Injecting semantic annotations into (geospatial) Web service descriptions

Patrick Maué, Henry Michels and Marcell Roth

Institute for Geoinformatics (ifgi)
University of Münster, Germany
Weseler Str. 253
48151 Münster
Germany e-mail: firstname.lastname@uni-muenster.de

Abstract. Coupling existing geospatial applications with Semantic Web technologies requires solutions to also semantically enable underlying information sources. Ontologies help to formally specify the semantics of underlying data models and Web service capabilities. Spatial data complies with well-established standards to support seamless integration into applications ranging from commercial Geographic Information Systems (GIS) to open source web mapping clients. To ensure compatibility with existing solutions, the semantic enablement of spatial data sources has to reflect both worlds. Semantic annotations close this gap by linking in between the existing non-semantic Web service metadata and their semantic counterparts. The open source Semantic Annotations Proxy (SAPR) is light-weight RESTful API deployed as free service which “injects” semantic annotations into existing Web service descriptions without breaking the standards. The presented approach decouples the annotations from the original metadata. This ensures the separation of concerns between data providers and end users with different and sometimes conflicting views on annotations. The presented approach is focusing on W3C- and OGC-compliant Web services, but can be theoretically applied on any kind of information source with structured metadata.

Keywords: Semantic Annotation, Semantic Web, Tools, Geospatial Semantic Web, Semantic Web Services

1. Introduction

Efficient discovery of content in the Web relies on search engines with sophisticated indexing and scoring techniques. Finding relevant records in a search engine’s index relies on metrics to compute the match between the query and the record. Contemporary solutions depend on text analysis for indexing and scoring, and are very efficient for text-based content such as web sites.

These techniques can not by simply applied on structured data without textual content (e.g. geospatial data), or any kind of binary data (e.g. pictures, videos, or audio files). Finding such content relies on annotations [1] which associate descriptive metadata with the resource [2]. The metadata is then again used for common indexing and scoring techniques. Semantic annotations link to formally specified vocabularies capturing the content’s meaning. These could be, for example, ontologies defined with the help of the Web Ontology Language (OWL), or taxonomies based on the Simple Knowledge Organization System (SKOS).

Besides the traditional information retrieval techniques based on indexing and string matching, semantic annotations linking to ontologies support logic-based reasoning. In the case of finding information, reasoning engines such as PELLET [3] can then precisely match the semantic queries to the semantically annotated content, and ensure better rates of recall [4].

We understand the semantic annotation as a reference which establishes a Link Annotation [2] between the application-specific meta-data and a shared external vocabulary. The semantic annotation links to one vocabulary term (i.e. the owl:Class in OWL or the skos:Concept in SKOS) which is related to the annotated item in the meta-data. For example, some spatial
Injecting Semantic Annotations

Data representing street features is encoded using the XML-based Geography Markup Language (GML). The data is structured according to its XML schema describing the feature types, which is part of the Web service metadata. The feature type 42001 (the official identifier for street data from German mapping agencies) is linked to the class Street, which is part of a globally shared ontology accessible on the Web. Such link annotations to ontology concepts support (a) non-domain experts to better understand what the data represents and (b) reasoning on the linked ontologies, e.g. to retrieve spatial dataset about streets also for search requests looking for transportation infrastructures.

Semantic annotation techniques exist for unstructured content such as Web sites (e.g. using Microformats) and media formats (e.g. photos). In this paper we focus on annotations for Web services. They serve data or processes through well-defined interfaces described in the Web service metadata. The latter is semantically annotated to automatically integrate the Web services into semantically enabled applications.

Research on semantic annotations for Web documents [5,6], multimedia [1,7], geospatial data [4,8,9,10], and Web services [11,12,13,14] has been around for several years. The description of Semantic Web Services (SWS) eventually led to the standard “Semantic Annotations for Web Service Description Language (SA-WSDL)” [15]. It specifies extension points W3C-compliant Web service metadata encoded with the Web Service Description Language (WSDL). SA-WSDL also includes recommendations to semantically annotate XML schema (which is embedded in the WSDL documents). This focus on WSDL does unfortunately impair its applicability for some scenarios. Although the W3C standards are widespread and well-established, domain-specific standardization organizations have come up with their own solutions. Geospatial Data, for example, is commonly served through Web services compliant to the standards of the Open Geospatial Consortium (OGC)1. These standards define their own approach for XML-based metadata compliant to well-established XML-schema. Depending on the type of the spatial resource (i.e. features for vector-based data, coverages for raster based data, and geoprocessing methods), different OGC Web service types exist. Well-established standards are the OGC Web Mapping Service (WMS) for the visualization, the OGC Web Feature Service (WFS), and the OGC Web Coverage Service (WCS) 2.

