Taking the LIDS off Data Silos

Sebastian Speiser
Karlsruhe Service Research Institute (KSRI)
Karlsruhe Institute of Technology, Germany
Englerstr. 11, D-76131 Karlsruhe
speiser@kit.edu

Andreas Harth
Institute AIFB
Karlsruhe Institute of Technology, Germany
Englerstr. 11, D-76131 Karlsruhe
harth@kit.edu

ABSTRACT

LInked Data Services (LIDS) denote the integration of data-providing services and Linked Data. LIDS are parameterised and formally described web resources which return RDF when dereferenced via HTTP. In this paper we present a general method for creating Linked Data Services; LIDS consist of data access interface conventions that are compatible to Linked Data principles and a lightweight formal description model. Our approach is based on established Web standards including HTTP, RDF and SPARQL. Additionally, we announce several LIDS that we have created from existing real-life services, unlocking vast amounts of triples to the Web of Data.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Web-based services

General Terms

Semantic Web Services

Keywords

Semantic Web, Linked Data, Web Services, Information Integration

1. INTRODUCTION

The trend towards publishing data on the Web is gaining momentum, particularly spurred by the Linking Open Data (LOD) project[1] and several government initiatives to publish public sector data. Data publishers often use Linked Data principles[2] which leverage established Web standards such as Uniform Resource Identifiers (URIs), the Hypertext Transfer Protocol (HTTP) and the Resource Description Framework (RDF[3]). Data providers can easily link their data to data from third parties via reusing URIs. The LOD project proves that the Linked Data approach is, in principle, capable of integrating data from a large number of sources.

However, interlinkage between data on the current Linked Data Web is still low, and a lot of data that could be beneficially interlinked with other data still resides in inaccessible data silos. Reasons include:

- data is dynamically changing, e.g., stock quotes or weather data;
- data is generated dynamically depending on possibly infinite different input data, e.g., distance between two geographical points;
- the data provider does not want arbitrary access to the data, e.g., prices of flight tickets.

Data is often provided via Web services, as services provide a restricted view on a possibly implicit or constantly changing data set. We refer to these services in the following as information or data services. These data-providing services are generally stateless and free of side effects, i.e., do not change the state of the world.

Linked Data interfaces for services have been created, e.g., in form of the book mashup [2] which provides RDF about books based on Amazon's API, or twitter2foaf[4], which encodes a Twitter follower network of a given user based on Twitter's API. These are useful examples for the integration of information services and Linked Data. However, the interfaces are not formally described and thus the link between services and data has to be established manually or by service-specific algorithms. For example, to establish a link between a person instance (e.g., described using the FOAF vocabulary[5]) and her Twitter account, one has to hard-code which property relates people to their Twitter username and the fact that the URI of the person's Twitter representation is created by appending the username to `http://twitter2foaf.appspot.com/id/`.

Vast amounts of idle data can be brought to the Semantic Web via a standardised method for creating Linked Data interfaces to services. The method should incorporate formal service descriptions that enable (semi-)automatic service discovery and integration. We present such an approach for what we call LInked Data Services (LIDS). Specifically, we present the following contributions:

- an access mechanism for LIDS interfaces based on URIs and HTTP (Section 3);
- a lightweight data service description formalism based

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

[2] `http://www.w3.org/DesignIssues/LinkedData`

[3] `http://www.w3.org/TR/rdf-concepts/`

[4] `http://twitter2foaf.appspot.com/`

[5] `http://xmlns.com/foaf/0.1/`

I-SEMANTICS 2010, September 1-3, 2010 Graz, Austria

on SPARQL (Section 4);
- algorithms for linking existing RDF data with LIDS (Section 5);
- application of the presented methods to existing services to expand the current Web of Data (Section 6).

2. USE CASE SCENARIO

Our use case is a scenario involving the analysis of technology companies. Consider an investor who wants to assess the outlook of a potential investment target. The investor could vet the company by navigating an integrated dataset containing basic company data, key personnel, competitors, job openings, IP portfolio and previous VC investments in the company. In addition, the dataset could contain Social Media from Twitter and blogs that allows the investor to gauge the media interest in the company.

All required information is available on the Web, but with three major drawbacks:

- The data is accessible via several protocols, e.g., some data is directly accessible via HTTP GET lookups while other data is hidden behind forms requiring HTTP POST and possibly HTTP cookies.
- The data is encoded in heterogeneous data formats, e.g., trademark data from the United States Patent and Trademark Office is available in HTML; patent data from the European Patent Office in Comma Separated Value files; CrunchBase company descriptions, Indeed.com job offers and Twitter messages in JSON; and data from GeoNames services in XML.
- The data is sparsely interlinked, e.g., there exists no link between a company's office location and its GeoNames location, and no link between a company and its trademarks.

