Undefined 0 (0) 1
IOS Press


# Injecting semantic annotations into (geospatial) Web service descriptions

**Editor(s):** Thomas Lukasiewicz, University of Oxford, UK
**Solicited review(s):** Jacek Kopecký, The Open University, UK; Tudor Groza, The University of Queensland, Australia; Marinos Kavouras, National Technical University of Athens, Greece

Patrick Maué [a] Henry Michels [a] and Marcell Roth [a]
[a] *Institute for Geoinformatics (ifgi)*
*University of Münster, Germany*
*Weseler Str. 253*
*48151 Münster*
*Germany e-mail: firstname.lastname@uni-muenster.de*

**Abstract.** Geospatial Web services comply with well-established standards to support seamless integration into applications ranging from commercial Geographic Information Systems (GIS) to open source web mapping clients. Descriptions of service capabilities contain information about the provided data. Updates to the underlying database result in changing descriptions. To ensure compatibility with existing solutions, semantic enablement of Geospatial Web services has to reflect both, standards and changing metadata. Semantic annotations link between legacy non-semantic Web service descriptions and their semantic counterparts. The open source Semantic Annotations Proxy (SAPR) is a light-weight RESTful API deployed as free service which "injects" semantic annotations into existing Web service descriptions without breaking the standards. This approach decouples the annotations from the original metadata, which ensures the separation of concerns between data providers and end users with different and sometimes conflicting views on annotations. In addition, the service is robust regarding changes of the service descriptions. 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 relies on search engines with sophisticated indexing and scoring techniques. These techniques can not be simply applied on structured data without textual content such as geospatial data, or any kind of binary data, including pictures or videos. Finding such content relies on annotations which associate descriptive metadata with the resource [1]. The metadata is then again used for common indexing and scoring techniques. *Semantic* annotations link to formally specified vocabularies capturing the content's meaning. 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 [27] can then precisely match the semantic queries to the semantically annotated content, and ensure better precision of search results [15].

We understand semantic annotation as a reference which establishes a *Link Annotation* [1] between the application-specific metadata and a shared external vocabulary. They enable reasoning on the linked ontologies and support non-domain experts to better understand what the data represents. In this paper we focus on annotations for Web services which describe their interfaces with standardized metadata. The latter is semantically

annotated to integrate the Web services into semantically enabled applications.

Research on semantic annotations for Web documents [16,11], multimedia [29,3], geospatial data [15,12,19], and Web services [30,28,17] has been around for years. The description of Semantic Web Services (SWS) eventually led to the standard "Semantic Annotations for Web Service Description Language (SAWSDL)" [14]. It specifies extension points for W3C-compliant Web service metadata encoded with the Web Service Description Language (WSDL). Although the W3C standards are widespread and well-established, domain-specific standardization organizations have come up with their own solutions. Geospatial Data Services, for example, comply to the standards of the Open Geospatial Consortium (OGC)[1]. These standards define their own approach for XML-based metadata. 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.

In this paper we present the implementation of the Semantic Annotations Proxy (SAPR)[2]. It builds on (and is part of) the Sapience API[3], which comprises libraries used to extract, store, and inject semantic annotations within Web service metadata without breaking the underlying standards. SAPR exposes the Sapience functionality following a Software-as-a-Service approach. It is a conceptually simple *proxy-based* [11] solution for injecting semantic annotations. Similar to traditional HTTP proxies (e.g. for Web browsing), SAPR acts on behalf of the proxied Web service and either redirects client requests to the original Web service or updates the returned metadata. End users are not aware of its existence: the proxy takes the identity of the proxied service. 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 reflecting the standards and changing metadata, as well as the discussion of SoC for Web service annotations should be considered as main contribution of this paper.

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 combined with semantic-enabled applications 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 [20].

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

[2] SAPR is a free service, available at: `http://semantic-proxy.appspot.com`

[3] See `http://purl.org/net/sapience/docs` for the access to the source code, issue tracking, and documentation.

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. The domain ontologies should be understood as formal specifications of reality, in particular of our geographic environment. A Web Feature Service (WFS) [31] could, for example, provide XML schema describing the data model of the feature type `GEOL50KType` with the two attributes `CODE` and `FORMATION` (see Figure 1). These attribute 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[4] 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 [23].

