified into "automatic mapping generation" and "domain semantics-driven mapping". Regarding the automatic mapping, one simply maps a RDB record to a RDF node, a column name from a RDB table to a RDF predicate and a RDB table cell to a RDF value. Such automatic mappings can serve as a starting point for a more customized and semantics-driven mapping, which actually aims at integrating and using existing vocabularies as output of the mapping process. Most prominent examples of such tools are Virtuoso[14], D2RQ [5], and Triplify [3]. For example, in D2RQ the ontology engineer manually defines a mapping document that contains SQL queries and the RDF classes and properties their results are mapped to.

To transform a flat file into RDF, there are tools like Karma [16]. the Datalift platform [21] and Open Refine[15]. In general, such tools allow modeling, cleaning up, and transforming data from one format into some other including RDF. In the case of Open Refine, writing RDF data is possible through the RDF Refine extension[16]. It allows for data reconciliation with SPARQL endpoints as well as RDF dumps and it supports searching the LOD cloud for existing data sets. Karma is a data integration tool which is similar to Open Refine regarding the usability. In addition, Karma provides a learning algorithm which can recommend terms from a pre-defined vocabulary as a column name.

3.1.4 Vocabulary Recommender Systems

Regarding concrete tools for recommending classes and properties from established vocabularies in the ontology engineering process, there are only a few approaches which are based on syntactic and semantic similarity measures and several other algorithms defining the popularity of a term. However, they all have a fixed index of vocabularies limited to specific domains.

Fernandez et. al.[10, 7] developed the collaborative system CORE (short for: Collaborative Ontology Reuse and Evaluation). As input, CORE receives a problem description in form of a set of initial terms and determines a ranked list of ontologies, which are considered appropriate for re-use. The approach uses WordNet[17] to expand the initial set of terms, performs keyword-based searches on its fixed ontology index, and finally evaluates the returned ontologies based on semantic correctness, popularity and other criteria. An additional component of the system uses manual user evaluations of the ontologies to raise the quality of the ranked list. A similar system was developed my Romero et al.[19]. The authors did not provide a specific name, so we will refer to it as "Romero's approach". The recommendation process is similar to the one of CORE, but it makes use of Web 2.0 mechanisms to measure the popularity of an ontology, e.g. the number of appearances in Wikipedia or bookmarks on Del.icio.us[18].

3.2 Tabular Comparison

Table 1 shows all mentioned tools and services with respect to the types of support we have presented in Section 2. No tool comprises all types to support the ontology engineer in modeling a Linked Data dataset. In fact, most systems provide support regarding only one or two types. If a tool or service provides a specific type of support for the ontology engineer, me mark it with ✓. If no such support features is existent, we denote it with an ×. Finally, if a tool or service allows a specific functionality but does not provide explicitly support for it, we mark it with ∼.

During the analysis of every tool and service, we focused on each of the types of support for the ontology engineer we have provided in Section 2. To present the results in a systematic way, we have structured the following discussion according to these types of support.

(1) No data transformation or ontology engineering tool provides the possibility to search for a term or vocabulary. Only the Datalift platform enables the ontology engineer to select vocabularies from a pre-defined vocabulary list which was composed by experts from various domains. Both ontology recommendation systems CORE and Romero's approach include the same sort of vocabulary search as the Datalift platform with the limitation of operating only on ontologies from one specific domain defined in the repository. Of course all the vocabulary search services include a specific search mechanism for finding established vocabularies.

(2) Only LOV, Swoogle, SchemEX, and LODatio provide rich meta information on the vocabularies they index. LOV and Swoogle display metadata about a vocabulary which is provided by the vocabulary publisher, whereas SchemEX and LODatio provide information on how and by whom the vocabulary is used within the LOD cloud. The service vocab.cc only provides a link to the RDF representation of the vocabulary. CORE and Romero's approach also provide meta information, but this meta information basically corresponds to the popularity of a vocabulary based on the collaborative approach in CORE or the Web 2.0 approach in Romero's approach.

(3) Only Karma provides a feature, where the ontology engineer can select a class or property for a mapping of a data element. Hereby, the recommendation mechanism regards the semantic correctness and classifies between classes and properties. As LOV and Swoogle incorporate the metadata provided by the vocabulary publisher, which also includes human readable comments on how to use the terms semantically correct, they are able to provide this information to the ontology engineer. SchemEX and LODatio on the other hand, extract the aggregated information from the schema index in order to conclude what term was used in what specific scenario. Therefore, they allow the engineer to retrieve this information but do not provide support for him to use such a feature. Regarding the ontology engineering tools, none provides information on how to use a specific term.

[14] `http://virtuoso.openlinksw.com/whitepapers/relational\%20rdf\%20views\%20mapping.html` accessed 2012/12/15

[15] `https://github.com/OpenRefine/OpenRefine/wiki` accessed 2012/12/14

[16] `http://refine.deri.ie/` accessed 2012/12/14

[17] `http://wordnet.princeton.edu/` accessed 2012/12/15

[18] `http://delicious.com/` accessed 2012/12/15

Table 1: Existing Tools and Services and Their Support for the Ontology Engineer

	(1) Vocabulary Search	(2) Vocabulary Meta Information	(3) Semantic Usage of Terms	(4) Meta Information on Terms	(5) Complement Existing Vocabularies	(6) Appropriate Vocabulary Mixing	(7) Ontology Engineering Principles	(8) Semi-Automatic Support
Ontology Development Systems								
Protégé	×	×	×	×	∼	∼	∼	✓
TopBraid	×	×	×	×	∼	∼	∼	✓
Neon Toolkit	×	×	×	×	∼	∼	∼	✓
Neologism	×	×	×	×	✓	∼	∼	✓
Data Transformation Tools								
D2RQ	×	×	×	×	×	∼	×	∼
Virtuoso	×	×	×	×	×	∼	×	∼
Triplify	×	×	×	×	×	∼	×	∼
Datalift	✓	×	×	×	×	✓	∼	∼
Open Refine	×	×	×	×	×	∼	∼	✓
Karma	×	×	✓	✓	×	×	∼	✓
Vocabulary Recommender Systems								
CORE	✓	✓	×	∼	×	✓	∼	✓
Romero's Approach	✓	✓	×	∼	×	✓	×	×
Vocabulary Search Engines								
LOV	✓	✓	✓	✓	×	×	∼	✓
Swoogle	✓	✓	✓	✓	×	×	∼	✓
vocab.cc	✓	×	∼	×	×	×	∼	✓
SchemEX	✓	✓	∼	✓	×	∼	∼	✓
LODatio	✓	✓	∼	✓	×	∼	∼	✓

(4) SchemEX and LODatio provide meta information. Whereas LOV and Swoogle provide meta information from the vocabulary published and further data such as a calculated amount of usage of a term, SchemEX and LODatio provide information on how many data providers use a specific term, how often is a specific term actively used, and in what context. CORE and Romero's approach do not necessarily provide support for the ontology engineer regarding meta information on terms, but they allow to investigate a term including comments from the vocabulary publisher.

