A Sitemap extension to enable efficient interaction with large quantity of Linked Data

Giovanni Tummarello

DERI Galway

Abstract. This paper describes an extension for the Sitemap protocol targeted at the efficient discovery and use of RDF data. Data publishers can state where RDF is located and alternative means to access it. Semantic Web clients and Semantic Web crawlers can use this information to access required RDF data in the most efficient way for the task they have to perform.

1. Introduction and rationale

Using Semantic Web standards such as RDF and SPARQL seems to many a very logical choice to make available structured data online. This has to do with the self describing nature of RDF: the use of shared ontologies can enables agent to understand the relationship between entities as well as the nature of the entities themselves. Furthermore, Semantic Web formats enjoy a large community and the support from W3C as well as a rich set of existing libraries and tools to process such data.

This said, many ways exists in which semantically structured data on the Semantic Web (SW) can be made available and consumed. For example, a SW database online could use retrievable URLs as URI for its internal concepts, according to the Linked Data paradigm [1] (from now on such URIs are here called URI/URLs). It could furthermore, or instead, offer dumps containing the entire database available for download, or offer a SPARQL access point, embed information in XML dialects such as RDFa, etc. While data published in different manners might be identical, the implications of accessing it in one way or the other might be quite significant.

For example, if a client wanted to execute a relatively simple query over the DBPedia database, it should probably use the available SPARQL service. On the other hand if it wanted to execute a large number of queries, it should probably download the RDF dump and run them locally. Equally, to access the most recent updates for a specific concept, said client should access DBPedia's specific URLs rather than downloading the entire database too often. Similar issue arises with Semantic Web crawlers: downloading a dump instead of fetching each URI/URL of a Linked Data site can very significantly reduce consumption of networking and computing resources both on the client and server side.

To address these issues we introduce the Semantic Crawler extension to the Sitemap protocol [2]. By using such an extension, a host can both avoid denial of service by compliant robots and help compliant clients find alternative and possibly better ways to access the host data.

2. Description

The Sitemap protocol defines a way to create an XML file by which automatic agents such as crawlers can obtain a list of URLs which they should index. This happens thanks to tags which describe the location of each crawlable resource along with meta-information such as, for example, the expected rate of change for each individual URL or the date when this was last modified [2]. The protocol also defines a way to extend robot.txt, so that a robot can find the location of said map in the website Sitemap [3]. An example of sitemap is shown in the following listing:

```
<?xml version="1.0" encoding="UTF-8"?>
 <urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
   <loc>http://www.example.com/</loc>
   <lastmod>2005-01-01</lastmod>
   <changefreq>monthly</changefreq>
   <priority>0.8</priority>
  </url>
 </urlset>
```

The Semantic Crawler extension that we propose in this paper consists of special tags to use within a Sitemap. The main focus is to describe equivalent but alternative ways for a Semantic Robot (client or crawler) to access Semantic Web data offered by a host. One such equivalence is declared by the `<cs:dataset>` tag defined in this extension, to be used at the same level as `<url>` tags in the Sitemap.

A dataset can have many *access options*, or ways to be accessed. The semantics of the `<sc:dataset>` tag is such that the data underlying each of the access options is the same. For example, this means that the merge of the RDF descriptions of all the linked data URLs hosted by the server should be contained (in RDF terms [3]) in the data pointed to by the `<sc:dataDumpLocation>` tag or in the database powering the SPARQL end point pointed to by the `<sc:sparqlEndpointLocation>` tag.

Cases where this is not true should be limited to technical delays of relatively minor importance; introduced, for example, by the creation of the RDF Dump happening only at certain times of the day or by server side caching of the RDF descripion of popular URI/URLs.

If a Sitemap contains several dataset definitions, these are treated independently.

Note that for the examples in this document we will assume that the `cs` namespace is mapped to the location of this extension: `http://sw.deri.org/2007/07/sitemapextension/scschema.xsd` . This is usually done in the `<urlset>` opening Sitemap tag, as shown in the example section.

2.1. Valid elements inside a dataset definition

<sc:linkedDataPrefix>

A prefix for Linked Data that a server hosts. URI/URLs that begin with this prefix will resolve to their RDF description.

There can be any number of `<sc:linkedDataPrefix>` in a dataset: in the case of multiple definitions, the dataset is said to contain the union of all the Linked Data served under the different prefixes.

