Categorisation of Semantic Web Applications

Michael Martin
University of Leipzig
Dept. Business Information Systems
Leipzig, Germany
martin@informatik.uni-leipzig.de

Sören Auer
University of Leipzig
Dept. Business Information Systems
Leipzig, Germany
auer@informatik.uni-leipzig.de

***Abstract*—The recent success of the Semantic Web in research, technology and standardisation communities has also resulted in a large variety of different standards, technologies and tools. This diversity and heterogeneity goes along with an increasing complexity in assessing, evaluating, selecting and combining different approaches for the development of Semantic Web Applications (SWA). With this work we aim at lowering the entrance barrier for the development and engineering of Semantic Web Applications by presenting a classification of SWAs according to the dimensions semantic technology depth, information flow direction, richness of knowledge representation, semantic integration and user involvement. This categorisation helps to establish and consolidate the conceptualisation with regard to the engineering of SWAs and facilitates the comparability of different SWAs. With its requirements and benefits, the categorisation of SWAs can also serve as a guideline for practitioners looking into the application of semantic technologies within their use cases. We give an overview over popular SWAs and present, with Vakantieland and LinkedGeoData, two semantic web applications with regard to the categorisation in detail.**

***Keywords*-Categorisation; Semantic Web; Web Applications;**

I. Introduction

Recently, we observed the Semantic Web and related technologies gaining traction. Oracle, for example, integrated support for semantic knowledge management into their database product [1], Google started to evaluate annotations [2] using Resource Description Framework attributes (RDFa) and the W3C has lately launched the second revision of the Web Ontology Language (OWL) standard [3].

The success of the Semantic Web in research, technology and standardisation communities has, however, also resulted in a large variety of different approaches, standards and techniques. For example, a variety of knowledge representation formalisms with different expressivity is available with RDF, RDF-Schema, and various OWL flavours; there exist different serialisations such as RDF/XML, N3, NTriple, RDFa, Trix; the semantic web technology space is complemented with a wealth of different reasoners, triples stores, rule processors, semantic web service infrastructures, various APIs, etc. This diversity and heterogeneity goes along with an increasing complexity in assessing, evaluating, selecting and combining different approaches. From a Web Engineering point of view, this diversity substantially enlarges the application space of semantic technologies, but at the same time complicates their application.

Compared to conventional Web Applications, Semantic Web Applications (SWA) employ a number of additional standards and technologies on the persistence, data interchange / transaction processing and user interface layers (cf. Table I). This work is based on defining a Web Application as a client-server software application, which uses the HTTP protocol for communication between client and server as well as user interface technologies, which common Web browsers are capable to process (i.e., often HTML, CSS and Javascript or to a lesser extend UI technologies such as SVG or proprietary equivalents such as Flash and Silverlight). Our definition of a Semantic Web Application extends the Web Application definition with the requirement of using some Semantic Web knowledge representation formalism at either one or multiple of the persistence, data interchange / transaction processing and user interface layers. Semantic Web knowledge representation formalisms are mostly based on the RDF data model and include standards such as RDF-Schema, OWL, RIF or RDFa. The use of semantic technologies has a great potential in particular for the adaptability of Web applications, the efficient and standardized syndication of structured information or for improved search within and across different SWAs.

With this work we aim at *lowering the entrance barrier* for the development and engineering of SWAs by presenting a classification of SWAs according to the dimensions semantic technology depth, information flow direction, richness of knowledge representation, semantic integration and user involvement. This categorisation helps to establish and consolidate the conceptualisation with regard to the engineering of SWAs and facilitates the comparability of different SWAs. With the description of requirements and benefits for each of the different characteristics, the categorisation of SWAs can also serve as a guideline for practitioners looking into the application of semantic technologies within their use cases.

The paper is structured as follows: We describe our categorisation model along a number of dimensions in Section II. We present an overview of popular Semantic Web Applications in the light of these categorisations together with an in-depth description of two particular Semantic Web

Table I
JUXTAPOSITION OF CONVENTIONAL AND SEMANTIC WEB APPLICATION TECHNOLOGIES.

