Review of semantic enablement techniques used in geospatial and semantic standards for legacy and opportunistic mashups

Laurent Lefort
CSIRO ICT Centre
GPO Box 664 Canberra, ACT 2601, Australia
laurent.lefort@csiro.au

Abstract

Networks of sensors are increasingly used to monitor essential environmental variables for biodiversity, water, and climate change research. Such multidisciplinary scientific projects require more flexible ways to publish and aggregate sensor observations from different networks as mashable web resources. Semantically-enabled and linkable descriptions of sensors and sensors services can simplify the integration of legacy backend sensor web services and make it easier for mashup developers to opportunistically combine these resources.

This paper reviews linking and annotation techniques applicable to the development of geospatial mashups services. It describes how approaches based on RDFa could supersede existing techniques for the semantic annotation of RESTful services. It highlights specific issues linked to the hybrid nature of mashups combining solutions based on XML, RDF and HTML standards and the failure risks attached to such multi-standards knowledge systems. It points out the pending technical issues, especially the ones where more coherent approaches are needed e.g. the upgrade of existing standards like XLink and SAWSDL or the integration of validation tools developed for each family of standards.

Keywords: semantic web, sensor web, geospatial standards, mashup, XLink, RDFa.

1 Introduction

As networks of sensors are increasingly used to monitor essential environmental variables for biodiversity, water, and climate change research, we need innovative approaches to simplify the integration of sensor observations from different networks into mashable web resources. Pairing geospatial standards developed by the Open Geospatial Consortium (OGC) and semantic web standards developed by the World Wide Web Consortium (W3C) can foster new approaches for applications that are not (or not yet) clear candidates as web standards.

Apart from the Keyhole Markup Language (KML), most OGC standards have been developed prior to the appearance of modern mashup techniques. The W3C Semantic Sensor Network Incubator Activity[1] (SSN-XG) develops semantic descriptions of sensors and sensors services to semantically enable services based on Sensor Web Enablement (SWE) standards like the Sensor Markup Language (SensorML) and the Sensor Observation Service (SOS). This review focuses on linking and annotation techniques which can support the discovery and composition of these services and their integration into web mashups (Le Phuoc and Hauswirth 2009).

This paper is structured in four main parts. Section 2 defines legacy and opportunistic mashups and how they can be combined in a multi-layered integration scheme. It also discusses how this scheme may evolve with the introduction of new mashup engines and technologies based on existing and actively developed semantic web standards. Section 3 reviews the XML, HTML and RDF-based linking, and annotation methods and their applicability in this context. Two practical examples are used in Section 4 to compare the available approaches and to identify the innovative features of RDFa which are applicable to the semantic annotation of RESTful services. The discussion in Section 5 identifies failure risks which are specific to knowledge systems including sources of interfaces problems likely to occur in such multi-standard setups. It also points out the pending technical issues, especially the ones where more coherent approaches are needed e.g. the upgrade of existing standards like XLink and SAWSDL or the integration of validation tools developed for each family of standards.

2 Typology of mashups

2.1 Multi-layered mashup framework

The Model for layered integration tools proposed by Gamble and Gamble (2008) groups pre-Web, Web 1.0 and Web 2.0 technologies into three separate integration zones with decreasing level of integration effort and increasing readiness for opportunistic development. In this framework, *legacy mashups* require more work because the integration of pre-Web and Web 1.0 resources generally requires the development of custom-made wrappers. First generation mashup engines such as Damia, Yahoo Pipes, Popfly, or Google Mashup Editor (Di Lorenzo et al. 2009, Koschmider et al. 2009) enable the creation of *opportunistic mashups* based on the most popular Web 2.0 service API (Application Programmable Interfaces). These mashup engines have been very successful even if they are often tied to proprietary APIs or platforms.

[1] http://www.w3.org/2005/Incubator/ssn/

Figure 1 illustrates the layered model defined by Gamble and Gamble (2008) where the two types of integration approaches cohabit. Legacy services are integrated in the first integration layer as legacy mashups. The resulting services are exploited in the second integration layer with more lightweight mashup methods.