<sc:dataDumpLocation>

Indicates the location of an RDF data dump.There can be any number of `<sc:dataDumpLocation>` and/or `<sc:sparqlEndpointLocation>` in a dataset; the interpretation is that of "mirror locations", both for dumps and for SPARQL endpoints.

Using a different host for a mirror is possible (e.g., a backup SPARQL server residing on a different host or a mirror RDF dump). A client however is expected to take this equivalence statement as "authoritative only if the same statements are also given in the Sitemap on the second host.

In the case of multiple such tags, the *priority* integer attribute can be specified in each tag to indicate the preference of the source:

- A value of 0 for priority means that the source should not be used unless the other one is not responding.
- A value different than 0 indicates that the source should be used with a probability equal to the priority divided by the sum of the priorities of all the mirror locations.
- For sources not specifying a priority, the default priority value is 1.

<sc:dataFragmentDump>

Indicates the location of a fragment of an RDF data dump, used for large datasets which are "split" over several files. There should be zero or more than one `<sc:dataFragmentDump>` in a dataset: if an RDF dataset is fragmented then the fragments should number at least two. Fragments can also be indicated to be hosted on different hosts, following the same "authoritativness" understanding as explained in the `<sc:dataDumpLocation>` tag description.

<sc:sparqlEndpointLocation>

The location of a SPARQL endpoint for the dataset. There can be any number of `<sc:sparqlEndpointLocation>` in a dataset; the interpretation is that of "mirror locations". As `<sc:dataDumpLocation>`, this tag allows a *priority* integer attribute.

<sc:datasetURI>

The URI of the current dataset. It is RECOMMENDED to set a URI for the dataset, as it facilitates the provision of further annotations. There can be zero or one `<sc:datasetURI>` and its value must not be the same of other datasets offered on the same site. As a reasonable practice, this URI is usually chosen to be a URI/URL (a URI which is also a URL that points to an RDF file describes it), thus possibly pointing to a further description of the dataset itself.

<sc:datasetLabel>

It is reccomended to set a describing label for the dataset. There can be zero or one `<sc:datasetLabel>`.

<changefreq>

This element, as defined by the Sitemap protocol, describes how often it is expected that the dataset will be updated (e.g. *monthly*). See [2] for more information. There can be zero or one `<changefreq>`.

<sc:sampleURI>

This element can be used to point to a URI within the dataset which can be considered a representative "sample". This element is particularly significant if the URI is a URI/URL, in which case such sample can be quickly retrieved. There can be any number of `<sc:sampleURI>` elements defined in a dataset.

2.2. Behavior from a compliant client

To be compliant with this extension, a Semantic Web spider or client must check and interpret dataset tags. This means checking for robot.txt, looking for the Sitemap location, retrieving the Sitemap and interpreting it to chose the most appropriate and in general, least intrusive way to access the data.

For example it must download a full data dump rather than crawl the entire resolvable URL space, while on the other hand it should not re-download the file (but rather access directly the URI/URL) if it is looking for a specific update. In agreement with the Sitemap protocol itself, a spider might however decide a re-crawling schedule contrary to the one indicated in the `<changefreq>` tag.

2.3. Serialising Linked Data: dump formats

Usually, RDF data can be serialised using the RDF/XML standard. However, the case of Linked Data is exceptional: individual RDF graphs are returned at each of the Linked Data URI/URLs. These individual graphs might somehow contain additional

informations, e.g. have different publication dates or authorship expressed in RDF statements which are not reflected in the single RDF/XML dump.

Serialising a Linked Data website would therefore, in theory, require a serialisation format whereby each triple is assigned to the source URL, thereby conforming to the "quadruple based" paradigm (or "named graph" paradigm).

Two cases can now be distinguished:

- Linked Data sites which generate the individual URI/URL graphs from a single RDF dataset by applying a simple deterministic rule (e.g., a Concise Bound Description operator [5] or union of all the Minimal Self Contained Graphs [6] involving the given URI). In this case a simple RDF/XML dump can be provided and will suffice most use cases as it will provide all the "actual knowledge" that the site is serving.
- Linked Data sites whereby the RDF models at each URI/URL consist of statements not abiding to a fixed rule. In this case the knowledge served by the site cannot in actuality be represented as a monolithic RDF model, but it should rather be considered as a set of graphs. In this case the dump should be provided in a format where the provenance of each triple is specified.