	Web Application	**Semantic Web Application**
Persistence Layer	Relational Database, ODBC,SQL	Triple Store, ODBC, SPARQL
Data Interchange & Transaction processing	REST-APIs, Web Services	SPARQL & LinkedData endpoint, Semantic Web Services
User Interface	(X)HTML, CSS, JS	(X)HTML, CSS, JS, RDFa, GRDDL

applications in Section III. We conclude and present related as well as future work in the Sections IV and V.

II. CATEGORISATION OF SEMANTIC WEB APPLICATIONS

In this section we discuss a number of dimensions along which semantic web applications can be characterised. These dimensions are the depth of the application architecture to which semantic technologies are applied, the direction(s) of semantic information flows, the richness of semantic knowledge representations, the intensity of the semantic integration with other SWAs and representation formalisms as well as the degree of user involvement.

A. Semantic Technology Depth

This categorisation dimension aims to capture to which degree the architecture of an SWA makes use of semantic technologies. Generally, SWAs can use semantic technologies in two different ways – externally and/or internally:

Extrinsic SWA: make use of semantic knowledge representation formalisms on the surface of the application in order to facilitate the interaction and integration with other SWAs and technologies. Implementation-wise, extrinsic SWAs are easy to realise, since conventional Web application development technologies and design patterns can be used. In order to map between internal persistence data models and semantic web taxonomies, vocabularies and ontologies, a number of tools exist [4]. Of particular importance are relational database schema, since their use is widespread, not only with Web applications. A comprehensive overview on approaches and technologies for transforming relational data to RDF is contained in [5]. Recently, the Linked Data paradigm has attracted quite some attention for exchanging and integrating data over the Web. Based on a relational to RDF mapping, Web applications can be easily equipped with a linked data interface (cf. e.g., [6]). Another popular approach to equip Web applications with a Semantic Web interface is RDFa standard [7] (sometimes also subsumed under Linked Data), which defines how conventional HTML can be annotated with RDF.

Intrinsic SWA: make direct internal use of semantic representations for their original application architecture. Here the situation is more complicated than with solely extrinsic SWA, since conventional technologies have to be complemented or replaced by their Semantic Web equivalents. On the persistence layer relational databases have to be replaced by triple stores. On the API layer Object-Relational-Mapping (ORM) techniques have to be replaced by corresponding APIs, which provide higher-level functions for handling RDF, RDF-Schema and OWL. In particular RDF data management, i.e., the querying performance of triple stores, is a decisive factor for the intrinsic use of semantic technologies in SWA (cf. e.g., [8], [9]). In recent years much progress has been made to improve the performance of triple stores by developing better storage, indexing and query optimisation. However, compared to querying data stored in a fixed relational database schema, querying a triple store is still usually slower by a factor of 5-50 (cf. e.g., BSBM results [10]). This shortcoming is due to the fact that columns in a relational database are typed and may be indexed more efficiently. By using a triple store, this efficiency is lost to the flexibility of amending and reorganising schema structures easily and quickly.

B. Information Flow Direction

The class of extrinsic SWAs can be further refined into SWAs, which produce, consume or produce and consume semantic representations.

Producing SWA: Based on either an intrinsic semantic information representation or on a mapping of other data models to RDF (as discussed in the previous section), four different types of Semantic Web interfaces can be distinguished:

- ETL-style dumping of information in RDF,
- provisioning of Linked Data, RDFa or GRDDL interfaces,
- declarative querying e.g., by means of SPARQL endpoints,
- Semantic Web Services or REST-style APIs, which return structured information adhering to the RDF data model.

The provisioning of semantically represented information in one of these forms helps to distribute and syndicate structured content. In particular, the re-usability and repurposability of information is facilitated. Compared to REST APIs and Web Services returning information in proprietary formats, these interfaces provide standardized means for accessing structured information. In order to build mashups, which combine information for various sources, Web developers would (when enabled to use one of these SWA interfaces) not be required to get acquainted with with various APIs and result formats. However, only REST APIs and Web Services are suited for transaction processing.

Consuming SWA: Information published as RDF is reusable by SWAs. If an SWA accesses information from the Data Web to enrich there own information space, it is classified as a Consuming SWA. A Consuming SWA can obtain information from either one or multiple of the methods used for publishing structured information used by producing SWAs. In most cases it will be sufficient for a consuming SWA to retrieve information via the HTTP protocol and parse one or multiple of the result formats RDF serializations, RDFa or SPARQL result formats. If producing SWAs offer RDF serialized according to the JSON specification [11], even specific parsing is not required, since JSON parsers are part of the standard functionality of most programming languages.