(5) Neologism is the only ontology engineering tool, which provides active support to complement an existing vocabulary by letting the engineer to import an existing vocabulary and to complement it with additional terms. This is also possible with Protégé, TopBraid Composer, and the NeOn Toolkit, but they do not actively support the engineer to use this feature.

(6) This type of support is provided by the majority of the systems. The Datalift platform comprises a repository of different vocabularies, which is provided to the ontology engineer. Hence, he is supported in re-using several vocabularies. Only for Karma the user has to import one particular vocabulary that is used for mapping the data elements, which implicates that no vocabularies can be mixed. All other systems allow this feature, but do not support the engineer to utilize it.

(7) Regarding the Ontology Engineering Principles, some systems allow the ontology engineer to follow these principles, but do not provide explicit support motivating him to model a dataset accordingly. D2RQ, Triplify, and Virtuoso do not allow to transform data according to these principles, since every transformation has to be done from beginning. This means, the ontology engineer might compile a first draft, but to refine it, he has to repeat the process and adjust the parameters of the transformation. Romero's approach follows an automatic computation for recommending appropriate terms. This way, it cannot refine the resulting dataset, as the metrics for calculations are not changeable. Every search engine allows first to search for specific domain ontologies and then to search for only classes or properties. During this process the ontology engineer becomes more and more concrete in his search request.

(8) If a tool or service provides some kind of support, most of the times it is a semi-automatic support. All ontology engineering tools allow the engineer to develop an ontology using a wizard, or choose specific relationships between concepts. All these supporting features require a verification by the engineer. D2R, Virtuoso, and Triplify allow the engineer to make the final decision, but do not support him to choose between several possibilities. Open Refine gathers several schema elements which have a similar label and recommends the ontology engineer to merge these schema elements by using one label. This way, the engineer is able to make the final decision, which makes this support feature semi-automatic. Karma's support for choosing an appropriate mapping is also semi-automatic. Datalift does not provide such a support, but it still allows the engineer to verify every refining step. CORE is the only recommendation system which provides the engineer a set of terms he can choose from. Since Romero's approach is based on automatic computation, it is fully automatic and does not provide the engineer the possibility to verify the mapping. Finally, all search services are semi-automatic, as they only provide a ranked list of terms from which the engineer has to choose.

In conclusion to the discussion of the existing tools and services, we can say that every system provides support for one or two guidelines, but none of them provides support for all of them. Therefore, it is a picture of an isolated application context and the ontology engineer has to use several systems, in order to obtain as much support as possible.

4. THE LOVER APPROACH

To alleviate the situation, we propose LOVER (short for: Linked Open Vocabulary EngineeRing). LOVER is a generic approach, which uses an iterative multimodal search mechanism to recommend the ontology engineer classes and properties from existing and actively used vocabularies in the LOD cloud. This way, the ontology engineer is able to re-use such terms to represent data in RDF. Hereby, LOVER incorporates different information sources such as Swoogle or the SchemEX index to gather relevant information on vocabularies mentioned in the previous section. Therefore, LOVER is able support the ontology engineer regarding all of the eight types of support stated above.

(1) LOVER contains a search mechanism, which supports the ontology engineer to search for classes and properties. (2) It provides meta information on vocabularies, such as its specific domain and the relative and the absolute number of occurrences in the LOD cloud. (3) LOVER provides information on the semantic correct usage of a term as it retrieves the information from Swoogle or other information sources and (4) further meta information on terms using the same information sources. (5) It supports the engineer in complementing existing vocabularies, as the iterative approach also updates its recommendation for a mapping after defining a class hierarchy. This way, it is possible to invent a class, map another class to an existing term, and set a subclass relationship between them. (6) LOVER supports the ontology engineer to mix a reasonable amount of vocabularies, since it tries to map all data elements to a vocabulary as soon as this vocabulary is incorporated. (7) LOVER is an iterative approach allowing the ontology engineer to design a first draft and then refine the data model by adding further details. Hereby, it supports the engineer by letting him first enumerate data elements, define classes and their hierarchy, and finally object and annotation properties. (8) Every step has to be verified by the ontology engineer. Thus, it is in every aspect a semi-automatic approach.

In the following, we provide a more detailed illustration of the multimodal search functionality from different sources in general and apply it to our context of searching for re-usable classes and properties. Subsequently, we explain how an iterative approach is advantageous in creating a vocabulary for publishing a dataset as Linked Data. Finally, we illustrate how LOVER's iterative approach supports the ontology engineer in developing a Linked Data dataset according to the ontology engineering principles. Hereby, we will illustrate where LOVER provides the different types of support for the ontology engineer.

A multimodal search utilizes several methodologies to retrieve information. Such methodologies can be a search by keywords and a search by concepts, where every concept is derived from a specific context. This allows the user to specify the search more precisely. The multimodal search component in LOVER includes context information about the dataset which has to be modeled as Linked Data. Such context information can be used to provide better search results with respect to the types of support we have identified. It contains information on the vocabularies already used in the dataset, information whether the search is for a property or for a class, as well as whether the data element is a domain, range, object property, or annotation property. By using the Swoogle API for an exact string match, LOVER is able to retrieve information provided by Swoogle on specific terms and vocabularies. This includes for example the number of a specific usage of a term, e.g. `dc:creator` is used 386 times as annotation property (as retrieved on 22/1/2013). The SchemEX index provides information on effective schema design in the LOD cloud, i.e., how properties and classes are used in practice. This information can be used to recommend, e.g., the most common object property between two concepts. Such a search mechanism can also be very helpful to complement existing vocabularies by defining own subclasses or sub-properties. For example, the ontology engineer models an schema element `ex:MalePerson`, which represents a person, whose gender is male. If some context information was available that it is a "Person", the recommendation might suggest that `ex:MalePerson` might be a subclass of `foaf:Person`. LOVER can incorporate this information from a manual input of the ontology engineer or from previous iterations.