GIS users directly load and visualize feature data from such a WFS. In this sense, the data is useable, but far from being useful. Client applications can be semantically enabled (i.e. supporting logic inference and visualization of the referenced knowledge). But without semantic annotations, these clients have no means to identify and retrieve the shared vocabularies to infer 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. In Figure 1, the attributes have an additional attribute `sawsdl:modelReference=".."`. This model reference is the link annotation which is not part of the original schema, but has been injected afterwards. 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[27] ease integration. They help to select appropriate visualization strategies (e.g. features representing water bodies are commonly coloured blue) or detect (and potentially avoid) semantic conflicts in geospatial workflows.

Semantic annotations linking to local application ontologies have been proven useful to capture the sometimes complex functional dependencies within data models [19]. 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 `FORMATION` could then, for example, be linked to a domain concept *GeologicEra*.

### *2.2. Annotations for describing data quality*

Communicating quality aspects of data is crucial for evaluating its usefulness for the intended application. Typical data quality metadata covered in the following are data provenance and uncertainty.

Geospatial data is the result of a measurement procedure (or sensor observation), and each measurement comes with some sort of error. The original input for the data representing geologic layers, for example, derives 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 [32] 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 dictionary[5]. Such information might not be necessary for some applications (i.e. simple navigation tasks). Their clients won't be able to process the updated uncertainty information. With the help of the proxy, this information about the uncertainty can simply injected during runtime, and

[4]The service can be found at: `http://envision.brgm-rec.fr/Data_Geology.aspx`

[5]Examples and the dictionary are available at: `http://dictionary.uncertml.org/`

```
<element type="GEOL50KType" substitutionGroup="gml:_Feature"/>
<complexType name="GEOL50KType">
 <complexContent>
  <sequence>
   <element name="CODE" type="string"
     sawsdl:modelReference="http://.../local/2zhe2#CODE"/>
   <element name="FORMATION" type="string"
     sawsdl:modelReference="http://.../Geology-Ontology#GeologicEra"/>
  </sequence>
 </complexContent>
</complexType>
```

Fig. 1. Semantic Annotations for XML (GML) schema

only (technically) 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. Clients might want to learn about the data provider'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. Which links to either FOAF profiles or Data Provenance documents are required has to be decided by the client application. The presented approach separates between 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

Legal directives such as INSPIRE (Infrastructure for Spatial Information in Europe)and the U.S. NSDI (National Spatial Data Infrastructure) call for a Web-based provision of spatial data from the public sector. In the last decade, large-scale spatial data infrastructures (SDI) have been deployed by public mapping agencies. 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 Web services to include aspects such as semantic annotations cannot be expected. The proxy-based solution of SAPR has initially been a pragmatic approach to integrate these legacy Web services into semantic-enabled applications. 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.

Selecting an appropriate strategy to benefit from annotations requires that applications understand the annotation's intended purpose. Bechhofer et al. [1] state that "we must be explicit about the assumptions that we make and the 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 [16] draws a line between formal/informal and explicit/tacit annotations. The presented approach is restricted to Link Annotations pointing to formal (and explicit) metadata.

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 metadata. 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 for-

mat (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 [9]. 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) metadata. The client application has to specify which annotations are to be injected. The presented proxy generates a unique identifier for each set of annotations, and expects this identifier as parameter to allocate the according annotations in the repository.

SoC supports *delegation* of the annotation from the data provider to the domain experts. Library Research has always faced the problem of the information gap between potential readers and indexing catalogers due to differing backgrounds [8]. 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 issue: the data provider is an expert in data acquisition and authoring, but she might lack skills in identifying appropriate ontologies for semantic annotations or the correct equations to describe the uncertainty inherent in the data[6]. Furthermore, data providers usually have very a specific end user in mind when creating the metadata. It requires domain experts to close the information gap between the data providers and the potential users. It is the domain expert who, on behalf of the data provider, identifies the appropriate shared vocabularies and creates the semantic annotations.