In [10] we discuss potential extension points for OGC standards which can be used for link annotations. Extending XML-based meta-data compliant to an existing standard has to rely on such existing extension points. Simply applying SA-WSDL in this context would invalidate the XML documents and break support by generic spatial clients such as Desktop GIS systems of web-mapping applications.

In this paper we are going to present our implementation of the Semantic Annotations API (sapience) and the Semantic Annotations Proxy (SAPR)3. Sapience comprises libraries which can be used to extract, store, and inject semantic annotations within Web service meta-data without breaking the underlying standards. SAPR is a service built upon sapience to provide its functionality following a Software-as-a-service approach. It is a conceptually simple proxy-based solution for injecting semantic annotations. The “injection”-procedure refers to the semantic enrichment of XML-based documents by writing link annotations directly into the data stream. We explain how the proxy-based solution for the semantic enablement of existing Web service descriptions ensures a clear Separation of Concern (SoC).

The proposed solution for loosely-coupled metadata (separating between client-specific annotations and the provider-specific metadata) supports multiple annotations for one service description. This makes different application scenarios feasible, without relying on the service provider for specifying the annotations. The introduced implementation as well as the discussion of SoC for web service metadata should be considered as main contribution of this paper.

The proxy-based solution has initially been a pragmatic answer to the problem of integrating already existing Web services into semantically enabled applications.

---

1The reason why OGC has not (yet) adapted W3C standards is simple: the OGC standards have been developed several years before SOAP/WSDL emerged.

2See http://www.opengeospatial.org/standards for the complete list of OGC standards.

3See http://purl.org/net/sapience/docs for the access to the source code, issue tracking, and documentation.
abled applications. In the last decade, large-scale spatial data infrastructures (SDI) have been deployed by public mapping agencies. A subsequent semantic annotation is a challenging and resource-intensive endeavor. Initiatives like the European SISE (Single Information Space in Europe) or legal directives such as INSPIRE (Infrastructure for Spatial Information in Europe) or the U.S. NSDI (National Spatial Data Infrastructure) call for a Web-based provision of spatial data from the public sector. But the migration from local installations to Web-enabled infrastructures has been (and still is) a tedious and cost-intensive task. That service providers will update and re-deploy existing services to include aspects such as semantic annotations cannot be expected. SAPR enables client application developers in need for semantically enriched Web services to simply annotate Web service in question and register it to the proxy. The original Web service is not modified and the outcome remains standards compliant. The only modification from a client’s perspective is the change of the URL representing the Web service location.

The following Section 2 introduces two application scenarios which illustrate the benefit of injecting annotations. A more in depth discussion about the need for injecting references is following in Section 3. The implementation of SAPR is introduced in Section 4, followed by an evaluation in Section 5. Section 6 lists related work about Semantic Web Services. The paper concludes with an outlook and a summary in Section 7.

2. Application Scenarios

Annotations support better interpretation and evaluation of the referenced data. Using a proxy ensures that different sets of annotations for applications deployed in different scenarios can be requested. In the following two sections we illustrate one typical application scenario for semantic annotations (linking to vocabularies describing the data’s relation to reality) and one for data quality annotations (linking to vocabularies describing aspects such as trust or uncertainty).

2.1. Annotations for capturing data semantics

The most prevalent application of annotations is semantic enrichment: the individual entities in a data model are linked to concepts in shared vocabularies to explain what the data represents in reality. Semantic annotations can be useful for a varying field of applications such as Web service discovery, automatic integration into existing business or scientific workflows, logic data integrity tests, semantic validation of Web service compositions, or inferring data visualization strategies. A more in depth discussion of these applications can be found in [10].

The linked vocabularies can be commonly accepted thesauri of one particular domain, shared domain ontologies developed for particular use cases, or local application-specific ontologies. Shared domain ontologies should be understood as formal specifications of reality, in particular of our geographic environment. A Web Feature Service [16] could, for example provide XML schema describing the data model of the feature type GEO150KType with the two attributes CODE and FORMATION (see Figure 1). These attributes labels are unfortunately cryptic, its meaning remains hidden to non-experts (or experts from different information communities). Requesting the actual data returns “2” for CODE and “Plio-Pléistocène” for FORMATION, which does not reveal any meaning as well. This example is taken from one of the scenarios of the European research project ENVISION. It adequately reflects the current situation: today’s use of (geospatial) Web services is significantly impaired by the lack of meaningful metadata [17].

