Towards Linked Data Services

Sebastian Speiser and Andreas Harth

Institute AIFB, Karlsruhe Institute of Technology (KIT), Germany
`lastname@kit.edu`

Abstract. Large amounts of data reside in sources which are not web-accessible. Wrappers – small software programs that provide uniform access to data – are often used to transform legacy data sources for use on the Semantic Web. Wrappers, as well as links between data from wrapped sources and data that already exists on the Web, are typically created in an ad-hoc fashion. We propose a principled approach to integrating data-providing services with Linked Data. Linked Data Services (LIDS) can be used in various application scenarios to provide uniform access to legacy data and enable automatic interlinkage with existing data sets.

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 publishing 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].

Web services are often used to access frequently changing data sets or data which is computed based on supplied input parameters. Wrappers that provide uniform access to services have been created, e.g., in form of the book mashup [1] which returns RDF about books based on Amazon's API, or twitter2foaf[4], which provides access to the follower network of a given user based on Twitter's API. These are useful examples of exposing data from services in a common protocol (HTTP) and data format (RDF).

A lot of data – even data that is already accessible via a uniform access mechanism – still exists in form of unconnected data islands; interlinkage between data on the current Linked Data Web is low. Using ad-hoc wrappers, interlinkage has to be done manually or by service-specific algorithms. Users have to manually associate existing URIs with URIs from wrappers. 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 the relation between an existing

[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/`

URI and the person's Twitter wrapper URI, which is created by appending the username to `http://twitter2foaf.appspot.com/id/`.

Legacy data should be published so that automated integration and processing is possible, which requires:

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

We present preliminary work on such an approach to create what we call LInked Data Services (LIDS). Using LIDS, vast amounts of idle data can be brought to the Semantic Web via a standardised method for creating Linked Data interfaces to services. LIDS, in addition, enable (semi-)automatic service discovery and integration.

2 Scenario

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. All required information is available on the Web, but with three major drawbacks: i) data is accessible via several incompatible protocols ii) data is encoded in various character encodings and syntaxes and iii) data is sparsely interlinked.

Consider a description of a company office (e.g. `#deu-karlsruhe-`), which contains latitude and longitude attributes (e.g. 48.996 and 8.463), and a GeoNames service that finds nearby places described in Wikipedia[6]. Calling the service with the given latitude/longitude returns:

```
<geonames> <entry>
  <title>Turmbergbahn</title>
  <summary>The Turmbergbahn is a funicular railway near
       Karlsruhe in Germany...</summary>
  <wikipediaUrl>http://en.wikipedia.org/wiki/Turmbergbahn</wikipediaUrl>
  ...
 </entry>
 <entry>
  <title>Durlach</title>
  <summary>Durlach is a borough of the German city of Karlsruhe..</summary>
  ...
 </geonames>
```

Based on the available data one could establish a `foaf:based_near` connection between `#deu-karlsruhe-` and `http://dbpedia.org/resource/Turmbergbahn`, however, that step would require specialised code.

[6] `http://ws.geonames.org/findNearbyWikipedia`

3 Linked Data Services

In the following we describe the LIDS approach using an example. We focus on data-providing services, which return data that is related in a specific way to the given parameters. We extend that notion for Linked Data Services as follows:

> *A Linked Data Service (LIDS) provides HTTP 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. Both input and output of a LIDS are formally described using SPARQL.*

For example, a LIDS wrapper for the exemplary GeoNames service is available at `http://geowrap.openlids.org/findNearbyWikipedia`. Input parameters are encoded in the query string of the service URI, e.g. `http://geowrap.openlids.org/findNearbyWikipedia?lat=48.996&lng=8.463`. To establish an "non-information" URI we add a localname to derive the entity URI, e.g. `http://geowrap.openlids.org/findNearbyWikipedia?lat=48.996&lng=8.463#point`. Looking up the entity returns a "non-information" URI denoting the input point and the relation to URIs of nearby places from Wikipedia (we substitute Wikipedia URIs with those from DBpedia[7]):

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dbp: <http://dbpedia.org/resource/> .
<http://geowrap...Wikipedia?lat=48.996&lng=8.643#point>
                                foaf:based_near dbp:Turmbergbahn ;
                                foaf:based_near dbp:Durlach .
```

A LIDS returning a URI with the input parameter allows for adding additional descriptions to that URI; e.g., we could use the `owl:sameAs` property to establish equivalence between `http://geowrap.openlids.org/findNearbyWikipedia?lat=48.996&lng=8.463#point` and `#deu-karlsruhe-`.

We propose a simple vocabulary for LIDS[8] that defines a class for LIDS and a description property relating a LIDS to a SPARQL query describing the service. The LIDS description of the geowrap service is as follows:

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

The use of unsafe variables and the meaning of the FROM clause are not completely adhering to the SPARQL standard, but have their intuitive meaning. The FROM clause encodes the base URI of the service. Please note the user of unsafe variables (here: `?feature`), which are bound by the service during execution. The variable appearing in both CONSTRUCT and WHERE clause (here: `?point`) denotes the input entity and is used for building entity URIs.

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

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

4 Related Work

In contrast to our proposal, 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[9] and WSMO [4] 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. Semantic descriptions of stateless services (e.g. [3, 2, 5]) define service functionality in terms of input and output conditions, however, employ proprietary description formalisms and/or static definition of inputs and outputs. Our approach is more flexible and better integrates with the Linked Data principles.

5 Conclusions and Future Work

We have presented preliminary work on an approach to integrating data services with Linked Data. A uniform access method – compatible to Linked Data principles – in combination with a lightweight service description, enables the creation of LInked Data Services (LIDS). LIDS have formal, yet lightweight and flexible descriptions based on SPARQL, a language which is familiar to many Semantic Web users and developers.

LIDS can be used in a number of scenarios. In the future, we plan to develop and study algorithms that automatically enrich existing Linked Data with links to LIDS, which can happen in different settings, e.g.:

- processing of static RDF data, inserting links to LIDS, and storing the result;
- dynamically adding links to LIDS to the result of a Linked Data endpoint;
- locally augmenting retrieved data with data from LIDS in a data browser;
- using data from LIDS during SPARQL query processing.

For a formal treatment of LIDS, more example services and up-to-date information visit `http://openlids.org/`.

References

1. 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.
2. D. Hull, E. Zolin, A. Bovykin, I. Horrocks, U. Sattler, and R. Stevens. Deciding Semantic Matching of Stateless Services. *AAAI06*, pages 1319–1324, 2006.
3. 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.
4. 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.
5. W.-f. Zhao and J.-l. Chen. Toward Automatic Discovery and Invocation of Information-Providing Web Services. In *Asian Semantic Web Conference*, 2006.

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