The LOVER approach is an iterative mechanism, meaning that the recommendation uses the context information from every iteration to recommend the most appropriate terms. One iteration is complete, if the user models a schema element by either reusing a term or specifying an own term. This way LOVER recommends but also adapts and updates the recommendations of classes and properties, even for data elements which have already been mapped to specific terms. Therefore, LOVER is most likely to achieve an optimized mapping according to the specification of an greedy algorithm. This means, to recommend an optimal term for a mapping is an optimization problem. A greedy algorithm utilizes a specific heuristic to provide the optimal choice at each iteration of a process. The heuristic comprises an update of the context information after each iteration and performs a further search for appropriate terms. To find a global optimum is a NP-complete problem, but the underlying heuristic may result in a local optimum, which approximates a reasonable solution. By this, it supports increasing the reuse of existing vocabularies and mixing a fair amount of different vocabularies with respect to the ontology engineering principles. In the first iteration, LOVER provides a rough draft of the model which is filled up with more and more details after each further iteration.

At the beginning of the data publishing process, the ontology engineer defines keywords, which determine domain and scope of the Linked Data dataset he intends to model and to publish. This is the first step of (7). Hereby, LOVER incorporates the specified information from the ontology engineer to use it as context in its first search for an appropriate vocabulary. For example, the ontology engineer intends to model a dataset containing data on people and their relationships to each other. This information can be used to allocate the first re-usable vocabularies, in this case FOAF as the data is about people. As next step, the ontology engineer enumerates all schema elements which describe the dataset. This is the second step regarding (7), as it is important to get the list of elements which are supposed to be mapped to classes and properties from existing vocabularies. LOVER performs a first keyword-based search to provide and recommend the ontology engineer terms from already published vocabularies for each enumerated schema element. This is support of type (1). As part of the recommendation, LOVER displays several meta information we have mention before on the vocabulary and its terms. This supports the engineer regarding the types (2), (3), and (4). This meta information is also used in accordance with the collected context information to rank the results of the term search. Hereby, terms from vocabularies which are already taken into account are ranked higher than terms from other vocabularies. This enables the ontology engineer to specify several mappings to one vocabulary, which results in a mixing of a reasonable amount of vocabularies. This supporting feature is executed with respect to the type (6). Furthermore, if the search involves several information sources, the ontology engineer is presented several result lists from which he may choose the preferred term. This semi-automatic support, which refers to type (8), enables the engineer to make a decision. To this end, the ontology engineer performs a first draft of a mapping. Then, he defines which schema elements are classes within his dataset, which is the further step of (7). LOVER uses this information to perform another search in order to recommend classes from established vocabularies for the schema elements the engineer has already marked as mapping classes. This supports the engineer with respect to type (3). As next step, he defines the hierarchy between the classes. This enables the ontology engineer to select a mapping for the parent class. If the subclass is not mappable to any existing term, LOVER is at least able to provide a recommendation to complement an existing vocabulary, which is a support with respect to type (5). Once the classes and their hierarchy has been modeled, the ontology engineer has to define the object properties, in order to refine their representation. This is again a further step of (7). As the engineer defines an object property between two specific classes, LOVER uses this information as input to search in the SchemEX index. It retrieves the possible object properties and recommends these to the ontology engineer. As last step, the ontology engineer has to define a datatype and annotation properties. LOVER then searches for such datatype and annotation properties of the already modeled class. In addition, it displays the best keyword-based search results from another information source. The ontology engineer chooses the terms according to (8), and finishes modeling his dataset. When having completed these several iterations, LOVER has made it possible to provide support for the ontology engineer for modeling a Linked Data set including all eight types of support stated above.

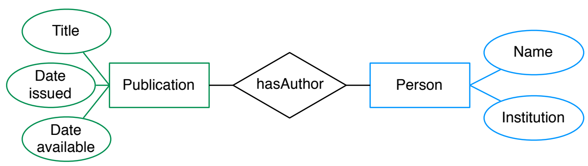

Figure 1: Database scheme from the scenario

While with this setup LOVER is in principle capable of providing all types of support the systems remains extensible to integrate further sources for vocabulary recommendation. One intended purpose is to aggregate information and recommendations obtained from various sources. Another purpose is to be capable to consider and make use of different types of context not covered so far.

5. MODELING AN EXAMPLE DATASET USING LOVER

To illustrate the LOVER approach and to provide a first preliminary and conceptual evaluation of it, we define the following scenario with real world data from the domain of social sciences: An ontology engineer intends to publish a subset of the data of the GESIS Social Science Open Access Repository (SSOAR)[19] as Linked Data. The Entity Relationship (ER) model in Figure 1 illustrates the schema of the dataset in our scenario. The goal is to model this schema as Linked Data according to the best practices, especially regarding the re-use and mix of existing and actively used vocabularies and their terms. The ontology engineer either uses terms from external vocabularies directly or defines his own vocabulary but links its terms to equal classes or properties from external vocabularies with `owl:equivalentClass` or `owl:equivalentProperty`. Again, while illustrating the modeling process using the LOVER approach, we explain where LOVER incorporates the types of support provided in 2.