GIS users directly load and visualize feature data from such a WFS. In this sense, the data is usable, but far from being useful. Without semantic annotations, potential clients have no means to find out what these attributes represent. And without a precise knowledge about the underlying data models, the data itself can be hardly used for critical geospatial tasks like decision making or risk modelling. Linking a feature type and its attributes to appropriate well-defined terms defined in globally shared vocabularies can help to avoid such semantic conflicts. In Figure 1, the attributes have an additional attribute sawsd1: modelReference="...". This model reference is the link annotation which is not part of the original schema, but has been injected after-

4The service can be found at: http://envision.brgm-rec.fr/Data_Geology.aspx
Injecting Semantic Annotations

wards. Semantic-enabled clients follow these links pointing to classes in RDF-based ontologies. On this level, descriptive metadata and meaningful (and multilingual) labels exist which are needed for the interpretation of the data models. More importantly, reasoning with common engines such as Pellet for OWL-DL[3] ease integration. They help to select appropriate visualization strategies (e.g., features representing water bodies are commonly colored blue) or detect (and potentially avoid) semantic conflicts in geospatial workflows.

We distinguish between local and global semantic annotations. The former links to service-specific ontologies which re-model the data. The latter (also coined DomainReferences) directly link to common vocabularies. This distinction has been proven useful to capture the sometimes complex functional dependencies in between attributes of data models [9]. In the given example, CODE identifies the geological formation (i.e. all features representing geological layers formed in the pleistocene era have the code “2”). Built-in ontology constructs such as constraints on properties in OWL help to re-model this inner relationships of the data. To make these local ontologies useful, they have to be aligned to globally shared ontologies. The domain references directly link to such domain ontologies. The FORMATION is for example directly linked to (made-up) domain concept GeologicEra.

2.2. Annotations for describing data quality

Communicating data quality is crucial for the evaluation of the usefulness of the data for the client’s intended application. Typical data quality aspects covered in the following are data provenance and uncertainty.

Geospatial data is the result of a measurement procedure (or sensor observation), and each measurement inhibits some sort of error. The original input for the data representing geologic layers, for example, comes from core samples. A three-dimensional interpolation algorithm estimates the distribution of the layers according to these samples. This deviation from truth is commonly called the Uncertainty of geographic information. Applications in need for high accuracy of certain properties, e.g. urban planning, should be aware of imprecise data. Otherwise, the error is hidden in the result (and high precision is simulated), leading eventually to wrong calculations or more serious issues. UncertML [18] has been developed as extension for geospatial data models to formalize the uncertainty. The link annotations then point to the appropriate definitions in the UncertML dictionary5. For many applications such information is not necessary (e.g. for simple navigational tasks), and many clients won’t be able to process the uncertainty information. With the help of the proxy, this information about the uncertainty can simply injected during runtime, and only compatible clients can request this data if needed.

Specifying provenance of (geospatial) data helps to build confidence (or trust) into the data. Such metadata includes, amongst others, detailed information about the publishing organization and a description of the data lineage. The former could be as simple as a link to a publicly shared FOAF profile. The client has to be able to learn about the data creator’s reputation to infer its usefulness for critical applications. Data lineage refers to the actual process for creating the data. Here, link annotations point to external (but application-specific) documents containing detailed information about the creation process. If links to FOAF profiles or Data Provenance documents are needed has to be decided by the client’s application. The presented approach clearly separates between the original provider-specific metadata and application-specific annotations, making it possible to support semantically-enabled applications and uncertainty-aware applications simultaneously.

3. Separation of Concerns for Semantic Annotations

Semantic query processing with annotations linking to shared ontologies can return more precise discovery results. Annotations pointing to information about uncertainty or trust can help to better communicate certain aspects of the data. But selecting an appropriate strategy to benefit from annotations requires that applications understand the annotation’s intended purpose. Bechhofer et al. [2] state that “we must be explicit about the assumptions that we make and the

5Which can be reached here: http://dictionary.uncertml.org/
context within such annotations should be interpreted". They distinguish between Decoration, Linking, Instance Identification and Instance Reference. Aboutness and Pertinence as typical annotation types. Marshall [5] draws a line between formal/informal and explicit/tacit annotations. The presented approach is restricted to Link Annotations pointing to formal (and explicit) meta-data.