As there is currently no widely supported standard for the latter case, we report several alternatives:

- A zip or gzip file whereby each file name is the URL encoded URI/URL and the content is an RDF/XML serialisation of the content served by resolving the URI/URL. It is to be noticed that the original zip format has a limitation of 65535 files per archive. Sites using this method might be forced to provide fragmented dumps. Tar/Gzipped archives do not have this limitation.
- A dump using TRIX [7] or TRIG [8] or NQUADS [9]: formats which all support triples with context (named graphs serialisation formats).

Arguing for or against any of these formats or methods is outside the scope of this document. In absence of standardised formats, it is expected that data producers will follow previous examples and thus probably use the formats specified above. Clients will thereby required to be flexible enough to automatically comprehend the dump format and potential modifiers (e.g. a possible zip or gzip compression) and process it correctly.

2.4. Security Issues

In adopting a dataset definition on a website which serves RDF, the adopters must be aware that, just like robots.txt or Sitemaps, the mechanism works on a voluntary basis. It is up to Semantic Crawler developers to keep the most respectful behavior toward the resources offered by the server by accessing sitemap file and properly interpreting it.

3. Examples

The following examples illustrate Sitemaps using this extension. Please feel free to copy and modify these examples for your own purposes.

3.1. The URLSET preamble

To use the hereby defined extended terms, the usual Sitemap urlset element needs to include the proper namespace definitions. In practice, rather than the usual:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
.....
.....
```

The following should be used:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
         http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"
         xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
        xmlns:sc="http://sw.deri.org/2007/07/sitemapextension/scschema.xsd">
....
....
```

which points the "sc" namespace to this extension as defined by the schema available online at http://sw.deri.org/2007/07/sitemapextension/scschema.xsd

3.2. Advertising the Sitemap in robot.txt

Once a Sitemap has been composed, it should be saved on the root directory,of the web site, usually named "sitemap.xml", and the following line should be added to the robots.txt to enable autodiscovery:

```
Sitemap: http://www.example.com/sitemap.xml
```

3.3. A basic example

The following example states that a dataset with label "Product catalog for Example.org" is available at http://example.org/cataloguedump.rdf. Furthermore, such data provides the content for resolution of URI/URLs in the space http://example.org/products/ . Finally, this data is said to change with a monthly frequency.

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
         http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"
         xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
        xmlns:sc="http://sw.deri.org/2007/07/sitemapextension/scschema.xsd">
   <sc:dataset>
      <sc:datasetLabel>Product Catalog for Example.org</sc:datasetLabel>
       <sc:dataDumpLocation>http://example.org/cataloguedump.rdf</sc:dataDumpLo-
cation>
       <sc:linkedDataPrefix>http://example.org/products/</sc:linkedDataPrefix>
       <changefreq>monthly</changefreq>
 </sc:dataset>
  <url>...
    </url>
    <url>...
    </url>
```

```
  .....
</urlset>
```

3.4. A complete example

The following extends the previous with statements which state that the same dataset is available in fragmented form. Also a URI is defined for the dataset.

This URI is a URL minted in the same host space (http://example.org/aboutcatalog.rdf#catalog). While there is no need for such a URI to resolve to a HTTP retrievable document (in this case to http://example.org/aboutcatalog.rdf) the Semantic Web Linked Data paradigm strongly advises to do so. In so doing, the concept URI resolves to a description of the concept itself (in this case resolving the URI would give the aboutcatalog.rdf file which could, and should in fact, contain statements about http://example.org/aboutcatalog.rdf#catalog). Finally, the address of a SPARQL endpoint for the dataset is also stated.

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
         http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"
         xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
        xmlns:sc="http://sw.deri.org/2007/07/sitemapextension/scschema.xsd">
<sc:dataset>
   <sc:datasetLabel>Product Catalog for Example.org</sc:datasetLabel>
   <sc:datasetURI>http://example.org/aboutcatalog.rdf#catalog</sc:datasetURI>
<sc:dataDumpLocation>http://example.org/cataloguedump.rdf</sc:dataDumpLocation>
 <sc:dataFragmentDumpLocation>http://example.org/cataloguedump_part1.rdf
</sc:dataFragmentDumpLocation>
 <sc:dataFragmentDumpLocation>http://example.org/cataloguedump_part2.rdf
</sc:dataFragmentDumpLocation>

<sc:linkedDataPrefix>http://example.org/products/</sc:linkedDataPrefix>
 <sc:sparqlEndPoint>http://example.org/queryengine/sparql</sc:sparqlEndPoint>
      <changefreq>monthly</changefreq>
 </sc:dataset>


</urlset>
```

3: Data on different hosts with priorities

In this example the website wishes to state it has data served as an RDF Dump and also by two SPARQL endpoints: one on the same website and one on another. Please note that for a statement involving an external site, the same statement should also be placed also in the Sitemap of the other site.