Initially we determine the domain and the scope of the dataset, as it is the first step of (7). The domain is *social sciences* and the scope comprises the elements *Person* and *Documents*. LOVER uses this information for a first retrieval of some vocabularies the ontology engineer might be able to re-use. In our example this would be the Dublin Core and the FOAF ontology, as these are the primary results of searching for *Documents* and *Person*. As next step, the ontology engineer enumerates all schema elements within the dataset, i.e., *Publication*, *Title*, *Date issued*, *Date available*, *hasAuthor*, *Person*, *Name*, and *Institution*. LOVER performs a keyword-based search using the Swoogle API and provides the ontology engineer a set of terms for each schema element, which is a support with respect to (1). In our case we can assume that all provided terms are from FOAF and Dublin Core. Then, the recommendations include (2) meta information on FOAF and Dublin Core, (3) the semantic usage of each term within these two vocabularies, and (4) other meta information on each term, such as domain and range of properties and class hierarchies for classes. These information are provided by the vocabulary publishers and can be retrieved via the Swoogle API. The engineer chooses the terms `foaf:publication` for *Publication*, `dcterms:URI` for *URI*, `dcterms:title` for *Title*, `dcterms:issued` for *Date issued*, `dcterms:available` for *Date available*, `foaf:Person` for *Person*, and `foaf:name` for *Name*. This is by all means a semi-automatic mapping and thus is provided according to support type (8). Let us say that for *hasAuthor* and *Institution* LOVER was not able to suggest a useful recommendation. The engineer now enumerates the classes, i.e., *Person* and *Publication* like it is provided by (7). In this scenario, there is no need to define a class hierarchy. If the ontology engineer defined such a hierarchy and provided the parent class with a mapping, LOVER would support to complement the existing vocabulary according to (5). In our scenario, the enumerated classes are used by LOVER to perform another multimodal search for classes only to represent these schema elements. Again, LOVER provides support with respect to type (3). In the case of *Person* no changes occur, but in the case of *Publication* LOVER adapts and updates its recommendation to `swrc:Publication`, since this is the best fitting string-based result for a class. The ontology engineer realizes that this is a better semantic mapping than before and changes the mapping to `swrc:Publication`. Now, the SWRC ontology is integrated in the Linked Data dataset. Thus, LOVER performs a routine to examine if other schema elements can be mapped to terms from the SWRC ontology. This way, it provides support according to type (6). LOVER can yet again update a recommendation to achieve a better mapping. In our case, LOVER has found a suitable mapping for *hasAuthor* with `swrc:author` and `swrc:institution` for *Institution*. The ontology engineer chooses this mapping. The next step is to refine the object properties. The engineer specifies that the property `swrc:author` has the domain `swrc:Publication` and the range `foaf:Person`. LOVER uses this information to search the SchemEX index for the number of occurrences of such a `swrc:Publication swrc:author foaf:Person` triple. In this step, LOVER queries SchemEX several times leaving out in turn the domain, the range, and the object property, to search for the number of occurrences without restricting the the subject type, object type, and concrete property. This way, LOVER retrieves and presents further meta information on the actual usage of types/properties in the Linked Data cloud and provides support for the ontology engineer with respect to type (4). In example, LOVER finds a few occurrences of such a triple. But what if the object property is withdrawn from the search? By doing this, LOVER is able to find more triples including other object properties, such as `dcterms:creator` between the two classes. This triple has occurred way more often in the Linked Open Data cloud. The ontology engineer might want to change the object property to `dcterms:creator`, but both would be semantically correct according to (3). Finally the ontology engineer specifies that the properties `swrc:institution` and `foaf:name` are annotation properties of `faof:Person` and `dcterms:URI`, `dcterms:title`, `dcterms:issued`, and `dcterms:available` are annotation properties of `swrc:Publication`.

After applying LOVER, the resulting data scheme looks like as depicted in Figure 2. It is described in turtle format representing an example instance. Overall, the use of the LOVER approach has a positive effect on finding and applying appropriate vocabularies and terms for re-use.

[19]`http://www.ssoar.info/` accessed 2012/12/20

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
    ns#>.
@prefix swrc: <http://swrc.ontoware.org/ontology#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.

<http://ex1/001> rdf:type swrc:Publication;
        dc:title "Example␣Title";
        dc:issued "Example␣Issued␣Date";
        dc:available "Example␣Available␣Date";
        dc:creator <http://ex1/name/xyz>.