The limitation on Link Annotations for the injection is based on the two following assumptions: (1) Link Annotations support loose coupling of the data models and the domain-specific meta-data. This enables separation of concerns and delegation. (2) Annotations are pointing to explicit specifications, either captured in ontologies or in shared vocabularies encoded in a well-known format (e.g. in the Resource Description Framework RDF).

3.1. Decoupling metadata

Separation of Concerns (SoC) refers to the idea of separating distinct features in software applications. Typical concerns are concurrency, persistence, or failure recovery [19]. In the case of metadata, the already mentioned information about data semantics, uncertainty, and trust could represent the different concerns. Link Annotations loosely couple implementation-specific data models with application-specific (or domain-specific) meta-data. The client application has to specify which annotations are to be injected.

SoC supports delegation of the annotation from the data provider to the domain experts. The data provider is an expert in his own domain (data acquisition and authoring), but he might lack skills in identifying appropriate ontologies for semantic annotations, or the correct formulas to describe the uncertainty inherent in the data. Furthermore, data providers usually have very specific end user in mind when creating the metadata. It requires domain experts who are able to close the information gap between the data providers and the potential users. In Library Research, this expert is the cataloger.

The cataloger must envisage the needs of the reader, endeavoring in every way to make it a simple process for him to find books. He should, like the librarian, adopt a neutral stand between the reader and his books, giving emphasis to what the author intended to describe rather than to his own views.

This quote [20] illustrates the expert’s dilemma. Library Research has always faced the problem of this information gap between potential readers and indexing catalogers due to differing backgrounds [21]. It is the librarian who is responsible for indexing a book for a library. She knows best how potential readers might be looking for it, and what search terms they might be using. The book’s author, on the other hand, might have had a very specific reader in mind; hence his description of book would be semantically narrow. Creating metadata for Web services is facing the similar issues.

The ad-hoc injection procedure enables decoupling of the semantic annotations from the source metadata defined by the data provider. Depending on the client’s request, different annotations are added to the metadata. Experts coming from different information communities are able to de-
Injecting Semantic Annotations

fine their own sets of annotations, without risk of conflicts with other annotations.

3.2. Semantics of Link Annotations

The categorization of books in libraries is following a well define scheme which has been adopted by the popular Dublin Core standard for resources on the Web. Capturing the semantics of data, and describing the inner relationships of data entities, is considerably more complex. The Link Annotation has been proposed as means to connect the individual elements in the data schema to the appropriate ontology concepts in shared vocabularies. But the link itself does not indicate the nature of the linked resource, or how the client application has to process the annotations. The resource may be just a different representation of the annotated item (the Instance Identification according to [2]). Or it provides information about it, e.g. how to use it (the Aboutness). The SAWSDL standard proposes the ModelReference for the Instance Identification of XML schema or WSDL elements [22]. Its purpose is the separation between two different encodings of the same entity. The standard also recommends to complement the reference with pointers to scripts which translate between the different encodings. In [10], the DomainReference extends the ModelReference and links between local implementation-specific data models and globally shared domain models.

Another option to let the client know about the nature of the linked resource is to link only to well-defined instances in a vocabulary. A link pointing to a resource modeled as instance of a Person from the FOAF vocabulary [23] can be used to infer trust information (if it is injected in the appropriate location). Links to SKOS concepts may be used to enhance discovery (e.g. browsing through categories). A similar approach can be taken for place names by pointing to entities served by gazetteers. Having all these different options could potentially result in a plethora of different kinds of Link Annotations with different identifiers. Avoiding this either requires an ontology of link annotations or committing to one commonly used method. In the case of the latter, only the ModelReference from the SAWSDL standard may be used. The referenced resource then has to be encoded in a way to let common reasoning engines infer its type. For example, instead of introducing a new reference for places (e.g. the http://.../PlaceReference), the ModelReference points to the OWL instance Paris, which is modeled as instance of the class City (which itself is a sub-class of Place).

4. The Semantic Annotations Proxy (SAPR)

The introduced proxy-based solution for injecting annotations into Web service descriptions has been implemented as a Web service itself. A REST-based interface supports simple integration into existing clients. The first URL in Figure 2 represents a typical description of a Web service using WSDL (this particular example is taken from the SAWSDL Testsuite⁶). Once registered to the proxy service, the semantically annotated description can be retrieved using the second URL. A list of registered Web services and a list of all references for one Web service can be requested using the URLs (3) and (4).