The priority values in the mirrored data dumps and SPARQL endpoints are set so that the backup dataset location will not be used unless the primary location is not responding. On the other hand, the host wishes that the secondary SPARQL service be used with a ratio of 1/(10+default) = 1/11 with respect to the main one.

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
         http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"
         xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
        xmlns:sc="http://sw.deri.org/2007/07/sitemapextension/scschema.xsd">

<sc:dataset>
   <sc:datasetLabel>Product Catalog for Example.org</sc:datasetLabel>
```

```
   <sc:datasetURI>http://example.org/aboutcatalog.rdf#catalog</sc:datasetURI>
   <sc:dataDumpLocation>http://example.org/cataloguedump.rdf</sc:dataDumpLocation>
   <sc:dataDumpLocation
    priority="0">http://backup.example.org/cataloguedump.rdf</sc:dataDumpLocation>
   <sc:sparqlEndPoint priority="10">http://example.org/queryengine/sparql
   </sc:sparqlEndPoint>
   <sc:sparqlEndPoint>http://secondary.example.org/queryengine/sparql
   </sc:sparqlEndPoint>
         <changefreq>monthly</changefreq>
</sc:dataset>
</urlset>
```

4 Schema, Examples Online, implementation and known uses

A schema for this extention is available at: http://sw.deri.org/2007/07/sitemapextension/scschema.xsd. The schema is to be considered "non-normative" with respect to the specification document.

Online examples of the semantic sitemap in action on large semantic web databases online include, at the time of the writing:

UNIPROT - http://purl.uniprot.org/sitemap.xml,
U.S. Census - http://www.rdfabout.com/sitemap.xml

An implementation of a Java library to support the retrieval and processing of sitemaps using this extension is made available for download from the specification page itself.

From the data consumer side, we report that this extension is currently parsed and used by the Sindice semantic web indexing service[1] and by the latest version of the DBin semantic web client[2].

3.5. Conclusions

Thanks to the best practices developed by the Linking Open Data on the Semantic Web initiatives, a great number of databases have been made available on the semantic web today. While resolving the single URI/URLs seems a sensible thing to do for many use cases, this paradigm is not itself well suited for crawling and indexing large amounts of such data. The Semantic Sitemap extension we have presented in this paper efficiently addresses not only this use cases but also others that are likely to arise in Semantic Web clients.

We conclude by saying that the reaction by data producers to this proposal has so far been very positive: the deployment of a Semantic Sitemap is a very simple task and is perceived as effective in rationalizing the consumption of resources for serving large quantity of semantically structured data.

3.6. Acknowledgments

Gratitude goes to the following persons for contributing to this document or to the discussion. Chris Bizer (Free University Berlin), Richard Cyganiak (Free University Berlin), Andreas Harth (DERI Galway), Aidan Hogan (DERI Galway), Leandro

1 Http://sindice.com
2 Http://dbin.org

Lopez (independent), Stefano Mazzocchi (SIMILE- MIT), Christian Morbidoni (SEMEDIA - Universita' Politecnica delle Marche), Michele Nucci (SEMEDIA - Universita' Politecnica delle Marche), Eyal Oren (DERI Galway), Leo Sauermann (DFKI)

References

[1] Linking Open Data on the Semantic Web - http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
[2] The Sitemap protocol - http://www.sitemaps.org/protocol.php
[3] The Sitemap protocol: robot.txt extension - http://www.sitemaps.org/protocol.php#submit_robots
[4] RDF Semantics - http://www.w3.org/TR/rdf-mt/
[5] P. Stickler, "Concise Bounded Description", W3C Member Submission http://www.w3.org/Submission/CBD/
[6] G. Tummarello, C. Morbidoni, P. Puliti, F. Piazza, "Signing individual fragments of an RDF graph", 14th International World Wide Web Conference WWW2005, Poster track, May 2005, Chiba, Japan
[7] J. Carrol, P. Stickler "TriX : RDF Triples in XML", Technical report HPL-2004-56 http://www.hpl.hp.com/techreports/2004/HPL-2004-56.pdf
[8] C. Bizer, R. Cyganiak, "The TriG syntax" http://www.wiwiss.fu-berlin.de/suhl/bizer/TriG/Spec/
[9] A. Harth, SWSE dumps in NQUADS, data files and format explanation http://sw.deri.org/2005/04/semwebbase/