C. Richness of Knowledge Representation

SWAs can be further classified according to their use of rich knowledge representation formalisms:

- *Shallow KR SWA.* Comprise SWA, which e.g., primarily use taxonomies, simple hierarchies and relatively simple knowledge representation formalisms such as RDF and RDF-Schema.
- *Strong KR SWA.* Comprise SWA, which use higher level knowledge representation formalism such as different OWL variants, rules etc.

A navigator for the expressivity and complexity of description logics is also available [12]. Already the declarative querying of knowledge bases by means of SPARQL currently adds a substantial performance overhead to SWAs compared to relational database backed Web applications without even considering implicit information, which is must be revealed by reasoning. This is why we do not expect comprehensive description logic reasoning to be part of standard SWAs in the short to medium term. Instead there might be some light inferencing, which can be performed (on demand or in certain intervalls) by executing inference rules directly within triple stores (e.g., for resolving co-references, inverse relationships and computing transitive closures).

D. Semantic Integration

This categorisation dimension measures how well an SWA is integrated within the Semantic Web. The integration can be measured on the schema and instance level. On the schema level, for example, the number of overall schema elements (i.e., RDF/OWL classes and properties) can be put in relation to the number of reused schema elements, i.e., schema elements, which are either defined elsewhere or for which a `owl:sameAs` relation with an external element is defined.

Similarly, we can measure the semantic integration on the instance level. Semantic integration on the schema level appears to be slightly more important, than instance level integration, since in most cases there are more SWA, which publish information of a certain type (e.g., about Cities), than SWA, which publish information about a certain entity (e.g., Vienna).

For the integration and reuse on the schema level the availability of suitable upper level ontologies is important. For the semantic integration on the instance level interlinking hubs or crystallization ontologies such as DBpedia [13] are crucial. Depending on the level of semantic integration, we call representatives integrated (respectively isolated):

- *Isolated SWA* are categorized by a limited reuse of shared identifiers, vocabularies and ontologies.
- *Integrated SWA* are categorized by a strong reuse of shared identifiers, vocabularies and ontologies.

E. User Involvement

Another important characteristic of SWAs is the degree of end-user involvement. End-users can be roughly classified into spontaneous contributors, advanced users and knowledge engineering experts. Subsequently, an SWA can be categorized according to the sizes and ratios in which these different end-user groups are participating in the creation of semantic knowledge representations within an SWA. Also, it can be made clear which of these groups are restricted to contributions on the instance level and which participate in refining the knowledge schema. Other facets of the user involvement, which are not specific to SWAs are for example: the degree of closed user group, free for all, edit functionality for all information or just parts of the content.

F. Requirements and Benefits of characterization dimensions

We give an overview of the requirements and benefits of the presented categorisation dimensions for the implementation of SWAs in Table II. Based on the categorization dimensions different classes of SWAs can be distinguished:

- **Search engine / crawler.** Semantic search engines / crawler are extrinsic SWAs with a consuming information flow direction and mostly a shallow semantic richness. If such SWAs also process and republish retrieved RDF information, they can be considered as semantically integrated.
- **Collaborative knowledge acquisition.** Representatives of this class of SWAs are usually tailored towards a certain knowledge domain, although generic applications such as Semantic Wikis falling into that category exist. SWAs in this class are community oriented, mostly extrinsic and intrinsic, have a producing information flow direction and are often semantically integrated.
- **Visualization oriented**: SWAs of this visualization oriented class heavily use own or extrinsic retrieval and publish the received information with regards to a certain usage scenario and environment. Such SWAs have a consuming information flow direction and are semantically integrated.

Table II
SWA CHARACTERISATION OVERVIEW INDICATING REQUIREMENTS AND BENEFITS.