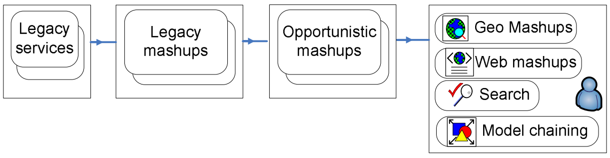

Figure 1: Multi-layered mashups

2.2 Non-semantic mashups

Geospatial and Sensor web service-oriented platforms can combine Web 2.0 technologies like Ajax to global geospatial data resources like Google to enable the online publication of geospatial and sensor datasets and services. Mashable APIs are now available for geospatial and sensor web resources like Google Maps[2] or Pachube[3] and from popular GIS tools like ArcGIS[4].

Figure 2 presents a simple example of multi-layered geospatial mashup. ArcGIS can be used to integrate data from OGC web services and expose it through proprietary Javascript APIs5 which can be further mashed up in Web 2.0 tools like Google maps.

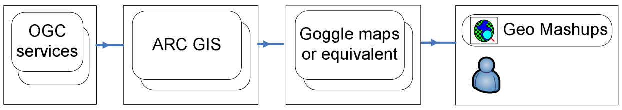

Figure 2: A simple geo-mashup based on Arc GIS

2.3 Semantic mashups

The lack of extensibility of existing APIs is driving the development of the next generation of *semantic mashup* engines based on semantic web standards developed by W3C. SAWSDL (Kopecký et al. 2007) uses semantic descriptions to enable the composition of web services for *legacy semantic mashups*. These rich semantic descriptions help to compose geospatial services (Lemmens et al. 2007, Vaccari et al. 2009). Custom-made operators are often developed to transform the data from XML to RDF (Henson et al. 2009) and to better manage its provenance (Sahoo et al. 2008).

Opportunistic semantic mashups generally use RDF (triple stores) resources applying the Linking Open Data conventions (Bizer et al. 2007) via standard APIs based on SPARQL (Prud'hommeaux et al. 2008, Clark et al. 2008) or via proprietary query languages offered by Web-based development environments such as Metaweb ACRE[6] or Yahoo Pipes[7] designed to offer the possibility for end users to develop and share their mashups.

Opportunistic semantic mashups can also source data from HTML pages, especially from RDFa (Adida et al. 2008) snippets embedded in web pages. RDFa, originally designed as an extension of XHTML2 and now ported[8] to HTML5[9] is a hybrid method devised to sprinkle RDF data or metadata in a web page and make it available for further content aggregation down the track, e.g. at the level of search engines (Benjamins et al. 2008). Search platforms like Google and Yahoo SearchMonkey[10] exploit RDFa content to improve search results and use it in search engine results as richer snippets (Goel et al. 2009).

DERI Pipes (Le Phuoc et al. 2009), MashQL (Jarrar and Dikaiakos 2009) and TopQuadrant's SparqlMotion[11] are three examples of semantic mashup engines which allow end users to chain (or pipe) simple URI-based data integration operators. DERI Pipes users can fetch data from XML using XQuery, from RDF using SPARQL and extract embedded RDFa and microformat data from HTML using purpose-built operators. Figure 3 presents a semantic mashup architecture implemented by Le Phuoc and Hauswirth (2009) which combines a semantic wrapper for Sensor Observation Service similar to SemSOS (Henson et al. 2009) with a SensorMasher application based on DERI pipes. In this implementation, SPARQL is used to query data from the sensor ontologies and from the sensor data streams.

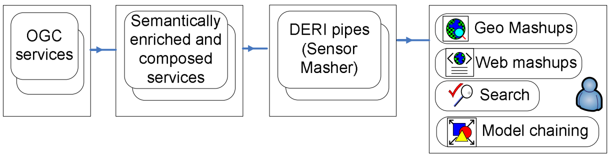

Figure 3: A multi-layered semantic mashup

2.4 Semantic enablement methods

There are four basic *semantic enablement* methods for legacy and opportunistic mashups applicable at different levels of the multi-layered scheme described in Figure 1:

- Inclusion of remote RDF (or SKOS/OWL) resources in XML using XLink,
- Annotation of web services with SAWSDL,
- Annotation of RESTful web services using hRESTs (or SA-REST, MicroWSMO),
- Inclusion of remote RDF (or SKOS/OWL) resources in HTML using RDFa.