<http://ex1/name/xyz> a foaf:person;
        foaf:name "xyz";
        swrc:institution "Example␣Institution".
```

Figure 2: Resulting RDF for the scenario

6. CONCLUSION

In this paper, we have presented (a) the several types to support an ontology engineer in modeling a Linked Data dataset, (b) discussed the conformance of established systems and services with respect to these types, and (c) proposed LOVER as novel approach for incorporating eight different types of support to enable an ontology engineer to model Linked Data according to the best practices. We have illustrated how LOVER supports the ontology engineer to comply to the best practices such as re-use of established vocabularies and mixing a reasonable amount of different vocabularies. Its mechanism complies to the iterative approach of generating Linked Data starting from a rough draft and adding details afterwards, similar to the process of developing an ontology. In addition, it provides a multimodal search functionality, which approximates an optimal mapping of schema elements to terms from established vocabularies, and recommends these mappings to the engineer.

7. REFERENCES

[1] Alani, H., Noy, N. F., Shah, N., Shadbolt, N., and Musen, M. A. Searching ontologies based on content: experiments in the biomedical domain. In *K-CAP* (2007), D. H. Sleeman and K. Barker, Eds., ACM, pp. 55–62.

[2] Allemang, D., and Hendler, J. A. *Semantic Web for the Working Ontologist - Effective Modeling in RDFS and OWL, Second Edition*. Morgan Kaufmann, 2011.

[3] Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., and Aumueller, D. Triplify: Light-weight linked data publication from relational databases. In *WWW 2009* (2009), J. Quemada, G. León, Y. S. Maarek, and W. Nejdl, Eds., ACM, pp. 621–630.

[4] Basca, C., Corlosquet, S., Cyganiak, R., FernĞndez, S., and Schandl, T. Neologism: Easy vocabulary publishing. peer-reviewed.

[5] Bizer, C., and Cyganiak, R. D2RQ - lessons learned. *W3C Workshop on RDF Access to Relational Databases* (October 2007).

[6] Bizer, C., Cyganiak, R., and Heath, T. How to Publish Linked Data on the Web, July 2008.

[7] Cantador, I., Fernández, M., and Castells, P. Improving ontology recommendation and reuse in webcore by collaborative assessments. In *CKC* (2007), vol. 273, CEUR-WS.org.

[8] Ding, L., Finin, T. W., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V., and Sachs, J. Swoogle: a search and metadata engine for the semantic web. In *CIKM* (2004), pp. 652–659.

[9] Ding, L., Pan, R., Finin, T. W., Joshi, A., Peng, Y., and Kolari, P. Finding and ranking knowledge on the semantic web. In *ISWC* (2005), vol. 3729 of *LNCS*, Springer, pp. 156–170.

[10] Fernandez, M., Cantador, I., and Castells, P. Core: a tool for collaborative ontology reuse and evaluation. In *4th International Workshop on Evaluation of Ontologies for the Web (EON 2006)* (2006).

[11] Gómez-Pérez, A., Fernández-López, M., and Corcho-Garcia, O. *Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. 2003. ISBN: 1852335513.

[12] Gottron, T., Scherp, A., Krayer, B., and Peters, A. Get the google feeling: Supporting users in finding – relevant sources of linked open data at web-scale. In *Semantic Web Challenge, Submission to the Billion Triple Track* (2012).

[13] Heath, T., and Bizer, C. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.

[14] Hogan, A., Umbrich, J., Harth, A., Cyganiak, R., Polleres, A., and Decker, S. An empirical survey of linked data conformance. *Web Semantics: Science, Services and Agents on the World Wide Web 14* (2012), 14 – 44.

[15] Jain, P., Hitzler, P., Yeh, P. Z., Verma, K., and Sheth, A. P. A.p.: Linked data is merely more data. In *In: AAAI Spring Symposium 'Linked Data Meets Artificial Intelligence', AAAI* (2010), Press, pp. 82–86.

[16] Knoblock, C., Szekely, P., Ambite, J., Goel, A., Gupta, S., Lerman, K., Muslea, M., Taheriyan, M., and Mallick, P. Semi-automatically mapping structured sources into the semantic web. In *The Semantic Web: Research and Applications*, vol. 7295 of *Lecture Notes in Computer Science*. Springer Berlin/Heidelberg, 2012, pp. 375–390.

[17] Konrath, M., Gottron, T., Staab, S., and Scherp, A. Schemex - efficient construction of a data catalogue by stream-based indexing of linked data. *Web Semantics: Science, Services and Agents on the World Wide Web* (2012).

[18] Noy, N. F., and McGuinness, D. L. *Ontology development 101: A guide to creating your first ontology*. Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880, 2001.

[19] Romero, M. M., Vázquez-Naya, J. M., Munteanu, C. R., Pereira, J., and Pazos, A. An approach for the automatic recommendation of ontologies using collaborative knowledge. In *KES (2)* (2010), vol. 6277 of *LNCS*, Springer, pp. 74–81.

[20] Sahoo, S., Halb, W., Hellmann, S., Idehen, K., Thibodeau Jr, T., Auer, S., Sequeda, J., and

Ezzat, A. A survey of current approaches for mapping of relational databases to RDF, 01 2009.

[21] Scharffe, F., Atemezing, G., Troncy, R., Gandon, F., Villata, S., Bucher, B., Hamdi, F., Bihanic, L., Képéklian, G., Cotton, F., Euzenat, J., Fan, Z., Vandenbussche, P.-Y., and Vatant, B. Enabling linked-data publication with the datalift platform. In *AAAI 2012, 26th Conference on Artificial Intelligence, W10:Semantic Cities, July 22-26, 2012, Toronto, Canada* (07 2012).

Bisher erschienen

Arbeitsberichte aus dem Fachbereich Informatik

(http://www.uni-koblenz-landau.de/koblenz/fb4/publications/Reports/arbeitsberichte)

Thomas Gottron, Johann Schaible, Stefan Scheglmann, Ansgar Scherp, LOVER: Support for Modeling Data Using Linked Open Vocabularies, Arbeitsberichte aus dem Fachbereich Informatik 2/2013

Markus Bender, E-Hyper Tableaux with Distinct Objects Identifiers, Arbeitsberichte aus dem Fachbereich Informatik 1/2013

Kurt Lautenbach, Kerstin Susewind, Probability Propagation Nets and Duality, Arbeitsberichte aus dem Fachbereich Informatik 11/2012

Kurt Lautenbach, Kerstin Susewind, Applying Probability Propagation Nets, Arbeitsberichte aus dem Fachbereich Informatik 10/2012

Kurt Lautenbach, The Quaternality of Simulation: An Event/Non-Event Approach, Arbeitsberichte aus dem Fachbereich Informatik 9/2012