Dimension	Requirements	Benefits
Semantic technology depth		
Extrinsic	mapping between internal information structures and RDF	standardised interaction
Intrinsic	sufficient query processing power	increased schema flexibility
Information flow direction		
Consuming	mapping of RDF to internal information structures	wealth of additional structured information
Producing	mapping of internal information structures to RDF	increased information distribution
Semantic richness		
Shallow	availability of structured information	pay-as-you go strategy
Strong	comprehensive knowledge engineering	automated reasoning
Semantic integration		
Isolated	creation of own vocabularies and ontologies	simplified information governance
Integrated	vocabulary and identifier reuse on schema and/or instance level, co-reference and matching techniques	simplified syndication of semantic content
User involvement		
Com.-oriented	provisioning of simple interaction with semantic content	exploitation of crowd intelligence

- **Information Chaining**: SWAs of this class give users the possibility to get connected information from different distributed information spaces. In this case SWAs are extrinsic and have a consuming as well as a producing information flow direction. Furthermore, they are also intrinsic and semantically integrated, because they mostly store and process the received information.

III. CATEGORISATION EXAMPLES

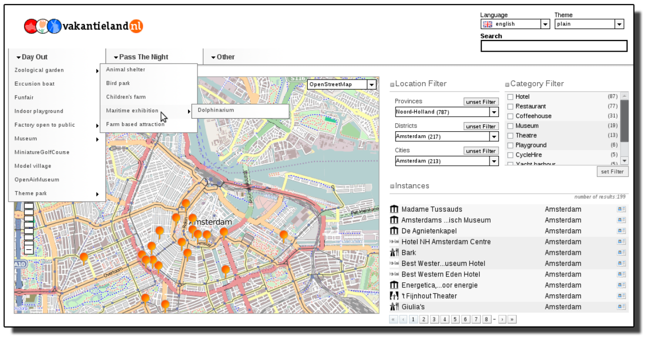
In this section we present an overview of existing SWAs according to the categorisation dimensions.

The selected SWAs are representatives of the exising SWA landscape, whose categorisations are presented in Table III. Some of the presented SWAs in this table, such as OntoWiki and Semantic Media Wiki, can not be categorised unambiguously. These SWAs are used to handle information of different domains in ways that the used vocabularies are defined elsewhere or created for the first time. However, if instances of such SWAs (i.e., the OntoWiki of the Leipzig Professors Catalogue [14] or OpenResearch [15] based on Semantic Media Wiki) will be investigated, it is possible to determine the correct classifications for the specific categorisation dimensions.

In the following we present two SWAs in more detail in order to explain the categorisation dimensions at an example.

A. Vakantieland

Vakantieland [16] publishes comprehensive information about 20,000 touristic points-of-interest (POI) in the Netherlands such as textual descriptions, location information and tourism features. The information is stored in a knowledge base containing almost 2 million triples. The Vakantieland data is structured using approximately 1,250 properties as well as 400 classes, which are used among others to provide different search and filter functionalities. As illustrated in Figure 1 it is possible to select a set of tourism classes which can be combined with other filter criteria such as terms from the free-text search as well as elements of the spatial hierarchy.



Language: english
Theme: plain
OpenStreetMap

Provinces: Noord-Holland (787)
Districts: Amsterdam (217)
Cities: Amsterdam (213)

□ Hotel	(87)
□ Restaurant	(77)
□ Coffeehouse	(31)
□ Museum	(19)
□ Theatre	(13)
□ Playground	(6)
□ CycleHire	(5)
□ Yacht harbour	(5)

number of results:199

Madame Tussauds	Amsterdam
Amsterdams ...isch Museum	Amsterdam
De Agnietenkapel	Amsterdam
Hotel NH Amsterdam Centre	Amsterdam
Bark	Amsterdam
Best Wester...useum Hotel	Amsterdam
Best Western Eden Hotel	Amsterdam
Energetica,...oor energie	Amsterdam
't Fijnhout Theater	Amsterdam
Giulia's	Amsterdam

Figure 1. The Vakantieland Semantic Web Application.

The depicted map acts also as an interactive map-bounding-box filter. According to the search and filter criteria a set of POIs is then being presented. Every POI description of such a result set can also be visited on a separate details page, consisting of properties arranged in a property hierarchy.