[2] http://code.google.com/apis/maps/

[3] http://www.pachube.com/

[4] http://www.esri.com/software/arcgis/

[5] http://www.esri.com/javascript

[6] http://www.freebase.com/apps/

[7] http://pipes.yahoo.com/

[8] http://dev.w3.org/html5/rdfa/rdfa-module.html

[9] http://www.w3.org/TR/html5/

[10] http://developer.yahoo.com/searchmonkey/

[11] http://www.topquadrant.com/products/SPARQLMotion.html

The next section reviews the basic XML, HTML and RDF-based linking and annotation standards and their relevance to the four semantic enablement methods defined above. For this purpose, the following terminology is used. *Mashable content* corresponds to any type of remotely managed resources which can be used in a mashup. *Links* specifies the inclusion of remotely managed resources. *Semantic annotations* define how to map service capabilities to semantic definitions to enable the discovery or composition of web services. The transition from XML-based services to RDF-based services is called a *lifting* operation (Farrell and Lausen 2007) and the inverse one, from RDF to XML is called a *lowering* operation.

3 Linking and annotation methods

3.1 Handling mashable content with javascript

Mashable content can be extracted from XML, RDF (OWL) and HTML resources, and from RDFa snippets included in web pages. Different javascript libraries (see Table 1) can be used to process data sourced from different origins.

Mashed up content	Javascript library
XML resource	JQuery http://jquery.com/
RDF resource	JSON http://www.json.org/ used to serialise SPARQL results http://www.w3.org/TR/rdf-sparql-json-res/
OWL resource	JOWL (JQuery extension) http://jowl.ontologyonline.org/
HTML snippet	JQuery http://jquery.com/
RDFa snippet	rdfQuery (JQuery extension) http://code.google.com/p/rdfquery
Microformat snippets	A custom-made javascript library is needed for each different microformat

Table 1: Types of mashable content

Interest for RDFa is growing fast because the prospect for being able to extend documents without having recourse to standards organisations is enormous and because the addition of RDFa content to already published web pages can be done without forcing the web site designers to change the look of their sites.

Microformats are available for a number of specific applications with various levels of popularity and support. The HTML5 Microdata proposal is an attempt to offer a generic alternative to the existing Microformat coding conventions. It is not reviewed here because this set of requirements (Hickson 2009) can be considered as a subset of the requirements addressed by RDFa.

3.2 Linking methods

Links are defined here as mechanisms used to extend available content from any type of resources with information sourced from remotely managed content (type or instance). Links are possible between two documents of the same type or between documents of different types. Table 2 lists the techniques used to link documents to each other on a range of use cases which can occur in mashups.

Linked resource type	Linking method	Type of link
XML	XLink	XML to XML
XML	XLink	XML to URNs
XML	XLink	XML to RDF
XML	RDFa	XML to RDF
RDF	OWL mapping properties or weaker alternatives like umbel:isLike	RDF to RDF
SKOS	SKOS mapping properties	SKOS to SKOS
OWL	OWL mapping properties	OWL to OWL
HTML	Microformats	HTML to "data"
HTML	RDFa or Common Tag	HTML to RDF

Table 2: Linking methods

The XML Linking language or XLink (DeRose et al. 2001) is a W3C standard which allows the creation of links between XML resources. It is commonly used in OGC standards to include references to external vocabularies managed with URNs.

To link RDF-based vocabularies, ontologies or Linking Open Datasets (LOD) content, the most common approach is to use the basic relationships defined in the Web Ontology Language OWL: owl:sameAs, owl:equivalentClass, owl:equivalentProperty although for plain LOD content, weaker alternatives may be preferable like the one proposed by the UMBEL[12] developers. SKOS[13] offers a richer range of properties (exactMatch, closeMatch, broaderMatch, narrowerMatch) to specify the relationships between concepts.

3.3 Semantic annotation methods

Different semantic annotations methods are needed for WSDL web services and RESTful web services.

Upgrading WSDL web services into semantically enabled services can be done with the help of SAWSDL (Kopecký et al. 2007), now a W3C Recommendation (Farrell and Lausen 2007). The SAWSDL specification has three main features:

- Semantic definitions (in a RDF-based format like OWL) may be included in the WSDL file.
- A small set of elements and attributes can be added in different parts of the WSDL service description to create links from XML schemas elements and attributes to their *model references* which are semantic definitions.
- And finally, additional attributes can be used to associate a schema type or element with a mapping script describing lifting transformation from XML to RDF and lowering transformation from RDF to XML.

