Geospatial Linked Open Services

Barry Norton[1] and Reto Krummenacher[2]

[1] AIFB, Karlsruhe Institute of Technology, Germany
[2] Semantic Technology Institute, University of Innsbruck, Austria
firstname.lastname@{kit.edu | sti-innsbruck.at}

Abstract. Linked Open Services are a principled attempt to guide the creation and exposure of online services, in order to enable productive use of, and contribution to, Linked Data. They communicate RDF directly, via HTTP-based interaction, and their contribution to the consumer's knowledge, in terms of Linked Data, is described using SPARQL. To achieve this they bring together, and extend, the principles of Linked Data and of REST. In this paper we focus on the development of Linked Open Services for the geospatial domain and explain how they overcome the short-comings of existing services over the geonames dataset, a core part of the Linked Open Data Cloud.

1 Introduction

On the most basic level Linked Open Services (LOS[1]) [3] concern the open exposure of functionalities on the Web using semantic technologies. This is by no means a novel aim, and has been pursued by so-called 'Semantic Web Services' for a number of years. Semantic Web Services, however, largely build on the WS-* Stack, apply semantics within closed platforms — such as the OWL-S Virtual Machine [5], WSMX [2] and IRS-III [1] — and target integration within (XML-based) enterprise systems. LOS, on the other hand, are primarily intended to provide reusable functionalities to Linked Data-aware clients.

The aim of LOS is not just to apply the existing Linked Data principles to services, an approach already proposed in [6], but to propose a list of further service-specific principles to be followed in openly exposing services over Linked Data:

1. Describe services as *LOD prosumers* with input and output descriptions as *SPARQL graph patterns*.
2. *Communicate RDF* by *RESTful content negotiation*.
3. Communicate and describe the *knowledge contribution* resulting from service interaction, including *implicit knowledge* relating input, output and service provider.

Associated with the last principle is an optional fourth:

4 When wrapping non-LOS services, extend the (lifted, if non-RDF) message to make explicit the implicit knowledge, and to use Linked Data vocabularies, using *SPARQL CONSTRUCT* queries.

[1] Or, more ambitiously, the exhortation 'LOS!'

Perhaps the most important important vocabulary for geospatial information is the encoding of the coordinate system of the 1984 World Geodetic System, WGS84 [4], in RDF Schema[2]. The International Civil Aviation Organisation (ICAO) is responsible for giving identifiers to airports, and for a standard scheme for encoding weather observations, taken up internationally by the World Meteorological Organization, METAR [7]. While there is a record of an RDF encoding of ICAO codes for airport information, this has been lost due to the Web. Similarly there was a, pre-Web Ontology Language, partial METAR encoding in DAML recorded from the University of Maryland (in the DAML Ontology Library[3]), which is lost, and an extant one (of unclear relationship to the other) from the University of Aberdeen[4].

In order to support location-based weather reports as Linked Open Services, we therefore work with a standards-based (RDFS) adaptation to the existing DAML METAR ontology, convert (JSON-based) METAR reports into RDF using the classes and properties defined, and then recreate Linked Data resources for ICAO-identified airports, minting new URIs based on these codes. As well as referring to these resources in weather requests from ad hoc locations, also encoded using the WGS84 vocabulary, we attach airport-specific requests to the resources themselves in RESTful style, demonstrating the natural fit of the full scope of RESTful services (beyond GET methods) and Linked Data. In the following Section 2 explains how the conversion fits over the existing geonames service as a Linked Open Service wrapper, Section 3 explains how the resource-based services operate alongside standard Linked Data principles. Finally, we conclude in Section 4.

2 LOS Wrappers

Currently XML and JSON (plus, perhaps, YAML) are the predominant syntaxes for the exchange of dynamic information on the Web in general, including specifically, which is unfortunate, for a significant number of services over the Linked (Open) Data Cloud. We point out, for example, that all but one of the 36 GeoNames services, including METAR-based weather services, do not support RDF (most support JSON, with XML in close second). In order to expose such non-RDF Web services as LOS, it is necessary to transform from RDF to the expected data format of the service implementation, and to map the output of the service back into RDF. In the case of the procedure-oriented GeoNames *Weather* services (nearby to point, and within bounding box), the invocation is triggered by an HTTP POST request, and the result data is serialized as either a JSON Array or an XML document. Our LOS versions of these weather services are available at `http://www.linkedopenservices.org/services/geonames/weather/`, where the LOS-specific input and output descriptions are given as exemplified below.

Input (bounding box version):

```
[a wgs84:Point; wgs84:lat ?north; wgs84:long ?west]
[a wgs84:Point; wgs84:lat ?south; wgs84:long ?east]
```

[2] `http://www.w3.org/2003/01/geo/`
[3] `http://www.daml.org/ontologies/241`
[4] `http://www.csd.abdn.ac.uk/research/AgentCities/WeatherAgent/weather-ont.daml`

Output:

```
[ metar:weatherObservation [
  weather:hasStationID ?icao ;
  wgs84:lat ?lat ; wgs84:long ?lng ; wgs84:alt ?alt ;
  metar:datetime ?dateTime ;
  metar:observation ?observation ;
  weather:hasVisibilityEvent ?clouds ;
  weather:hasWindEvent [weather:windDirection ?windDirection],
                       [weather:windSpeedKnots ?windSpeed] ;
  weather:hasTemperatureEvent
    [a weather:CurrentTemperature ; weather:celsiusTemperature ?temperature],
    [a weather:CurrentDewPoint ; weather:celsiusTemperature ?dewPoint],
    [weather:humidityPercent ?humidity] ;
  weather:hasWeatherEvent ?condition ;
  weather:hasPressureEvent [metar:hectoPascal ?pressure ] ] ]
```