Horst Kutsch, Matthias Bertram, Harald F.O. von Kortzfleisch, Entwicklung eines Dienstleistungsproduktivitätsmodells (DLPMM) am Beispiel von B2b Software-Customizing, Fachbereich Informatik 8/2012

Rüdiger Grimm, Jean-Noël Colin, Virtual Goods + ODRL 2012, Arbeitsberichte aus dem Fachbereich Informatik 7/2012

Ansgar Scherp, Thomas Gottron, Malte Knauf, Stefan Scheglmann, Explicit and Implicit Schema Information on the Linked Open Data Cloud: Joined Forces or Antagonists? Arbeitsberichte aus dem Fachbereich Informatik 6/2012

Harald von Kortzfleisch, Ilias Mokanis, Dorothée Zerwas, Introducing Entrepreneurial Design Thinking, Arbeitsberichte aus dem Fachbereich Informatik 5/2012

Ansgar Scherp, Daniel Eißing, Carsten Saathoff, Integrating Multimedia Metadata Standarts and Metadata Formats with the Multimedia Metadata Ontology: Method and Examples, Arbeitsberichte aus dem Fachbereich Informatik 4/2012

Martin Surrey,Björn Lilge, Ludwig Paulsen, Marco Wolf, Markus Aldenhövel, Mike Reuthel, Roland Diehl, Integration von CRM-Systemen mit Kollaborations-Systemen am Beispiel von DocHouse und Lotus Quickr, Arbeitsberichte aus dem Fachbereich Informatik 3/2012

Martin Surrey, Roland Diehl, DOCHOUSE: Opportunity Management im Partnerkanal (IBM Lotus Quickr), Arbeitsberichte aus dem Fachbereich Informatik 2/2012

Mark Schneider, Ansgar Scherp, Comparing a Grid-based vs. List-based Approach for Faceted Search of Social Media Data on Mobile Devices, Arbeitsberichte aus dem Fachbereich Informatik 1/2012

Petra Schubert, Femi Adisa, Cloud Computing for Standard ERP Systems: Reference Framework and Research Agenda, Arbeitsberichte aus dem Fachbereich Informatik 16/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, Natalia A. Zenkova, Simulating social objects with an artificial network using a computer cluster, Arbeitsberichte aus dem Fachbereich Informatik 15/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, Simulating medical objects using an artificial network whose structure is based on adaptive resonance theory, Arbeitsberichte aus dem Fachbereich Informatik 14/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, Comparing the efficiency of serial and parallel algorithms for training artificial neural networks using computer clusters, Arbeitsberichte aus dem Fachbereich Informatik, 13/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, A parallel algorithm for selecting activation functions of an artificial network, Arbeitsberichte aus dem Fachbereich Informatik 12/2011

Katharina Bräunlich, Rüdiger Grimm, Andreas Kasten, Sven Vowé, Nico Jahn, Der neue Personalausweis zur Authentifizierung von Wählern bei Onlinewahlen, Arbeitsberichte aus dem Fachbereich Informatik 11/2011

Daniel Eißing, Ansgar Scherp, Steffen Staab, Formal Integration of Individual Knowledge Work and Organizational Knowledge Work with the Core Ontology *strukt*, Arbeitsberichte aus dem Fachbereich Informatik 10/2011

Bernhard Reinert, Martin Schumann, Stefan Müller, Combined Non-Linear Pose Estimation from Points and Lines, Arbeitsberichte aus dem Fachbereich Informatik 9/2011

Tina Walber, Ansgar Scherp, Steffen Staab, Towards the Understanding of Image Semantics by Gaze-based Tag-to-Region Assignments, Arbeitsberichte aus dem Fachbereich Informatik 8/2011

Alexander Kleinen, Ansgar Scherp, Steffen Staab, Mobile Facets – Faceted Search and Exploration of Open Social Media Data on a Touchscreen Mobile Phone, Arbeitsberichte aus dem Fachbereich Informatik 7/2011

Anna Lantsberg, Klaus G. Troitzsch, Towards A Methodology of Developing Models of E-Service Quality Assessment in Healthcare, Arbeitsberichte aus dem Fachbereich Informatik 6/2011

Ansgar Scherp, Carsten Saathoff, Thomas Franz, Steffen Staab, Designing Core Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 5/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, The prediction of currency exchange rates using artificial neural networks, Arbeitsberichte aus dem Fachbereich Informatik 4/2011

Klaus G. Troitzsch, Anna Lantsberg, Requirements for Health Care Related Websites in Russia: Results from an Analysis of American, British and German Examples, Arbeitsberichte aus dem Fachbereich Informatik 3/2011

Klaus G. Troitzsch, Oleg Kryuchin, Alexander Arzamastsev, A universal simulator based on artificial neural networks for computer clusters, Arbeitsberichte aus dem Fachbereich Informatik 2/2011

Klaus G. Troitzsch, Natalia Zenkova, Alexander Arzamastsev, Development of a technology of designing intelligent information systems for the estimation of social objects, Arbeitsberichte aus dem Fachbereich Informatik 1/2011

Kurt Lautenbach, A Petri Net Approach for Propagating Probabilities and Mass Functions, Arbeitsberichte aus dem Fachbereich Informatik 13/2010

Claudia Schon, Linkless Normal Form for ALC Concepts, Arbeitsberichte aus dem Fachbereich Informatik 12/2010

Alexander Hug, Informatik hautnah erleben, Arbeitsberichte aus dem Fachbereich Informatik 11/2010

Marc Santos, Harald F.O. von Kortzfleisch, Shared Annotation Model – Ein Datenmodell für kollaborative Annotationen, Arbeitsberichte aus dem Fachbereich Informatik 10/2010

Gerd Gröner, Steffen Staab, Categorization and Recognition of Ontology Refactoring Pattern, Arbeitsberichte aus dem Fachbereich Informatik 9/2010

Daniel Eißing, Ansgar Scherp, Carsten Saathoff, Integration of Existing Multimedia Metadata Formats and Metadata Standards in the M3O, Arbeitsberichte aus dem Fachbereich Informatik 8/2010

Stefan Scheglmann, Ansgar Scherp, Steffen Staab, Model-driven Generation of APIs for OWL-based Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 7/2010

Daniel Schmeiß, Ansgar Scherp, Steffen Staab, Integrated Mobile Visualization and Interaction of Events and POIs, Arbeitsberichte aus dem Fachbereich Informatik 6/2010

Rüdiger Grimm, Daniel Pähler, E-Mail-Forensik – IP-Adressen und ihre Zuordnung zu Internet-Teilnehmern und ihren Standorten, Arbeitsberichte aus dem Fachbereich Informatik 5/2010

Christoph Ringelstein, Steffen Staab, PAPEL: Syntax and Semantics for Provenance-Aware Policy Definition, Arbeitsberichte aus dem Fachbereich Informatik 4/2010

Nadine Lindermann, Sylvia Valcárcel, Harald F.O. von Kortzfleisch, Ein Stufenmodell für kollaborative offene Innovationsprozesse in Netzwerken kleiner und mittlerer Unternehmen mit Web 2.0, Arbeitsberichte aus dem Fachbereich Informatik 3/2010