Upgrading REST web services into semantically enabled services requires different tools because the service

[12] http://www.umbel.org/

[13] http://www.w3.org/TR/skos-reference/

declaration is generally made within a HTML web page and does not use an XML-based description format. SA-REST (Lathem et al. 2007, Sheth et al. 2007) and MicroWSMO (Kopecký et al. 2009) are two related efforts which use the same semantic annotation microformat, hRESTs (Kopecký 2008). The SA-REST approach is more closely related to the SAWSDL standard while MicroWSMO uses a different ontology: WSMO-Lite.

3.4 Types of lifting operations

GRDDL (Connolly 2007) defines the syntax to embed the reference to a lifting script in any type of well-formed XML format. The file to which the GRDDL annotation has been added is used as the input of the specified lifting operation. The RDF output depends on the location of the GRDDL markup. If the corresponding transformation is available, any HTML files containing microformat-based annotations can use this mechanism to be transformed into RDF.

SAWSDL, SA-REST and MicroWSMO also require the development of custom-made scripts. A major difference is that these scripts specify how to process the XML data manipulated by the service, not the content of the file containing the annotations.

RDFa defines a generic lifting mechanism to transform the annotations included in an HTML file into RDF. In this case, there is no need for user-developed scripts.

Lifting scripts may use languages like XSL transformations[14] (XSLT) or XQuery[15]. Lowering scripts may use hybrid approaches like XSPARQL (Akthar et al. 2008), a W3C Member Submission[16] which mixes XQuery and SPARQL. RDFa users can also use alternative implementations such as the ones available in javascript (Table 1).

4 Comparison of key linking methods

A short summary of the key features of each method is provided below. A more direct comparison is also done on two examples to complete this analysis in relation to two critical issues:

- Choice between the hRESTs microformat and RDFa for the semantic annotations of REST-based services and consistency of these approaches with existing ones (SAWSDL),
- Choice between XLink and RDFa as the linking technique used for XML content.

The first example focuses on semantic annotation requirements to guide the future work on REST services and also bridge the gap between these new methods and what can currently be used for WSDL.

The second example illustrates the differences between the XML-friendly solution based on XLink and the alternative approach based on RDFa.

[14] http://www.w3.org/TR/xslt20/

[15] http://www.w3.org/TR/xquery/

[16] http://www.w3.org/Submission/2009/01/

4.1 Key attributes for each approach

RDFa: for the purpose of this review, we use the W3C Recommendation version of RDFa (Adida et al. 2008).

Attribute	Description	Intended RDF
about	The identification of the resource (to state what the data is about)	rdf:about of domain resource
typeof	RDF type(s) to associate with a resource	rdf:about of class of a resource
href	Partner resource of a relationship ('resource object')	rdf:about of range resource
property	Relationship between a subject and some literal text ('predicate')	rdf:about of datatype property
rel	Relationship between two resources ('predicate')	rdf:about of object property
rev	Reverse relationship between two resources ('predicate')	rdf:about of (inverse) object property
src	Base resource of a relationship when the resource is embedded 'resource object')	rdf:about of domain resource
resource	Partner resource of a relationship that is not intended to be 'clickable' ('object')	rdf:about of range resource
datatype	Datatype of a property	XML type range of datatype property
content	Machine-readable content ('plain literal object')	Value for datatype property

Table 3: RDFa attributes

In RDFa, the about and resource attributes plays the role of rdf:about and rdf:resource attributes in RDF. They can be encoded as compact URIs or CURIES (Birbeck and McCarron 2009), a syntax inspired by the prefix management conventions used in SPARQL. The content of a datatype property can be included as an extra attribute (content) or retrieved from the element content.

hRESTs: hRESTs focuses on the capture of mapping information between the service description and a reference ontology. The additional information is provided through the coding of the lifting script applicable to the service outputs. The hRESTs microformat specification used here is the one published by Kopecký et al. (2009) and the associated examples.