In the simplest case the transformation consists of applying the generic ‘JSON2RDF’ library[5] (which creates no real semantics, but puts the data into graph form so it can be queried), followed by a SPARQL CONSTRUCT query (into instances in existing vocabularies) in order to automatically create a useful RDF model. As well as using existing standard predicates to give semantics to the output data *in itself*, the query is enabled to create links between input and output data, encoding knowledge that is normally only implicit in service interaction (“this is the weather nearest to the point...”), and the machinery may also encode provenance information (“according to an invocation of [service X] at [time Y]”.

The query has a head equal to the output pattern of the LOS, and is hence an explicit specification of how to transform the internal json2rdf data into an RDF graph that provides the promised knowledge contribution of the service. For the particular case of our LOS! GeoNames Weather service we introduced an additional intermediate step that lifts the string values of, for example, json2rdf:clouds to a URL of the enhanced METAR ontology (metar:FewClouds). [6] Among the extensions we have made there are internationalised labels for the different weather conditions, extended beyond those offered by the geonames services.

3 Resource-Oriented LOS

The services in the previous section return lists of airport-based weather stations (identified by ‘ICAO’ codes) with the most recent weather observation, according to their proximity to some transient geographical input. Geonames also, however, offers a service ‘weatherIcao’ that returns the weather for a given METAR station. Unfortunately neither does geonames give a Linked Data style resolvable URI for these stations, nor does the previous attempt to form Linked Data from them[7] seem still to exist. At the LOS community website we have therefore decided to adopt a resource-oriented (i.e., truly RESTful) approach to METAR stations as spatial resources. The collection `http://www.linkedopenservices.org/services/geo/SpatialResources/point/` has been expanded to include the current METAR lists.

[5] `http://www.linkedopenservices.org/wiki/index.php/JSON2RDF`

[6] The ontology at `http://www.linkedopenservice.org/ns/METAR` is an extension to the DAML ontology discussed above.

[7] See: `http://lists.xml.org/archives/xml-dev/199907/msg00415.html`

For ICAO (we have also minted URIs for IATA, International Air Transport Association, codes) more than 50 000 triples have been created of the following pattern:

Output:

```
_:airport rdf:type point:ICAO ;
       wgs84:lat ?lat ; wgs84:long ?long ; wgs84:alt ?alt ;
      rdf:label ?label
OPTIONAL {_:airport owl:sameAs ?iata . ?iata a point:IATA}
```

The most important consequence of the exposure of these resources for LOS, however, is that they are the basis for supporting a resource-oriented version of the weather service, which can be invoked simply by GETting from corresponding URIs of the form:

`http://www.linkedopenservices.org/services/geo/SpatialResources/point/{ex}/weather`

This returns RDF of a similar scheme to that presented in the last section.

4 Conclusion

In this paper we have motivated and introduced the principles guiding the creation of Linked Open Services and their application to geospatial data. We have illustrated how simple wrappers can be formed from existing procedure-oriented services and furthermore how a resource-oriented service can optimally combine the principles of REST and of Linked Data. In both cases RDF is available, via HTTP content negotation, for direct communication and SPARQL graph patterns are used to describe the required input, and the expected output, capturing the full knowledge contribution of service execution.

A LOS wrapper specifies not only what graph patterns a service consumes and produces, respectively, but also how the input RDF is 'lowered' to the expected data format of the service, respectively how the output of the service is 'lifted' back to RDF by means of a SPARQL CONSTRUCT with a head equal to the output pattern. We have already produced a helpful library, JSON2RDF, to aid this wrapping process.

Acknowledgement: The work is supported by the EU projects SOA4All and SEALS.

References

1. Domingue, J., Cabral, L., Hakimpour, F.and Sell, D., Motta, E.: IRS III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services. In: Proceedings of the Workshop on WSMO Implementations (WIW 2004) (2004)
2. Haller, A., Cimpian, E., Mocan, A., Oren, E., Bussler, C.: Wsmx - a semantic service-oriented architecture. In: ICWS '05: Proceedings of the IEEE International Conference on Web Services. pp. 321–328. IEEE Computer Society (2005)
3. Krummenacher, R., Norton, B., Marte, A.: Towards Linked Open Services. In: 3rd Future Internet Symposium (September 2010)
4. National Imagery and Mapping Agency: World geodetic system 1984. Tech. rep., National Imagery and Mapping Agency (2004)
5. Paolucci, M., Ankolekar, A., Srinivasan, N., Sycara, K.: The DAML-S virtual machine. In: The SemanticWeb - ISWC 2003. LNCS, vol. 2870, pp. 290–305. Springer (2003)
6. Pedrinaci, C., Domingue, J., Krummenacher, R.: Services and the Web of Data: An Unexploited Symbiosis. In: AAAI Spring Symposium (March 2010)
7. World Meteorological Organization: Aerodrome reports and forecasts: A user's handbook to the codes. Tech. rep., World Meteorological Organization (2008)