- **Semantic technology depth:** Vakantieland is an intrinsic *and* extrinsic SWA, since it employs the RDF data model for internal representation of information. Its implementation is based on the *Erfurt* Semantic Web API. With regard to publication, Vakantieland provides a Linked Data interface, which includes RDFa.
- **Information flow direction:** The POIs presented in Vakantieland were stored formerly in a relational database. While redesigning this application as an SWA, the data was converted to RDF and stored

Table III
EXAMPLES OF SWAS CATEGORISED ALONG THE CATEGORISATION DIMENSIONS.

Application	**STD Extrinsic**	**STD Intrinsic**	**IFD Consuming**	**IFD Producing**	**SR Shallow**	**SR Strong**	**SI Isolated**	**SI Integrated**	**UI Com. Oriented**
Collaborative Knowledge Aquisition									
OntoWiki (http://www.ontowiki.net)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Revyu (http://www.revyu.com/)	✓	✓	-	✓	✓	-	-	✓	✓
Semantic Media Wiki (http://www.semantic-mediawiki.org/)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Information Chaining									
Deri Pipes (http://pipes.deri.org/)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Freebase (http://www.freebase.com/)	✓	✓	✓	✓	✓	-	-	✓	✓
Twine (http://www.twine.com/)	✓	✓	-	✓	-	✓	-	✓	✓
Search Engines									
Bing reference search (http://www.bing.com/reference)	✓	✓	✓	-	✓	-	-	✓	-
Geonames (http://www.geonames.org/)	✓	✓	n/a	✓	✓	n/a	-	✓	-
Google Squarred (http://www.google.com/squared)	✓	n/a	✓	-	✓	-	-	✓	-
Sig.ma (http://sig.ma/)	✓	✓	✓	-	✓	-	-	✓	-
Sindice (http://www.sindice.org/)	✓	✓	✓	-	✓	-	-	✓	-
Swotti (http://www.swotti.com/)	✓	n/a	✓	-	n/a	n/a	-	✓	✓
Visualization Oriented									
DBpedia Mobile (http://wiki.dbpedia.org/DBpediaMobile)	✓	-	✓	-	-	✓	-	✓	-
Facetted Wikipedia Search (http://dbpedia.neofonie.de/browse/)	✓	-	✓	-	-	✓	-	✓	-
RelFinder (http://relfinder.dbpedia.org/)	✓	-	✓	-	-	✓	-	✓	-

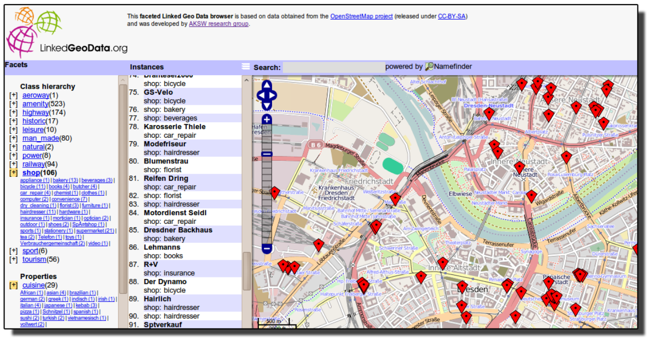
in a Triple-Store (OpenLink Virtuoso). In addition to publish the information for end-users with HTML/CSS/JS, the information is also provided as RDF (RDF/XML, Turtle, N3, JSON), which demonstrates that Vakantieland is a *producing* SWA. Except the geo-coordinates, which are retrieved from different geocoding services, Vakantieland does not consume RDF data from other SPARQL or LinkedData endpoints at this time.

- **Semantic richness:** With regard to the expressivity of the used knowledge representation techniques, Vakantieland is rather constrained and mostly in the RDF and RDF-Schema space. The used OWL features are confined to class and property definitions. In this case the semantic richness of this information space can be categorised as *shallow*.
- **Semantic integration:** The semantic integration is medium. On the schema level Vakantieland reuses vocabularies such as DublinCore [17], WGS84 [18] and GoodRelations [19], but also defines a large number of own schema elements, such as tourism classes, tourism object features, tourism offerings as well as different address, geospatial and contact properties. In future Vakantieland will become a fully integrated SWA since it is planned to link instances with DBpedia resources.
- **User involvement:** Vakantieland is a moderated tourism Wiki. At the moment, it is possible to edit fulltext-descriptions, address and contact information, which already helps to decrease costs for maintaining and to increase the quality of presented information. Only predefinied properties are editable by end users. An appropriate moderation process will be included to prevent publication of inappropriate material. Vakantieland is *community oriented* but not as much as other Semantic Wikis.