Attribute	Description	Intended RDF
class	Type of XML or WSDL element (service, operation, address, method, input, output, label)	rdf:about of class of domain resource
href next to rel="model"	association between a WSDL or XML schema component and a concept in some semantic model	rdf:about of range class = modelReference
href next to rel="lifting"	Lifting script URL	N/A
href next to rel="lowering"	Lowering script URL	N/A
id	Locally declared id of WSDL element (to be combined with the document URL)	rdf:about of domain resource

Table 4: hRESTs Microformat attributes

The HRESTs microformat mandates the use of blocks with class elements in a rigid parent-child hierarchy (e.g. service contains operation) which will be implicitly transposed in the resulting RDF file.

XLink: For the purpose of this review, we will use the XLink guidelines documented for the Geography Markup Language standard (Portele 2007) rather than the original W3C specification Xlink (DeRose et al. 2001). Table 5 summarises the attributes defined by this specification.

Attribute	Description	Intended RDF
xlink:href	Identifier of the resource which is the target of the association, given as a URI	rdf:about of range resource
xlink:role	Nature of the target resource, given as a URI	rdf:about of class of range resource
xlink:arcrole	Role or purpose of the target resource in relation to the present resource, given as a URI	rdf:about of object property linking domain element to range resource
xlink:title	Text describing the association or the target resource	rdfs:comment

Table 5: XLink attributes

4.2 Feature comparison: hRESTs and RDFa

Kopecký et al. (2009) also specify how hRESTs can be expressed in RDFa. Table 6 is based on this input. The main difference is that hRESTs in RDFa allows the user to specify the target ontology through the definitions of the typeof, rel, property and datatype attributes.

RDF mapping	hRESTs in Microformats	hRESTs in RDFa
Domain instance	id (URL-prefixed)	about
Domain class	class (closed list)	typeof
Object property	ref="model"	rel
Inverse object property		rev
Range instance		href or resource
rdf:about of range class	href	typeof
Datatype property		property
Datatype property type		datatype
Range value		content or element content

Table 6: Comparison of RDFa and hRESTs

4.3 Feature comparison: XLink and RDFa

The direct comparison done in Table 7 can help to locate the major difference between XLink and RDFa which is that the two specifications cover different types of RDF triples:

- XLink: predicate (role) and object (href) for object properties
- RDFa: subject (about), predicate (rel) and object (href) for object properties and subject (about), predicate (property) and object (content or element content) for datatype properties

RDF mapping	Xlink	RDFa
Domain instance		about or src
Domain class		typeof
Object property	arc role	rel
Inverse object property		rev
Range instance	href	href or resource
Range class	role	typeof
Datatype property		property
Datatype property type	role	datatype
Range value		content or element content

Table 7: Comparison of XLink and RDFa

4.4 Examples of semantic annotations

The National Digital Forecast Database is a web service[17] developed by the U.S. National Weather Service to test the Digital Weather Markup Language (DWML). This forecast service (see also Al-Muhammed et al. 2007) is used here because it is simultaneously implemented as a WSDL service and as a REST service. Figure 4 shows an example of SAWSDL annotation in the WSDL file.

[17] http://www.nws.noaa.gov/ndfd/technical.htm

```
<wsdl:part name="endTime"
sawsdl:modelReference="http://sweet.jpl.nasa.gov/2.0/time.owl#End
" type="xsd:dateTime "/>
```

Figure 4: WSDL file with SAWSDL annotations (extract)

Table 8 lists the concepts defined in the SWEET 2.0 ontologies[18] which can be used as model references for the message parts of the NFDGen operation. Model references for service parameters like the product type (Time series or "glance") and the output type are specific to DWML and are not available in SWEET 2.0.

wsdl:part	SWEET 2 ontologies	Class
latitude	spaceCoordinates.owl	Latitude
longitude	spaceCoordinates.owl	Longitude
startTime	time.owl	Start
endTime	time.owl	End

Table 8: Types of mashable content

Many REST services are only documented through a web page. This is why semantic annotation methods like SA-REST or MicroWSMO can use any type of web page describing a service. The two options are to annotate the HTML page (or form) used to run the service (Figure 5) or a "WSDL-inspired" documentation page (Figure 6).

Figure 5: HTML form for a REST service (simplified)

In the first case, the HTML form can host the semantic annotations for the input data and for other elements used to run the service. An advantage of this approach is that the annotated form (Figure 5) can still be used to test that the service works. But additional content is required to annotate the output data (representations and faults).