The ad-hoc injection procedure allows for decoupling semantic annotations from the original service metadata. Depending on the client request, different sets of annotations may be added to the metadata. Experts coming from different information communities are able to define their annotations, without risking conflicts with other annotations. The presented implementation does not support the domain expert in the actual semantic annotation; this is expected to happen before the service is registered to SAPR.

Clients for OGC Web services don't separate between the location of the Web service and its description. Descriptions are typically generated on the fly, since updates to underlying data sets are common. Each OGC Web service implements the GetCapabilities-operation providing the service metadata. Supported operations and service-specific information like feature types are listed in this document. W3C-compliant Web services, on the other hand, support separation of descriptions from services. Best practice may imply that WSDL descriptions are provided by the services (by concatenating the "?wsdl" to the URL). But this is not part of the standard; SoC is thus inherently supported by W3C Web services. However, updating such service would require to locate and update all WSDL descriptions linking to it. Hence, services typically have implementation and description at the same location, which again raises the issue of SoC (and makes the presented proxy useful for semantic 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 [1]). Or it provides information about it, e.g. how to use it (the Aboutness). The SAWSDL standard proposes the *ModelReference* to add semantic annotations into XML schema or WSDL elements [14].

[6] This discussion is focusing on the separation between the two roles of the data provider and the domain expert. One person (or organization) can still play both roles.

Its main purpose is the separation between two different encodings of the same entity (even though the standard itself is intentionally unconstrained). The standard also recommends to complement the reference with pointers to scripts which translate between the different encodings. In [20], we discuss the*DomainReference* as extension to the ModelReference to align local implementation-specific data models with globally shared domain models.

Another option to let the client know about the nature of the referenced resource is to link only to well-defined instances in a vocabulary. A link pointing to a resource modelled as instance of a Person from the FOAF vocabulary [5] 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 modelled as instance of the class `City` (which itself is a sub-class of `Place`).

## 4. The Semantic Annotations Proxy (SAPR)

The concept of a network proxy has been around for decades [26]. A proxy acts on behalf of the service it encapsulates by taking its identity. Clients can access the original service only through the proxy; the proxied service stays hidden behind the proxy. Clients are not aware of the existence of the proxy. Even though the service location changes (including the query part with the parameters), the client interacts with the proxy as with the original service . This approach is similar to Web browsers accessing the Web through a proxy. The semantic annotations proxy only acts on a requests for metadata which has been registered to the proxy. All other requests - if supported by the selected HTTP method - are redirected to the proxied service. The client then directly interacts with the original service to, for example, request the actual data. In all other cases (including invocation requests using HTTP Post, which doesn't support redirects) the proxy forwards the request to the services, and streams the result back.

The introduced proxy-based solution for injecting annotations into Web service descriptions has been implemented as a Web service itself. The first URL in Figure 2 represents a typical request for the capabilities description of an OGC Web service. Once registered to the proxy service, the semantically annotated description can be retrieved using the second URL by using the original request parameter (in this case "request=GetCapabilities") and the service id [7]. A list of registered Web services and a list of all references for one Web service could then be requested using the URLs (3) and (4).

### *4.1. Extracting the annotations*

The following scenario assumes that the document with the annotations listed in Figure 1 is uploaded to the proxy using the manual upload form (URL (5) in Figure 2). The XML stream is forwarded to the reference extractor, which first identifies the service type by scanning the XML header. Each service type is configured through an identification pattern (a regular expression) and the patterns for the reference locations as specified in the according metadata standard. These locations are path expressions based on the XPath [2] syntax. In this example, the metadata is provided by an OGC WFS (version 1.1.0). The according pattern for the header is listed as (1) in Figure 3. Each of the three given patterns (e.g. `.*Capabilities.*`) has to be part of the document's root element. For the WFS 1.1.0, the path expressions (2) and (3) in Figure 3 are configured. The first describes the location of a `modelReference` (as defined in the SAWSDL standard) as attribute of a `simpleType`-Element in XML schema. The second specifies the `MetadataURL` in the `FeatureType` element as valid