B. LinkedGeoData

LinkedGeoData is an effort to add a spatial dimension to the Web of Data. LinkedGeoData uses the information collected by the OpenStreetMap project and makes it available as an RDF knowledge base according to the Linked Data principles. It interlinks this data with other knowledge bases in the Linking Open Data initiative. The benefits of revealing the structured information in OSM are accessible in a faceted based browser [20] as depicted in Figure 2.

This user interface allows to browse the world by using a slippy map. Once a region is selected, the browser analyses the descriptions of nodes and ways in that region and generates facets for filtering. Once a facet or a specific facet value has been selected, matching elements are displayed as markers on the map and in a list. If the selected region is changed, these are updated accordingly. If a user logs into the application by using her OSM credentials, the displayed elements can directly be edited in the map view. For this, the browser generates a dynamic form based on existing properties. The form also allows to add arbitrary additional



LinkedGeoData.org

This **faceted Linked Geo Data browser** is based on data obtained from the OpenStreetMap project (released under CC-BY-SA) and was developed by AKSW research group.

Facets | Instances | Search: | powered by Namefinder

Class hierarchy
[+] aeroway(1)
[+] amenity(523)
[+] highway(174)
[+] historic(17)
[+] leisure(10)
[+] man_made(80)
[+] natural(2)
[+] power(8)
[+] railway(94)
[+] **shop(106)**
appliance (1) | bakery (13) | beverages (3) | bicycle (11) | books (4) | butcher (4) | car_repair (4) | chemist (1) | clothes (1) | computer (2) | convenience (7) | dry_cleaning (1) | florist (3) | furniture (1) | hairdresser (11) | hardware (1) | insurance (1) | mortician (1) | optician (2) | outdoor (1) | shoes (2) | SpÃ¤tshop (1) | sports (1) | stationery (1) | supermarket (21) | tea (2) | Telefon (1) | toys (1) | Verbrauchergemeinschaft (2) | video (1) |
[+] sport(6)
[+] tourism(56)

Properties
[+] cuisine(29)
African (1) | asian (4) | brazilian (1) | german (2) | greek (1) | indisch (1) | irish (1) | italian (4) | japanese (1) | kebab (3) | pizza (1) | Schnitzel (1) | spanish (1) | sushi (2) | turkish (2) | vietnamesisch (1) | vollwert (2) |

74. **Drahtesel2000**
shop: bicycle
75. **GS-Velo**
shop: bicycle
76. shop: bakery
77. shop: beverages
78. **Karosserie Thiele**
shop: car_repair
79. **Modefriseur**
shop: hairdresser
80. **Blumenstrau**
shop: florist
81. **Reifen Dring**
shop: car_repair
82. shop: florist
83. shop: hairdresser
84. **Motordienst Seidl**
shop: car_repair
85. **Dresdner Backhaus**
shop: bakery
86. **Lehmanns**
shop: books
87. **R+V**
shop: insurance
88. **Der Dynamo**
shop: bicycle
89. **Hairlich**
shop: hairdresser
90. shop: hairdresser
91. **Sptverkauf**

Figure 2. The LinkedGeoData browser Semantic Web Application.

properties. In order to encourage reuse of both properties and property values, the editor performs a type-ahead search for existing properties and property values and ranks them according to the usage frequency. When changes are made, these are stored locally and propagated to the main OSM database by using the OSM API.