The alternative is to have a documentation page in HTML which describes hierarchically the service, its operations, and the input and output format for each operation. This style of web page is comparable to what could be generated out of a WSDL file (when such a file is available). Figure 6 illustrates this approach with an HTML file generated out of (an extract of) the WSDL file with an existing XSL transformation[19].

Figure 6: HTML service description derived from WSDL

The two following examples present two types of annotations: hRESTs Microformat (Figure 7), and RDFa (Figure 8) applicable to the HTML form.

The hRESTs example (Figure 7) only includes semantic references for the sawsdl:modelReference attributes in SAWSDL. While the hRESTs solution may seem easier to use, it also requires extra effort for the end user to learn how the mapping between the class annotations used in the microformat (operation, action, input …) and the ontology used for the generated RDF content. This mapping may depend on the hRESTs toolset and on the availability of custom-made lifting and lowering scripts.

```
<FORM method="get" name="NDFDgenForm"
action="http://www.weather.gov/forecasts/xml/sample_products/bro
wser_interface/ndfdXMLclient.php"
class="operation">
  <DIV id="GmlTimePeriod" style="display: block; ">
    <P>Valid Time Range ?</P>
    <OL>
      <TABLE border="1" cellpadding="4" width="60%">
        <TBODY class="input">
          <TR rel="model"
href="http://sweet.jpl.nasa.gov/2.0/time.owl#End">
           <TD><span class="label">End Time</span>: <INPUT
type="text" name="endTime">
size="40" maxlength="80" value=""
onfocus="document.NDFDgenForm.endTime.value = &#39;2010-
01-01T00:00:00&#39;;">2010-01-01T00:00:00</TD>
        </TR></TBODY></TABLE>
    </OL> </DIV>
</FORM>
```

Figure 7: hRESTs example

18 http://sweet.jpl.nasa.gov/ontology/

19 http://tomi.vanek.sk/index.php?page=wsdl-viewer

The RDFa example (Figure 8) includes semantic references defining the type of annotations (e.g. sarest:operation). This approach gives more control to the end user for the choice of the service ontology and simplifies the task for the programming of tools which interprets the annotations. The RDFa specification (Adiba et al. 2008) defines processing rules which helps to combine these two types of semantic references seamlessly.

```
<FORM method="get" name="NDFDgenForm"
typeof="[sarest:action]"
action="http://www.weather.gov/forecasts/xml/sample_products/browser_interface/ndfdXMLclient.php"
about="http://www.weather.gov/forecasts/xml/sample_products/browser_interface/ndfdXMLclient.php">>
 <DIV id="GmlTimePeriod" style="display: block; ">
   <P>Valid Time Range ?</P>
   <OL>
     <TABLE border="1" cellpadding="4" width="60%">
       <TBODY rel="[sarest:input]">
         <TR
about="http://www.weather.gov/forecasts/xml/sample_products/browser_interface/ndfdXML.htm#GmlTimePeriod.endTime"
typeOf="[sweet20:End]" >
          <TD><span property="[rdfs:label]">End Time</span>:
<INPUT type="text" name="endTime">
size="40" maxlength="80" value=""
onfocus="document.NDFDgenForm.endTime.value = &#39;2010-01-01T00:00:00&#39;;">2010-01-01T00:00:00</TD>
       </TR>
     </TBODY></TABLE>
   </OL>  </DIV>
</FORM>
```

Figure 8: hRESTs in RDFa example

4.5 Examples of semantic links

OGC standards like GML (Portele 2007) define the use of XLink to add annotations in XML files. These annotations can point to extra sources of information (e.g. a file) or to Uniform Resource Name (URN).

The first use case is described in the GML specification as "composition by inclusion of remote resources": in this case, the XLink annotation use the xlink:href attribute to reference an external file containing additional data (Figure 9).

```
<component name="weatherStation"
xlink:href="http://vast.uah.edu/downloads/sensorML/v1.0/examples/sensors/DavisWeather/DavisMonitorII-WeatherStation.xml"/>
```

Figure 9: XLink used in SensorML to include extra data

Transposing this example to RDFa requires the inclusion of an annotation which identifies the concept in a repository of sensor descriptions with an about attribute: the URI would then point to an individual or instance (Figure 10) providing access to the data to be included.

```
<component
name="weatherStation"about="http://vast.uah.edu/downloads/sensorML/v1.0/examples/sensors/DavisWeather/DavisMonitorII-WeatherStation"/>
```