[7] These are examples; the service identifiers change continuously and the repository is purged often. Use URL (3) for a list of valid services.

```
(1) http://www.example.com/wfs?request=GetCapabilities&...
(2) http://semantic-proxy.appspot.com/api/get?request=GetCapabilities&...&sid=ae34aa09
(3) http://semantic-proxy.appspot.com/api/list/services
(4) http://semantic-proxy.appspot.com/api/list/references?sid=ae34aa09
(5) http://semantic-proxy.appspot.com/html/upload.html
```

Fig. 2. Example URLs to access the Semantic Proxy

destination. The nature of the reference is concatenated, and used to identify the location in the original metadata document: *child* indicates that the location of the discovered annotation should point to the parent element, i.e., the annotation should later be injected as child of the given location. In the case of *attribute* or *sibling* the full location is stored; the metadata is then either injected as attribute into the element located by the path expression, or directly after it as new element.

The extraction and injection procedures make use of the Streaming API for XML (STAX)[8]. The extractor computes the path, implemented as a stack of XML elements, for every element in the XML stream and computes the match with the configured patterns (also represented in stacks). The elements in the stacks are recursively compared. Namespaces of the individual elements are ignored. In the case of a match, the location is extracted and persistently stored together with the reference. In Figure 3, Pattern (4) represents such a location. In consists of the path expression describing the location within the XML tree and the location indicator (sibling, attribute, or child). This example states that the associated reference is injected after the element `Name` (since it is configured as sibling) with the text "Geol50k". The result of the extraction procedure is the service identifier, which is required to retrieve the annotations. Each registration results in a new service id. One document (with different sets of annotations) can therefore be uploaded multiple times without risking conflicts.

### *4.2. Injecting the annotations*

The injection procedure is illustrated in Figure 4. The client software requests annotated metadata using the proxy, a 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 first extracts the service identifier from the request and retrieves the annotations as well as the original service request from the cache (which is built initially from the data store on startup). If the service id is not registered, an exception is returned to the client. In the next step, the other request parameters (e.g. the *request*-parameter of an OGC service) are extracted and compared to the parameter in the original service request. If they don't match, the client receives a redirection response (HTTP Code 302) with the original location in the HTTP header. If they match, the service metadata is retrieved from its original source and forwarded to the injection engine. The XML stream from the source is read element by element (using again StAX). Similar to the extraction procedure, a path 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. The resulting stream is directly forwarded to the client.

Any errors during this procedure, e.g. due to missing parameters or parsing problems, cause appropriate HTTP status errors. It is the client's responsibility to communicate the error to the end user. Requests to the service which do not serve metadata (e.g. requesting the feature data from an OGC Web service) are redirected.

### *4.3. Reflecting change*

Changes in the original metadata (e.g. new feature types have been added) don't affect the injection procedure as long as the stored reference locations (the XPath statements) are still valid. The injection procedure works on the stream, and only stops if the XPath Stack at the current position in the stream matches a XPath expression stored with the annotations. New feature types, or

[8] We use Woodstox, a "high-performance validating namespace-aware StAX-compliant (JSR-173) Open Source XML-processor", see `http://woodstox.codehaus.org/`.

```
(1) ".*Capabilities.*",".*xmlns.*http://www.opengis.net/wfs.*",".*version=\"1.1.0\".*"
(2) //xsd:simpleType[@sawsdl:modelReference]/attribute()
(3) //wfs:FeatureTypeList/wfs:FeatureType/wfs:MetadataURL/sibling()
(4) //Capabilities/FeatureTypeList/FeatureType/Name/text()=\'Geol50k\'/sibling()
```

Fig. 3. Patterns for the extraction of annotations.

new operations in a WSDL file, are ignored; previously existing metadata will still be annotated. If, however, the location of the stored XPath element changes (e.g., if the feature type which is annotated is renamed, and the name attribute is in the path), the annotation has to be updated accordingly.