Consider data about company offices, which contains latitude and longitude attributes:

```
#usa-palo-alto-hq geo:lat "37.416" .
#usa-palo-alto-hq geo:long "-122.152" .
```

A GeoNames service call to find nearby populated places[6] returns:

```
<geonames>
  <geoname>
    <name>College Terrace</name>
    <geonameId>5338647</geonameId>
    ...
  </geoname>
</geonames>
```

Based on the available data one could establish a `foaf:based_near` connection between `#usa-palo-alto-hq` and `http://sws.geonames.org/7288147/`, however, that step would require specialised code. Unlocking the data for automated integration and processing requires:

- Linked Data interfaces to all services and data sources, so that data can be easily accessed and integrated;
- formal service descriptions, so that links between data from different sources can be created automatically.

3. LIDS METHOD

Information services provide data that is related in a specific way to the given parameters. For example, the GeoNames `findNearbyWikipedia` service relates a populated place to the given latitude/longitude parameters. We extend that notion for Linked Data Services as follows:

A Linked Data Service (LIDS) provides URIs for entities representing service inputs that encode parameters as key-value pairs in the query string. Dereferencing the URI via HTTP GET returns an RDF description of the service input entity, its relation to the service output and the output data itself.

For example, the LIDS wrapper for the GeoNames `findNearbyWikipedia` service requires geocoordinates as parameters encoded in the URI of the wrapper: `http://geowrap.openlids.org/findNearbyWikipedia?lat=37.416&lng=-122.152`. We explain how to construct these URIs from service descriptions in Section 5.

Looking up the service URI returns URIs of nearby places from Wikipedia (we substitute Wikipedia URIs with those from DBpedia[7]) and the relation to a "non-information" URI denoting a point:

```
@prefix dbp: <http://dbpedia.org/resource/> .
<http://geowrap...Wikipedia?lat=37.416&lng=-122.152#point>
   foaf:based_near dbp:Palo_Alto%2C_California ;
   foaf:based_near dbp:Packard%27s_garage .
```

To establish equivalence between `#usa-palo-alto-hq` and `http://geowrap.openlids.org/findNearbyWikipedia?lat=37.416&lng=-122.152#point` we can use `owl:sameAs`.

URIs are generally constructed in the following way:

```
[endpoint](pars)#InputName
pars = ?parameter1=value1&parameter2=value2&...
```

or if only one required parameter with value `value1` exists, additionally the following shortcut URI is supported:

```
[endpoint]/value#InputName
```

The local part `InputName` is replaced by the type of input given to the service and is used to distinguish between the document containing the service's result and the contained result individual.

4. SERVICE DESCRIPTIONS

We define a simple vocabulary for LIDS[8] that defines a class for LIDS and a description property relating a LIDS to a SPARQL query using the CONSTRUCT operator and unsafe variables. A service description is given in the following way:

```
CONSTRUCT { [io-relation] } FROM   [endpoint]
                            WHERE {  [input] }
```

We restrict both `[input]` and `[io-relation]` to basic graph patterns, i.e. conjunctions of triple patterns.

- `[input]`: The required input values and their relations to each other. The variables in the input relation are the service parameters as defined in Section 3.
- `[endpoint]`: A URI that is used as the base URI when constructing a service call as described in Section 3.
- `[io-relation]`: Relates one of the input variables (corresponding to the `InputName` of Section 3) to the output variables, which are unsafe, i.e. did not appear in the WHERE clause. The expression does not have to fully specify all descriptions that will be returned, but

[6] `http://ws.geonames.org/findNearbyPlaceName?lat=37.416&lng=-122.152`

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

[8] `http://openlids.org/vocab`

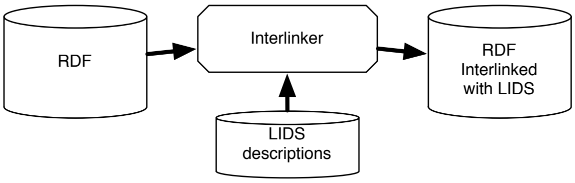

Figure 1: Interlinking of RDF with LIDS

only the minimum information, that is returned. Thus output is semi-structured data, and can be arbitrarily extended.

The use of unsafe variables and the meaning of the endpoint are not completely adhering to the SPARQL standard, but have their intuitive meaning. An example description of the geo service presented in Section 3 is given in the following:

```
CONSTRUCT { ?point foaf:based_near ?feature }
FROM <http://geowrap.openlids.org/findNearbyWikipedia>
WHERE { ?point geo:lat ?lat . ?point geo:long ?lng }
```

The unsafe variables (here: `?feature`) are bound by the service. The variable appearing both in `[input]` and `[io-relation]` (here: `?point`) is the input object and used as `#InputName` (here: `#point`) when building service call URIs.

We generally assume that a service operates on literal values, so only those variables specified in the `[input]` basic graph pattern have to be given.If the actual URI of an individual is needed, a URI can be expressed as Literal using the `log:uri` property as proposed by Berners-Lee et al. [1].