Maria Wimmer, Dagmar Lück-Schneider, Uwe Brinkhoff, Erich Schweighofer, Siegfried Kaiser, Andreas Wieber, Fachtagung Verwaltungsinformatik FTVI Fachtagung Rechtsinformatik FTRI 2010, Arbeitsberichte aus dem Fachbereich Informatik 2/2010

Max Braun, Ansgar Scherp, Steffen Staab, Collaborative Creation of Semantic Points of Interest as Linked Data on the Mobile Phone, Arbeitsberichte aus dem Fachbereich Informatik 1/2010

Marc Santos, Einsatz von „Shared In-situ Problem Solving“ Annotationen in kollaborativen Lern- und Arbeitsszenarien, Arbeitsberichte aus dem Fachbereich Informatik 20/2009

Carsten Saathoff, Ansgar Scherp, Unlocking the Semantics of Multimedia Presentations in the Web with the Multimedia Metadata Ontology, Arbeitsberichte aus dem Fachbereich Informatik 19/2009

Christoph Kahle, Mario Schaarschmidt, Harald F.O. von Kortzfleisch, Open Innovation: Kundenintegration am Beispiel von IPTV, Arbeitsberichte aus dem Fachbereich Informatik 18/2009

Dietrich Paulus, Lutz Priese, Peter Decker, Frank Schmitt, Pose-Tracking Forschungsbericht, Arbeitsberichte aus dem Fachbereich Informatik 17/2009

Andreas Fuhr, Tassilo Horn, Andreas Winter, Model-Driven Software Migration Extending SOMA, Arbeitsberichte aus dem Fachbereich Informatik 16/2009

Eckhard Großmann, Sascha Strauß, Tassilo Horn, Volker Riediger, Abbildung von grUML nach XSD soamig, Arbeitsberichte aus dem Fachbereich Informatik 15/2009

Kerstin Falkowski, Jürgen Ebert, The STOR Component System Interim Report, Arbeitsberichte aus dem Fachbereicht Informatik 14/2009

Sebastian Magnus, Markus Maron, An Empirical Study to Evaluate the Location of Advertisement Panels by Using a Mobile Marketing Tool, Arbeitsberichte aus dem Fachbereich Informatik 13/2009

Sebastian Magnus, Markus Maron, Konzept einer Public Key Infrastruktur in iCity, Arbeitsberichte aus dem Fachbereich Informatik 12/2009

Sebastian Magnus, Markus Maron, A Public Key Infrastructure in Ambient Information and Transaction Systems, Arbeitsberichte aus dem Fachbereich Informatik 11/2009

Ammar Mohammed, Ulrich Furbach, Multi-agent systems: Modeling and Virification using Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 10/2009

Andreas Sprotte, Performance Measurement auf der Basis von Kennzahlen aus betrieblichen Anwendungssystemen: Entwurf eines kennzahlengestützten Informationssystems für einen Logistikdienstleister, Arbeitsberichte aus dem Fachbereich Informatik 9/2009

Gwendolin Garbe, Tobias Hausen, Process Commodities: Entwicklung eines Reifegradmodells als Basis für Outsourcingentscheidungen, Arbeitsberichte aus dem Fachbereich Informatik 8/2009

Petra Schubert et. al., Open-Source-Software für das Enterprise Resource Planning, Arbeitsberichte aus dem Fachbereich Informatik 7/2009

Ammar Mohammed, Frieder Stolzenburg, Using Constraint Logic Programming for Modeling and Verifying Hierarchical Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 6/2009

Tobias Kippert, Anastasia Meletiadou, Rüdiger Grimm, Entwurf eines Common Criteria-Schutzprofils für Router zur Abwehr von Online-Überwachung, Arbeitsberichte aus dem Fachbereich Informatik 5/2009

Hannes Schwarz, Jürgen Ebert, Andreas Winter, Graph-based Traceability – A Comprehensive Approach. Arbeitsberichte aus dem Fachbereich Informatik 4/2009

Anastasia Meletiadou, Simone Müller, Rüdiger Grimm, Anforderungsanalyse für Risk-Management-Informationssysteme (RMIS), Arbeitsberichte aus dem Fachbereich Informatik 3/2009

Ansgar Scherp, Thomas Franz, Carsten Saathoff, Steffen Staab, A Model of Events based on a Foundational Ontology, Arbeitsberichte aus dem Fachbereich Informatik 2/2009

Frank Bohdanovicz, Harald Dickel, Christoph Steigner, Avoidance of Routing Loops, Arbeitsberichte aus dem Fachbereich Informatik 1/2009

Stefan Ameling, Stephan Wirth, Dietrich Paulus, Methods for Polyp Detection in Colonoscopy Videos: A Review, Arbeitsberichte aus dem Fachbereich Informatik 14/2008

Tassilo Horn, Jürgen Ebert, Ein Referenzschema für die Sprachen der IEC 61131-3, Arbeitsberichte aus dem Fachbereich Informatik 13/2008

Thomas Franz, Ansgar Scherp, Steffen Staab, Does a Semantic Web Facilitate Your Daily Tasks?, Arbeitsberichte aus dem Fachbereich Informatik 12/2008

Norbert Frick, Künftige Anfordeungen an ERP-Systeme: Deutsche Anbieter im Fokus, Arbeitsberichte aus dem Fachbereicht Informatik 11/2008

Jürgen Ebert, Rüdiger Grimm, Alexander Hug, Lehramtsbezogene Bachelor- und Masterstudiengänge im Fach Informatik an der Universität Koblenz-Landau, Campus Koblenz, Arbeitsberichte aus dem Fachbereich Informatik 10/2008

Mario Schaarschmidt, Harald von Kortzfleisch, Social Networking Platforms as Creativity Fostering Systems: Research Model and Exploratory Study, Arbeitsberichte aus dem Fachbereich Informatik 9/2008

Bernhard Schueler, Sergej Sizov, Steffen Staab, Querying for Meta Knowledge, Arbeitsberichte aus dem Fachbereich Informatik 8/2008

Stefan Stein, Entwicklung einer Architektur für komplexe kontextbezogene Dienste im mobilen Umfeld, Arbeitsberichte aus dem Fachbereich Informatik 7/2008

Matthias Bohnen, Lina Brühl, Sebastian Bzdak, RoboCup 2008 Mixed Reality League Team Description, Arbeitsberichte aus dem Fachbereich Informatik 6/2008

Bernhard Beckert, Reiner Hähnle, Tests and Proofs: Papers Presented at the Second International Conference, TAP 2008, Prato, Italy, April 2008, Arbeitsberichte aus dem Fachbereich Informatik 5/2008

Klaas Dellschaft, Steffen Staab, Unterstützung und Dokumentation kollaborativer Entwurfs- und Entscheidungsprozesse, Arbeitsberichte aus dem Fachbereich Informatik 4/2008

Rüdiger Grimm: IT-Sicherheitsmodelle, Arbeitsberichte aus dem Fachbereich Informatik 3/2008

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik 2/2008

Markus Maron, Kevin Read, Michael Schulze: CAMPUS NEWS – Artificial Intelligence Methods Combined for an Intelligent Information Network, Arbeitsberichte aus dem Fachbereich Informatik 1/2008

Lutz Priese,Frank Schmitt, Patrick Sturm, Haojun Wang: BMBF-Verbundprojekt 3D-RETISEG Abschlussbericht des Labors Bilderkennen der Universität Koblenz-Landau, Arbeitsberichte aus dem Fachbereich Informatik 26/2007