- **Semantic technology depth:** The LinkedGeoData browser uses a a data model in its persistence layer, which is close to the RDF data model, but at the same time also more tailored towards the specific requirements (e.g., handling of large volumes of semantically annotated geospatial data). Hence, the LGD browser represents some hybrid type with regard to the semantic technology depth. Since the LGD browser also offers LinkedData and SPARQL interfaces it can, however, be characterized to be extrinsic.
- **Information flow direction:** The LGD browser is primarily a producing SWA. However, it also draws substantially from OpenStreetMaps data (which uses a relational representation).
- **Semantic richness:** The LGD knowledge bases use very shallow KR formalisms, mostly RDF and RDF-Schema. Ontology reasoning is not feasible regarding the size of LGD (with more than 3 billion triples).
- **Semantic integration:** The semantic integration of LGD is still rather low, since most of the data (e.g., streets, buildings, areas etc.) and schema elements (taxonomies of spatial objects and categorisations) in LGD are still relatively unique on the Data Web. However, LGD uses a few vocabulary elements (e.g., from the W3Cs WGS vocabulary) and is interlinked with DBpedia.
- **User involvement:** LGD itself has a relatively small and rather passive user community. However, it substantially draws from the vast OpenStreetMaps community, which is also the reason, why the KR formalisms are rather shallow.

IV. Related Work

Other than for the engineering and development of Web Applications (e.g., [21], [22], few approaches specifically tailored for the engineering Semantic Web applications exist. The *Semantic Web Framework* (SWF), for example, is a component-based framework for rapidly analysing required components, the dependencies between them, and selecting existing solutions [23]. A characterization of large scale semantic applications is presented in [24]. Based on this characterization, a guideline for the specification and design of large scale semantic applications was developed. Other than the work presented in this paper, the characterization and guidelines focus on large semantic applications in general and are not specifically tailored towards smaller SWAs. Another approach tackling the design and development of Semantic Web Application based on existing standards was published in [25]. This work represents a framework for engineering SWAs, that spans over several enterprises by applying techniques, methodologies, and notations offered by software engineering, Web engineering, and Business Process modelling. Existing Web Engineering processes are about design, implementation and maintenance of Web Applications, but lack the generation of meta-data. The "Web Engineering for Semantic Web Applications" (WEESA) approach [26] particularly tackles this aspect.

V. Conclusions and Future Work

While the applicability of semantic technologies was substantially broadened by the growth of Semantic Web standards, tools and approaches, the engineering complexity of SWAs substantially increased. With this work we aimed to contribute, to establish and to consolidate the conceptualisation of SWAs and facilitate the comparability of different SWAs. One of the intentions of using formal knowledge representation techniques (such as ontologies) is the decoupling of data and the application and the transition to flexible interfaces between both. However, a complete separation between information structures and application logic will not be completely possible. Hence, it is paramount to outline methodologies for the co-design of SWAs and knowledge bases. In the next paragraphs we outline the from our point of view most pressing hurdles for the wide-spread adoption of SWAs.

Closing the performance gap between relational and RDF data management: It has been widely acknowledged that the querying performance of triple stores is a decisive factor for the large-scale deployment of semantic technologies in many usage scenarios (cf. e.g., [8], [9]). In recent years much progress has been made to improve the performance of triple stores by developing better storage, indexing and query optimization. However, compared to querying data stored in a fixed relational database schema, querying a triple store is still usually slower by a factor of 2-20 (cf. e.g., BSBM results in [10]). This shortcoming is due to the fact that columns in a relational database are typed and may be indexed more efficiently. By using a triple store, this efficiency is lost to the flexibility of amending and reorganizing schema structures easily and quickly. A circumstance currently not yet taken advantage of by triple stores is that in typical application scenarios only relatively small parts of a knowledge base change within a short period of time. Based on this observation SPARQL result caching and view materialization strategies can be developed, which accelerate access to frequently used information structures.

Authoring of semantic-rich content: The overwhelming success of the World Wide Web was to a large extend based on the ability of ordinary users to author content easily. In order to publish content on the WWW, users had to do little more than to annotate text files with few, easy-to-learn HTML tags. Unfortunately, on the semantic data web the situation is slightly more complicated. Users do not only have to learn a new syntax (such as N3, RDF/XML or RDFa), but also have to get acquainted with the RDF data model, ontology languages (such as RDF-S, OWL) and a growing collection of connected RDF vocabularies for different use cases, such as FOAF, SKOS and SIOC. Previously, many approaches were developed to ease the syntax side of semantic authoring [27], [28]. In order to enable ordinary users to author rich semantic representations easily, user interfaces of SWAs have to also hide the data model from ordinary users without giving up the flexibility of mixing and mashing different, evolving vocabularies.