5. ENRICHING LINKED DATA WITH LIDS

Existing Linked Data can be automatically enriched with links to LIDS. This can happen in different settings, consider e.g.:

- Processing of a static RDF data set, inserting links to LIDS, and storing the new data.
- A Linked Data endpoint that serves data, and dynamically adds links to LIDS to the result.
- A Linked Data browser that locally augments retrieved data with data retrieved from LIDS.

Given an RDF graph, determining matching data for a given service can be realised by evaluating a SELECT SPARQL query of the following form: `SELECT * WHERE { [input] }` The returned bindings can be used to construct an URI which is `sameAs` the binding value of the `InputName` variable. The sameAs-relation can be either explicitly added to the data set or e.g. in the case of a Linked Data browser the equivalent LIDS URI can be resolved and the obtained data can be added to the description of the input entity. The query for the geo service example is:

```
SELECT ?point ?lat ?lng WHERE
      { ?point geo:lat ?lat; geo:long ?lng }.
```

For a binding `?point = #usa-palo-alto-hq, ?lat = '37.416', ?lng = '-122.152'`, the following triple would be inferred:

```
#usa-palo-alto-hq owl:sameAs
  <http://geowrap.openlids.org/findNearbyWikipedia?
                             lat=37.416&lng=-122.152#point>.
```

Figure 1 illustrates this interlinking process, for which we provide a Java tool.[9]

6. APPLICATIONS

We created LIDS from the sources required to realise our use case scenario from Section 2. In the following we list the LIDS that we made publicly available. The services are also linked on `http://openlids.org` together with their formal LIDS descriptions and further information, such as URIs of example entities.

- CrunchBase Wrapper[10] provides information about tech companies, their funding, founders, top employees, products, and competitors.
- GeoNames Wrapper[11] provides three functions:
  - finding the nearest GeoNames feature to a given point,
  - finding the nearest GeoNames populated place to a given point,
  - linking a geographic point to resources from DBpedia that are nearby.
- Twitter Wrapper[12] links Twitter account holders to the messages they post.
- Feedwrapper[13] provides SIOC data about RSS and Atom feeds.

The services are deployed on Google's App Engine cloud environment.

7. RELATED WORK

Early Web service description formalisms, such as WSDL, do not model the relation between input and output data, which leaves space for ambiguities.

General Semantic Web Services approaches include OWL-S[14] and WSMO [5] but still lack practical applications, which can be partially explained by their complexity and their use of formalisms that are not familiar to all Semantic Web users. In contrast, our solution relies on standard and well-known technologies, namely SPARQL, HTTP and RDF. Furthermore LIDS are a match for the semi-structured and decentralised nature of Linked Data, despite having a logical foundation.

Most closely related to our service descriptions formalism are works on semantic descriptions of stateless services (e.g. [4, 3, 6]). Similar to our approach these solutions define service functionality in terms of input and output conditions. Most of them, except [4], employ proprietary description formalisms. In contrast, our approach relies on standard SPARQL. Furthermore our work provides the following key advantages: i) a methodology to provide a Linked Data interface to services, ii) semi-structured input and output definitions, compared to the static definition of required inputs and outputs in previous approaches.

8. CONCLUSIONS

We have presented an approach for the integration of data services with Linked Data. Using an uniform method for creating access interfaces that are compatible to Linked Data principles enables the creation of LInked Data Services (LIDS) from previously inaccessible data silos. LIDS have formal, yet lightweight and flexible descriptions based on SPARQL, a language which is familiar to many Semantic Web users and developers. By fulfilling a real-world use

[9] `http://code.google.com/p/openlids`

[10] `http://cbasewrap.ontologycentral.com/`

[11] `http://geowrap.openlids.org/`

[12] `http://twitterwrap.ontologycentral.com/`

[13] `http://feedwrap.openlids.org/`

[14] `http://www.w3.org/Submission/OWL-S/`

case covering data about about tech companies with LIDS, we evaluated our approach and contributed vast amounts of triples to the Web of Data.

9. REFERENCES

[1] T. Berners-Lee, D. Connolly, L. Kagal, Y. Scharf, and J. Hendler. N3Logic: A logical framework for the World Wide Web. *Theory and Practice of Logic Programming*, 8(03), 2008.

[2] C. Bizer, R. Cyganiak, and T. Gauss. The RDF Book Mashup: From Web APIs to a Web of Data. In *Workshop on Scripting for the Semantic Web, at ESWC*, 2007.

[3] D. Hull, E. Zolin, A. Bovykin, I. Horrocks, U. Sattler, and R. Stevens. Deciding Semantic Matching of Stateless Services. *AAAI06*, pages 1319–1324, 2006.

[4] K. Iqbal, M. L. Sbodio, V. Peristeras, and G. Giuliani. Semantic Service Discovery using SAWSDL and SPARQL. In *International Conference on Semantics, Knowledge and Grid*, 2008.

[5] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web service modeling ontology. *Applied Ontology*, 1(1):77–106, 2005.

[6] W.-f. Zhao and J.-l. Chen. Toward Automatic Discovery and Invocation of Information-Providing Web Services. In *Asian Semantic Web Conference*, 2006.