Stephan Philippi, Alexander Pinl: Proceedings 14. Workshop 20.-21. September 2007 Algorithmen und Werkzeuge für Petrinetze, Arbeitsberichte aus dem Fachbereich Informatik 25/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS – an Intelligent Bluetooth-based Mobile Information Network, Arbeitsberichte aus dem Fachbereich Informatik 24/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS - an Information Network for Pervasive Universities, Arbeitsberichte aus dem Fachbereich Informatik 23/2007

Lutz Priese: Finite Automata on Unranked and Unordered DAGs Extented Version, Arbeitsberichte aus dem Fachbereich Informatik 22/2007

Mario Schaarschmidt, Harald F.O. von Kortzfleisch: Modularität als alternative Technologie- und Innovationsstrategie, Arbeitsberichte aus dem Fachbereich Informatik 21/2007

Kurt Lautenbach, Alexander Pinl: Probability Propagation Nets, Arbeitsberichte aus dem Fachbereich Informatik 20/2007

Rüdiger Grimm, Farid Mehr, Anastasia Meletiadou, Daniel Pähler, Ilka Uerz: SOA-Security, Arbeitsberichte aus dem Fachbereich Informatik 19/2007

Christoph Wernhard: Tableaux Between Proving, Projection and Compilation, Arbeitsberichte aus dem Fachbereich Informatik 18/2007

Ulrich Furbach, Claudia Obermaier: Knowledge Compilation for Description Logics, Arbeitsberichte aus dem Fachbereich Informatik 17/2007

Fernando Silva Parreiras, Steffen Staab, Andreas Winter: TwoUse: Integrating UML Models and OWL Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 16/2007

Rüdiger Grimm, Anastasia Meletiadou: Rollenbasierte Zugriffskontrolle (RBAC) im Gesundheitswesen, Arbeitsberichte aud dem Fachbereich Informatik 15/2007

Ulrich Furbach, Jan Murray, Falk Schmidsberger, Frieder Stolzenburg: Hybrid Multiagent Systems with Timed Synchronization-Specification and Model Checking, Arbeitsberichte aus dem Fachbereich Informatik 14/2007

Björn Pelzer, Christoph Wernhard: System Description:“E-KRHyper“, Arbeitsberichte aus dem Fachbereich Informatik, 13/2007

Ulrich Furbach, Peter Baumgartner, Björn Pelzer: Hyper Tableaux with Equality, Arbeitsberichte aus dem Fachbereich Informatik, 12/2007

Ulrich Furbach, Markus Maron, Kevin Read: Location based Informationsystems, Arbeitsberichte aus dem Fachbereich Informatik, 11/2007

Philipp Schaer, Marco Thum: State-of-the-Art: Interaktion in erweiterten Realitäten, Arbeitsberichte aus dem Fachbereich Informatik, 10/2007

Ulrich Furbach, Claudia Obermaier: Applications of Automated Reasoning, Arbeitsberichte aus dem Fachbereich Informatik, 9/2007

Jürgen Ebert, Kerstin Falkowski: A First Proposal for an Overall Structure of an Enhanced Reality Framework, Arbeitsberichte aus dem Fachbereich Informatik, 8/2007

Lutz Priese, Frank Schmitt, Paul Lemke: Automatische See-Through Kalibrierung, Arbeitsberichte aus dem Fachbereich Informatik, 7/2007

Rüdiger Grimm, Robert Krimmer, Nils Meißner, Kai Reinhard, Melanie Volkamer, Marcel Weinand, Jörg Helbach: Security Requirements for Non-political Internet Voting, Arbeitsberichte aus dem Fachbereich Informatik, 6/2007

Daniel Bildhauer, Volker Riediger, Hannes Schwarz, Sascha Strauß, „grUML – Eine UML-basierte Modellierungssprache für T-Graphen“, Arbeitsberichte aus dem Fachbereich Informatik, 5/2007

Richard Arndt, Steffen Staab, Raphaël Troncy, Lynda Hardman: Adding Formal Semantics to MPEG-7: Designing a Well Founded Multimedia Ontology for the Web, Arbeitsberichte aus dem Fachbereich Informatik, 4/2007

Simon Schenk, Steffen Staab: Networked RDF Graphs, Arbeitsberichte aus dem Fachbereich Informatik, 3/2007

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik, 2/2007

Anastasia Meletiadou, J. Felix Hampe: Begriffsbestimmung und erwartete Trends im IT-Risk-Management, Arbeitsberichte aus dem Fachbereich Informatik, 1/2007

„Gelbe Reihe“
(http://www.uni-koblenz.de/fb4/publikationen/gelbereihe)

Lutz Priese: Some Examples of Semi-rational and Non-semi-rational DAG Languages. Extended Version, Fachberichte Informatik 3-2006

Kurt Lautenbach, Stephan Philippi, and Alexander Pinl: Bayesian Networks and Petri Nets, Fachberichte Informatik 2-2006

Rainer Gimnich and Andreas Winter: Workshop Software-Reengineering und Services, Fachberichte Informatik 1-2006

Kurt Lautenbach and Alexander Pinl: Probability Propagation in Petri Nets, Fachberichte Informatik 16-2005

Rainer Gimnich, Uwe Kaiser, and Andreas Winter: 2. Workshop "Reengineering Prozesse" – Software Migration, Fachberichte Informatik 15-2005

Jan Murray, Frieder Stolzenburg, and Toshiaki Arai: Hybrid State Machines with Timed Synchronization for Multi-Robot System Specification, Fachberichte Informatik 14-2005

Reinhold Letz: FTP 2005 – Fifth International Workshop on First-Order Theorem Proving, Fachberichte Informatik 13-2005

Bernhard Beckert: TABLEAUX 2005 – Position Papers and Tutorial Descriptions, Fachberichte Informatik 12-2005

Dietrich Paulus and Detlev Droege: Mixed-reality as a challenge to image understanding and artificial intelligence, Fachberichte Informatik 11-2005

Jürgen Sauer: 19. Workshop Planen, Scheduling und Konfigurieren / Entwerfen, Fachberichte Informatik 10-2005

Pascal Hitzler, Carsten Lutz, and Gerd Stumme: Foundational Aspects of Ontologies, Fachberichte Informatik 9-2005

Joachim Baumeister and Dietmar Seipel: Knowledge Engineering and Software Engineering, Fachberichte Informatik 8-2005

Benno Stein and Sven Meier zu Eißen: Proceedings of the Second International Workshop on Text-Based Information Retrieval, Fachberichte Informatik 7-2005

Andreas Winter and Jürgen Ebert: Metamodel-driven Service Interoperability, Fachberichte Informatik 6-2005

Joschka Boedecker, Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Masaaki Kikuchi, and Minoru Asada: Getting closer: How Simulation and Humanoid League can benefit from each other, Fachberichte Informatik 5-2005

Torsten Gipp and Jürgen Ebert: Web Engineering does profit from a Functional Approach, Fachberichte Informatik 4-2005

Oliver Obst, Anita Maas, and Joschka Boedecker: HTN Planning for Flexible Coordination Of Multiagent Team Behavior, Fachberichte Informatik 3-2005

Andreas von Hessling, Thomas Kleemann, and Alex Sinner: Semantic User Profiles and their Applications in a Mobile Environment, Fachberichte Informatik 2-2005

Heni Ben Amor and Achim Rettinger: Intelligent Exploration for Genetic Algorithms – Using Self-Organizing Maps in Evolutionary Computation, Fachberichte Informatik 1-2005