The injection procedure is illustrated in Figure 3. In step (1), the client software requests the Web service description using a URL from the proxy, the service identifier, and an open set of parameters. The second URL in Figure 2 is an example how to request a WSDL-based description of a Web service. The proxy resolves the given identifier to the URL of the original service (similar to URL shortening services) and opens a connection to the original Web service (2). The service identifier, coupled with the original parameters, forms the unique document identifier (docID), which is used for searching for annotations in the local annotations database. If no annotations are registered for the given docID, the stream coming from the open connection will be directly forwarded to the client. If annotations are registered, the XML stream is read line by line. For each line, a XPath [24] expression is generated and matched against the registered annotations. In the case of a match, the reference itself (which could be either attribute or a new XML element) is written into the output stream.

Any errors during this procedure, e.g. due to missing parameters or parsing problems, cause appropriate HTTP status errors. It is the client’s re-

⁶Available at: http://www.w3.org/2002/ws/sawSDL/CR/testsuite.html
Injecting Semantic Annotations

(1) http://www.w3.org/2002/ws/sawsdl/CR/wsdl2.0/00-plain.wsdl
(2) http://semantic-proxy.appspot.com/api/get?sid=6bbf18d0
(3) http://semantic-proxy.appspot.com/api/list/services
(4) http://semantic-proxy.appspot.com/api/list/references?sid=921e1da1

Fig. 2. URLs to access the Semantic Proxy

The current implementation supports semantic annotations for common Web services compliant to the W3C standards and described using the WSDL standard. In addition, the OGC standards for Web Processing Services (WPS), Sensor Observation Services (SOS), and Web Feature Services (WFS) have been configured. Support for new service types (or new extension points for semantic annotations) is added through XML-based configuration files. They contain a list of all possible locations of annotations in the various documents served by the standard. Simplified XPath expressions are used for the pattern definition. Only in the case of XML Schema or WSDL documents, the annotations are simple attributes which are added to existing elements. Annotating, for example, a process description coming from a WPS requires the injection of multiple lines of XML.

SAPR additionally comes with a RESTful API to interact (e.g. uploading, searching) with the registered services. The URLs (3), (4), and (5) in Figure 2 are examples. (3) can be used to list all services currently registered with SAPR. (4) is used to list extracted semantic annotations. The response, encoded in JSON, is a set of bindings between a location (the XPath-Expression) and a reference which has to be injected at this location (either an attribute or a whole chunk of XML). To register an annotated document, the file can be uploaded via the upload form available at (5). Clients are also able to directly register annotated metadata through the API.

5. Evaluation of the injection approach

The focus on the implementation is the dynamic injection of semantic annotations into structured metadata. It does neither contain an automatism how to create the annotations, nor do we add any functionality which makes of use of them (e.g. include reasoning on the semantic annotations). In the research project SWING [25], a visual interface has been implemented which supports the user in semantically annotating OGC Web Feature Services [26]. In the on-going research project Soa4All [27], clients for annotating WSDL files are developed. In the just recently started ENVISION project [28], SAPR is one core component for building a semantically enabled infrastructure for designing and publishing environmental models as Web services. The originally desktop-based semantic annotation interface implemented in SWING is currently migrated to the Web. It directly integrates with SAPR, making it possible for even non-ICT experts to semantically annotate Web service metadata.

In SWING, semantic annotations have been used for the semantic discovery. In the GDI-Grid project [29], we implemented an Eclipse-based plugin for the semantic validation of service compositions. The WSDL documents of the Web services in the workflow were semantically annotated using the SAWSDL standard. The semantic validation method extracted the semantic annotations and computed a pair-wise semantic match between the different Web services (checking if the output of
Injecting Semantic Annotations

one service is the valid input of the downstream service).

SAPR has been deployed as a Software-as-a-Service (SaaS) using Google App Engine. This approach comes with some limitations on the consumed resources, which have only a theoretical impact in the case of SAPR. The requirements for bandwidth, processing time, or storage are negligible for injecting references into meta-data streams. The benefits of cloud infrastructures such as scalability, availability, and performance outbalance the drawbacks caused by flexibility constraints. SAPR is a free Web service. In addition, it is free software and can be deployed in other locations as well. Working directly on the streams ensures very fast responses. The time difference between accessing the meta-data from the original source or the annotated document from SAPR is negligible, external factors, such as network latency or server response time, have a larger impact on responsiveness.