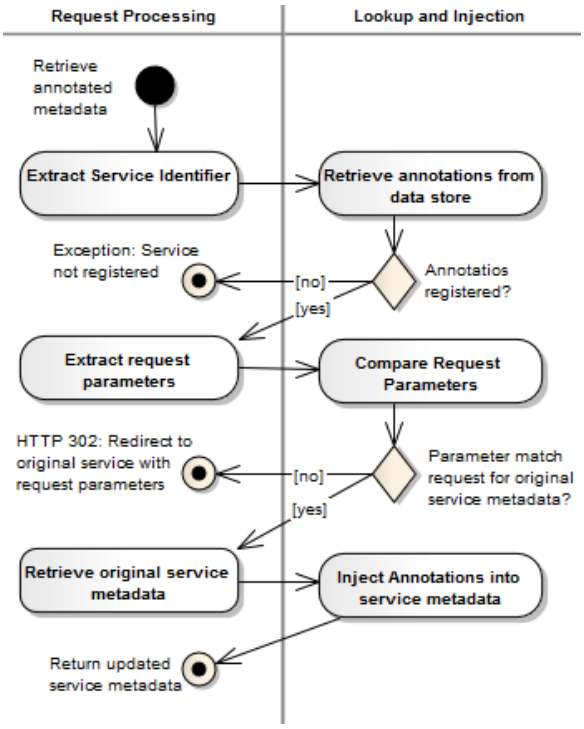


Fig. 4. The injection procedure

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. Annotations are usually simple attributes which are added to existing elements. But 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) lists extracted semantic annotations for one service. 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 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 [6]. The semantic annotations have been used in SWING for the semantic discovery. In the GDI-Grid project [21], 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. SAPR has been initially developed in GDI-Grid to support the semantic validation. In the ENVISION project [18], 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. Being a standard component for the semantic enablement of existing service infrastructures is the long-term vision for the Semantic Annotations Proxy.

SAPR has been deployed as a Software-as-a-Service (SaaS) using the Google App Engine[9]. 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 metadata streams. The benefits of cloud infrastructures such as scalability, availability, and performance [4] outbalance the drawbacks caused by flexibility constraints. Google App Engine manages load balancing by dynamically adding instances covering increased load. To further improve performance, the internal caching service of the App Engine is used for the retrieval of the annotations. SAPR is a free Web service. In addition, it is free software and can be deployed in other locations as well.

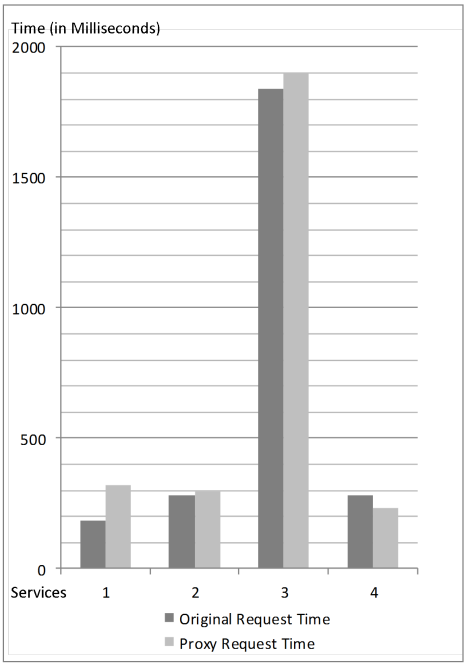


Fig. 5. Impact of the proxy on the response time.

[9] Available at `http://code.google.com/appengine`

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. Figure 5 depicts the average (50 runs each) response times (vertical axis) for requesting the original metadata or the injected metadata from the proxy (which includes the former). In particular the third (very slow) service illustrates that other factors have a greater impact on the response then the proxy. In the fourth case (requesting one file from the SAWSDL-Testsuite), the proxy request was consistently faster then the original. This is probably due to a more efficient implementation of HTTP request handling within the Google App Engine SDK (since the tests ran on a local setup, any benefits from the proxy running on the Google infrastructure are not reflected in this chart).

The required support of different service metadata standards has been thouroughly tested with module tests using appropriate test suites from the various standards (e.g. the already mentioned SAWSDL Test Suite). But the strict dependence on standards is also a problem: Web services with invalid service descriptions are common, and far too often we experienced unexpected errors during the XML parsing. In addition, the slow response times can be a limiting factor within the Google App Engine. The API cuts off requests exceeding ten seconds response time, which is not uncommon for Web services running not within large-scale infrastructures. SAPR has been implemented to be robust in respect to changes of the source metadata. Changing the metadata does not affect the injection procedure, as long as no elements are removed which are used within the extracted XPath expressions.