REFERENCES

[1] X. Lopez and S. Das, "Oracle database 11g semantic technologies, semantic data integration for the enterprise, white paper," Oracle Semantic Technologies Center, Tech. Rep., 2009.

[2] "Introducing rich snippets," http://googlewebmastercentral.blogspot.com/2009/05/introducing-rich-snippets.html.

[3] M. Schneider, "Owl 2 web ontology language rdf-based semantics," W3C Rec., Tech. Rep., 2009.

[4] "Converter To Rdf," http://esw.w3.org/topic/ConverterToRdf.

[5] S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau, S. Auer, J. Sequeda, and A. Ezzat, "A Survey of Current Approaches for Mapping of Relational Databases to RDF," 2009.

[6] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller, "Triplify: light-weight linked data publication from relational databases." in *WWW2009*. ACM, pp. 621–630.

[7] B. Adida, M. Birbeck, S. McCarron, and S. Pemberton, "RDFa in XHTML: Syntax and Processing," W3C, Rec., 2008.

[8] M. Schmidt, T. Hornung, G. Lausen, and C. Pinkel, "Sp2bench: A sparql performance benchmark," in *ICDE*. IEEE, 2009.

[9] C. Bizer and A. Schultz, "The Berlin SPARQL Benchmark," *International Journal On Semantic Web and Information Systems*, 2009.

[10] "Berlin SPARQL Benchmark," http://www4.wiwiss.fu-berlin.de/bizer/BerlinSPARQLBenchmark/.

[11] "RDF JSON Specification," http://n2.talis.com/wiki/RDF_JSON_Specification.

[12] "Complexity of reasoning in Description Logics," http://www.cs.man.ac.uk/~ezolin/dl/.

[13] J. Lehmann, C. Bizer, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia - a crystallization point for the web of data," *Journal of Web Semantics*, vol. 7, no. 3, pp. 154–165, 2009.

[14] T. Riechert, U. Morgenstern, S. Auer, S. Tramp, and M. Martin, "Knowledge engineering for historians on the example of the catalogus professorum lipsiensis," in *ISWC2010*, ser. LNCS. Shanghai / China: Springer, 2010.

[15] "Open Research," http://www.openresearch.org/.

[16] "Vakantieland Prototypes," http://staging.vakantieland.nl/.

[17] "The Dublin Core Metadata Initiative," http://dublincore.org/.

[18] "Basic Geo (WGS84 lat/long) Vocabulary," http://www.w3.org/2003/01/geo/.

[19] "GoodRelations - The Web Ontology for E-Commerce," http://purl.org/goodrelations/.

[20] "LinkedGeoData Browser," http://browser.linkedgeodata.org/.

[21] L. S. Al-Salem and A. Abu Samaha, "Eliciting web application requirements - an industrial case study," *J. Syst. Softw.*, vol. 80, pp. 294–313, 2007.

[22] F. Daniel and M. Matera, "Turning web applications into mashup components: Issues, models, and solutions. ICWE2009," ser. LNCS, vol. 5648. Springer, pp. 45–60.

[23] R. García-Castro, A. Gómez-Pérez, O. M. noz García, and L. Nixon, "Towards a component-based framework for developing semantic web applications," 2008, pp. 197–211.

[24] O. M. noz García and R. García-Castro, "Guidelines for the specification and design of large-scale semantic applications. ASWC2009," 2009, pp. 184–198.

[25] M. Brambilla, I. Celino, S. Ceri, D. Cerizza, E. D. Valle, and F. M. Facca, "A software engineering approach to design and development of semantic web service applications. ISWC2006," ser. LNCS, vol. 4273, pp. 172–186.

[26] G. Reif, H. Gall, and M. Jazayeri, "Weesa: Web engineering for semantic web applications. WWW2005." ACM, pp. 722–729.

[27] T. Tudorache, N. F. Noy, S. Tu, and M. A. Musen, "Supporting Collaborative Ontology Development in Protégé," in *ISWC2008*, ser. LNCS, vol. 5318. Springer, 2008.

[28] S. Auer, S. Dietzold, and T. Riechert, "OntoWiki – A Tool for Social, Semantic Collaboration," in *ISWC2006*, ser. LNCS, vol. 4273. Springer, 2006.