We have evaluated several aspects of the implementation. It has been thoroughly tested with module tests using appropriate test suites from the various standards (e.g. the already mentioned SAWSDL Test Suite). Issues such as scalability, performance, and sustainability were addressed already at the beginning. The streaming approach has been identified to address best the need for responsiveness. Deploying the service in the cloud ensures scalability and sustainability. The usefulness of the proxy-based injection of link annotations has been tested and shown in various research projects. A simple RESTful API for registering and retrieving annotations, as well as the focus on standard-compliance, ensures high usability and seamless integration into existing generic clients.

6. Related Work

Early work on semantic annotations was focused on the manual semantic annotation of Web content [6]. Semantic annotations for Web services, and the concept of Semantic Web Services (SWS), have been introduced by [31] and [12]. This work was primarily concerned about the semantically enabling W3C-compliant Web services, which eventually led to the development of the W3C recommendation for Semantic Annotations for WSDL (SA-WSDL) [22]. Enriching Geospatial Web services with ontologies has been investigated by [4] and [8].

The long-term vision of SWS is the automatic, reasoner-supported integration of Web services. Reasoning engines depend on specifications expressed in a logic-based language, e.g. the Web Ontology Language (OWL). Enabling semantics is therefore the task of either directly creating Web service descriptions in such a language, or by linking existing XML-based descriptions to these ontologies. The OWL-S Ontology (OWL for Services, [13] helps to (re-)model W3C-compliant Web services. The Web Service Modeling Ontology [32] (WSMO), and its recent descendant WSMO-Lite [15] are similar (but more sophisticated) solutions following the same approach. In this case, the capabilities of a Web services, as well as its information model, are completely covered by the ontologies. In the long run, semantically enabled Web services will directly deliver these ontologies (aligned to shared domain ontologies). XML-based metadata may then only be used to sustain backwards-compatibility.

Coupling Semantic Web technologies with often heterogenous SDIs has been subject of various research [9,33,8,17,4,25]. In [34], a semantic-enablement layer (SEL) is proposed to complement existing SDI components with Web services for ontology access and reasoning. SAPR can be considered as one important step towards SEL, since it provides the link between existing spatial data services and the SEL components.

7. Conclusion

The long-term vision of the Semantic Web assumes that there will be eventually a complete shift towards semantic-enabled Web resources. Until then, semantic annotations will act as bridge between the existing non-semantic resources and the more sophisticated logic-based approach. They are a short-term and non-intrusive solution to bring the benefits of semantics to existing infrastructures. By being aware of the well-established and mature standards, existing non-semantic clients won’t be locked out. The intro-

---

5 Available at http://code.google.com/appengine
duced SAWSDL standard [22] specifies semantic annotations for W3C-compliant Web services. In [10], solutions for semantic annotations which comply with the OGC Standards for Geospatial Web services are discussed.

The presented approach targets XML-based metadata of Web service descriptions. We have neither addressed the actual XML data nor other content types not encoded in XML. The former is simply an issue of configuring the proxy accordingly. The mentioned example of injecting details of data quality into the actual data is one target application of SAPR. Injecting annotations into non-XML based data was not covered, but is also required to integrate information hidden in raw data (e.g. sensor streams, satellite images, audio files). Here, the annotations could be injected as additional fields into existing metadata (e.g. the EXIF metadata for image files).

Dynamic injection relies on a reproducible way to identify the annotation’s location. Injecting annotations into unstructured data (i.e. data without a structure compliant to a schema) is difficult to achieve. Microformats and RDFa 8 are proposed markup formats for XHTML, and therefore means to annotate websites. The benefits of separating the annotations from the source data may also apply here, but are not realized in SAPR.

In this paper an implementation for a decoupled solution for semantic annotations of existing Web services has been presented. The Semantic Annotations Proxy (SAPR) has been developed in the context of multiple research projects dealing with semantic enablement of geospatial Web services. The implementation was conceptually designed to be non-intrusive, making it possible to stay compliant to the underlying standards for Web service descriptions. The stream-based injection approach, as well as the deployment in the cloud, ensured a very responsive system. SAPR is (or has been) used for the semantic annotation of W3C- and OGC compliant Web services. Future implementations will focus on the automatic translation between the semantically annotated Web service descriptions into ontology-based service models to ease the integration of existing Web services into Semantic Web applications.

8RDAa is a W3C recommendation for annotating websites. More information can be found here: http://www.w3.org/TR/rdfa-in-html/

8. Acknowledgments

The presented research has been funded by the BMF project GDI-Grid (BMBF 01IG07012, see http://www.gdi-grid.de) and the European research project ENVISION (FP7-249170, see http://www.envision-project.eu). Contributions by Carsten Kessler, Simon Scheider, and Mohammed Bishr have been of great help.

References