## 6. Related Work

Early work on semantic annotations was focused on the manual semantic annotation of Web content [11]. Semantic annotations for Web services, and the concept of Semantic Web Services (SWS), have been introduced by [22] and [28]. This work was primarily concerned about the semantically enabling W3C-compliant Web services, which even-

tually led to the development of the W3C recommendation for Semantic Annotations for WSDL (SAWSDL) [14]. Enriching Geospatial Web services with ontologies has been investigated by [15] and [12].

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, [17] helps to (re-)model W3C-compliant Web services. The Web Service Modelling Ontology [24] (WSMO), and its recent descendant WSMO-Lite [13] 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 heterogeneous Spatial Data Infrastructures (SDI) has been subject of various research [19,7,12,23, 15,25]. In [10], 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 vision of the Semantic Web relies on content which is either directly semantically represented (i.e., data encoded in an ontology language) or has been semantically annotated. Legacy infrastructures comply to well-established (and often domain-specific) standards. Here, semantic representations of the data would require updates to the standards and accordingly the running infrastructures. Injecting the semantic annotations with SAPR is a non-intrusive solution to bring the benefits of semantics to these systems. And by remaining compliant to the standards, existing non-semantic clients won't be locked out. The proxy itself is not semantically enabled: it does neither enable users to create semantic annotations, nor does it perform reasoning on the semantic annotations. The API supports registration of annotated documents and simple operations such as deleting or listing annotations. It does not support the discovery, since we assume that existing semantically enabled catalogues and service registries already perform this task. Registering a Web service semantically annotated with SAPR to such catalogues then enables the semantic discovery of the annotations.

Merging different annotations is not supported. During the registration of annotated metadata, SAPR assigns a unique identifier (the *SID*). It is later required to retrieve the annotated document, which could then, for example, be registered to a concept-based search engine. A document registered multiple times with different annotations results in different SIDs, and consequently new service metadata URLs. An URL pointing to the same document (but having a different SID) with data quality references would accordingly be used in trust-aware IR systems. SAPR is only providing means to inject annotations into legacy metadata. It is in the responsibility of the client application to propagate the resulting new service metadata links to appropriate applications like search engines.

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 [10] 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.

SAPR is currently used in the ENVISION project[11]. In ENVISION we aim to semantically enrich and chain environmental Web services, with the long-term goal to migrate existing environmental models into the Web. We have also released an API[12] which supports the translation of existing Web services into RDF-based service models. Ontology alignment tools allow for linking these service models to shared domain vocabularies. References to these service models are then added to the service metadata and uploaded to SAPR. The resulting new locations are registered to a search engine supporting concept-based and spatial query processing. In addition, we are currently investigating the semantic annotation of binary data formats such as NetCDF[13], which are common for environmental services. Future work will investigate how these formats can be supported in SAPR.

In this paper an implementation for a decoupled solution for semantic annotations of existing Web services has been presented. The proxy-based approach enables semantically aware clients to benefit from the injected semantic annotations into legacy metadata without touching the original services. The proxy acts on behalf of the hidden Web service; the client does not have to manage the two separate sources. Requests are mediated and result either in updated metadata or redirects to the original source. SAPR is used for the semantic annotation of W3C- and OGC compliant 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. Finally, the stream-based injection approach and the deployment in the cloud ensured a very responsive system useful also for applications with high demands for performance.

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

[11] See `http://www.envision-project.eu`

[12] See `http://kenai.com/projects/envision` for the source code

[13] A description of this format is available at `http://www.unidata.ucar.edu/software/netcdf/`

## 8. Acknowledgements

The presented research has been funded by the BMBF 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

[1] S. Bechhofer, L. Carr, C. A. Goble, S. Kampa, and T. M. Board. The Semantics of Semantic Annotation. In R. Meersman and Z. Tari, editors, *Proceedings of on the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, volume 2519 of *Lecture Notes in Computer Science*, pages 1152–1167. Springer, Berlin / Heidelberg, 2002.

[2] A. Berglund, S. Boag, D. Chamberlin, M. F. Fernández, M. Kay, J. Robie, and J. Siméon. XML Path Language (XPath) 2.0. W3C Recommendation, World Wide Web Consortium (W3C), 2007.

[3] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, Y. Avrithis, S. H, Y. Kompatsiaris, and M. G. Strintzis. Semantic annotation of images and videos for multimedia analysis. In *Proceedings of the Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece*, volume 3532 of *Lecture Notes in Computer Science*, pages 592–607. Springer, Berlin / Heidelberg, 2005.

[4] H. Erdogmus. Cloud Computing: Does Nirvana Hide behind the Nebula? *IEEE Software*, 26(2):4–6, 2009.

[5] M. Graves, A. Constabaris, and D. Brickley. FOAF: Connecting People on the Semantic Web. *Cataloging & classification quarterly*, 43(3):191–202, Apr. 2007.

[6] M. Grčar, E. Klien, and B. Novak. Using Term-Matching Algorithms for the Annotation of Geo-services. In B. Berendt, D. Mladenic, M. de Gemmis, G. Semeraro, M. Spiliopoulou, G. Stumme, V. Svátek, and F. Železný, editors, *Knowledge Discovery Enhanced with Semantic and Social Information*, volume 220 of *Studies in Computational Intelligence*, pages 127–143. Springer Berlin/Heidelberg, 2009.

[7] C. A. Henson, J. K. Pschorr, A. P. Sheth, and K. Thirunarayan. SemSOS: Semantic sensor Observation Service. In *Proceedings of 2009 International Symposium on Collaborative Technologies and Systems (CTS 2009)*, pages 44–53, Baltimore, MA, 2009. IEEE Computer Society.

[8] F. Heylighen. Collective Intelligence and its Implementation on the Web: Algorithms to Develop a Collective Mental Map. *Computational & Mathematical Organization Theory*, 5(3):253–280, Oct. 1999.

[9] W. Hürsch and C. V. Lopes. Separation of Concerns. Technical report, Computer Science Dept., Boston, MA, 1995.

[10] K. Janowicz, S. Schade, A. Bröring, C. Keß ler, P. Maué, and C. Stasch. Semantic Enablement for Spatial Data Infrastructures. *Transactions in GIS*, 14(2):111–129, Apr. 2010.

[11] J. Kahan, M.-R. Koivunen, E. Prud'Hommeaux, and R. Swick. Annotea: an open RDF infrastructure for shared Web annotations. *Computer Networks*, 39(5):589–608, Aug. 2002.

[12] E. Klien, D. Fitzner, and P. Maué. Baseline for Registering and Annotating Geodata in a Semantic Web Service Framework. In S. I. Fabrikant and M. Wachowicz, editors, *Proceedings of 10th AGILE International Conference on Geographic Information Science*, volume 3 of *Lecture Notes in Geoinformation and Cartography*, page 0, Aalborg, Denmark, 2007. Springer Berlin/Heidelberg.

[13] J. Kopecký and T. Vitvar. WSMO-Lite: Lowering the Semantic Web Services Barrier with Modular and Light-Weight Annotations. In *Second IEEE International Conference on Semantic Computing*, volume 0, pages 238–244, Santa Clara, CA, Aug. 2008. IEEE Computer Society.

[14] J. Kopecký, T. Vitvar, C. Bournez, and J. Farrell. SAWSDL: Semantic Annotations for WSDL and XML Schema. *IEEE Internet Computing*, 11(6):60–67, Nov. 2007.

[15] M. Lutz. Ontology-Based Descriptions for Semantic Discovery and Composition of Geoprocessing Services. *GeoInformatica*, 11(1):1–36, Mar. 2007.

[16] C. C. Marshall. Toward an Ecology of Hypertext Annotation. In *HYPERTEXT âĂŹ98: Proceedings of the ninth ACM conference on Hypertext and Hypermedia : Links, Objects, Time and Space - Structure in Hypermedia Systems*, pages 40–49, Pittsburgh, PA, 1998. ACM.

[17] D. Martin, M. Burstein, D. McDermott, S. McIlraith, M. Paolucci, K. Sycara, D. McGuinness, E. Sirin, and N. Srinivasan. Bringing Semantics to Web Services with OWL-S. *World Wide Web*, 10(3):243–277, Sept. 2007.

[18] P. Maué and D. Roman. The ENVISION Environmental Portal and Services Infrastructure. In J. Hrebícek;, G. Schimak;, and R. Denzer, editors, *Environmental Software Systems. Frameworks of eEnvironment*, volume 359 of *IFIP Advances in Information and Communication Technology*, pages 280–294, Brno, Czech Republic, 2011. Springer Berlin/Heidelberg.

[19] P. Maué and S. Schade. Data Integration in the Geospatial Semantic Web. *Journal of Cases on Information Technology (JCIT)*, 11(4):100–122, Oct. 2009.

[20] P. Maué, S. Schade, and P. Duchesne. Semantic Annotations in OGC Standards. {OGC} discussion paper, Open Geospatial Consortium (OGC), July 2009.

[21] A. Padberg and C. Kiehle. Spatial Data Infrastructures and Grid Computing : the GDI-Grid project. In *Proceedings of EGU General Assembly 2009*, volume 11 of *Geophysical Research Abstracts*, page 4242, Vienna, Austria, 2009. European Geosciences Union.

[22] M. Paolucci, K. Sycara, and T. Kawamura. Delivering Semantic Web Services. In G. Hencsey and B. White, editors, *Proceedings of the Twelfth International World Wide Web Conference*, pages 111–118, Budapest, Hungary, 2003. ACM.

[23] F. Reitsma and J. Albrecht. Modeling with the Semantic Web in the Geosciences. *IEEE Intelligent Systems*, 20(2):86–88, 2005.

[24] D. Roman, J. De Bruijn, A. Mocan, H. Lausen, J. Domingue, C. Bussler, and D. Fensel. WWW: WSMO, WSML, and WSMX in a nutshell. In Z. Shi and F. Giunchiglia, editors, *The Semantic Web - ASWC 2006*, volume 4185 of *Lecture Notes in Computer Science*, pages 516–522, Beijing, China, 2006. Springer Berlin/Heidelberg.

[25] D. Roman and E. Klien. SWING - A Semantic Framework for Geospatial Services. In A. Scharl and K. Tochtermann, editors, *The Geospatial Web*, Advanced Information and Knowledge Processing, chapter 23, pages 229–234. Springer London, 2007.

[26] M. Shapiro. Structure and Encapsulation in Distributed Systems: the Proxy Principle. In *Proceedings of the 6th International Conference on Distributed Computing Systems*, pages 198–204, Cambridge, MA, USA, 1986. IEEE Computer Society.

[27] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, June 2007.

[28] K. Sivashanmugam, K. Verma, A. Sheth, and J. Miller. Adding semantics to web services standards. In L.-J. Zhang, editor, *Proceedings of the 1st International Conference on Web Services (ICWS'03)*, pages 395–401, Las Vegas, USA, 2003. CSREA Press.

[29] G. Stamou, J. van Ossenbruggen, J. Pan, and G. Schreiber. Multimedia Annotations on the Semantic Web. *IEEE Multimedia*, 13(1):86–90, 2006.

[30] K. Verma and A. Sheth. Semantically Annotating a Web Service. *IEEE Internet Computing*, 11(2):83–85, 2007.

[31] P. Vretanos. OpenGIS Web Feature Service (WFS) Implementation Specification. {OGC} implementation specification, Open Geospatial Consortium (OGC), 2005.

[32] M. Williams, L. B. D. Cornford, and B. Ingram. Describing and Communicating Uncertainty within the Semantic Web. In F. Bobillo, editor, *Proceedings of the Fourth International Workshop on Uncertainty Reasoning for the Semantic Web*, volume 423 of *CEUR Workshop Proceedings*, Karlsuhe, Germany, 2008. CEUR-WS.