Figure 10: RDFa example: additional data

The second use case corresponds to the inclusion of a "model reference to an ontological description". In this case, the XLink annotation use the xlink:arcrole attribute to define the type of the referenced object (Figure 11). The definition attribute in the SWE schemas and the descriptionReference in the GML schemas are scoped for this particular usage.

```
<member xlink:arcrole="urn:ogc:def:process:OGC:SensorInstance">
```

Figure 11: XLink used in SensorML to define a type

In RDFa, the typeof attribute can be used for the same purpose (Figure 12). A URN pointing to a type definition (or class) is then used

```
<member typeof=="urn:ogc:def:process:OGC:SensorInstance">
```

Figure 12: RDFa example: reference to ontological def.

The example above shows that the current use of Xlink in OGC schemas can be mirrored in RDFa.

In our generalised mashup approach, the semantic annotations should be exploitable by generic or user-defined lifting operators to create the corresponding RDF statements. When this RDF is lowered back into XML, there is a risk of losing some of the information previously available. XLink can be used to maintain some of this lowered content. Table 7 defines the mappings between the two approaches which are possible with the present XLink specification. It also shows that there are other usages which are possible in RDFa but not in the "simple" style of XLink.

5 Directions for future work

5.1 Guidelines for the application of hRESTs

For RESTful services, the format of the HTML content which should be annotated is not specified by the proposed specifications. This is an issue which should be addressed. The form-embedded annotation approach is preferable to the description-based one in general for the part of the description which describes how to run the service, because the annotated form can still be used to test that the service works. For the part of the description which covers the output data (results and error messages), a different approach is required, to be based on an embedded XML schema (this is what WADL does) or on another form of testable content.

5.2 SAWSDL vs. hRESTs in RDFa

The relative complexity and rigidness of the SAWSDL and of the hRESTs Microformat specification contrasts with the flexibility of the approaches based on RDFa (e.g. hRESTs in RDFa), where the choice of the service ontology can be made by the end user without requiring any new developments for the lifting of the semantic annotations into semantic web tools.

This extra flexibility is important not just for RESTful services. Further work is required to upgrade SAWSDL so that it can also let the end user select the service ontology they want if they are not satisfied by the definitions brought by the SA-REST or WSMO-Lite ontologies.

5.3 Ontologies for other types of services

Other service description languages like WADL (Hadley 2009) and WSDL 2.0[20] may provide a better basis for RESTful services. The hybrid ontology and rule-base framework proposed by Zhao and Doshi (2009) handles three categories of composable RESTful services to add access and transform resources.

SensorML (Botts and Robin 2007) is an OGC-developed markup language for the description of sensors. It includes a process model which is comparable to the other service ontologies discussed above. The challenge for the W3C Semantic Sensor Network Incubator Activity is to develop an ontology describing sensor services based on SensorML and use it for semantic annotations in a context where the boundary between the application-specific ontologies and the service ontologies and between non-semantic and semantic mashups is harder to define.

5.4 Replacement of custom-made lifting scripts

Any solution requiring the development of custom-made lifting mechanisms should be avoided if alternative approaches based on standards which fully specify this critical step like RDFa are available. The dependency on user-developed transformations for the lifting scripts is one of the factors which have slowed down the adoption of semantic annotation standards for services like SAWSDL and hRESTs/SA-REST/MicroWSMO.

As discussed above, the hRESTs in RDFa format provides a generic approach for the transformation of the semantic annotations into a RDF-based format and it should be possible to develop a similar approach for SAWSDL and to also suppress the requirement to develop custom-made scripts for this purpose.

But, it is not yet possible to automatically derive the lifting script for the second type of lifting operation discussed in 3.4, where the script goal is to process the XML data manipulated by the service and not the file containing the annotations. The MyMobileWeb project (Berrueta et al. 2009) has been looking at RDFa for a similar problem, to describe the bindings to data sources and enable multi-device mobile access to semantically enriched information portals.

5.5 Controlled upgrade of legacy standards

Ad hoc semantic upgrade of legacy standards such as XLink should be monitored closely to minimise the risks of failure caused by problematic extensions by end users.

In many cases, techniques bound to one family of standards (XML) have been later adapted to a different context without any assurance that the new usage respects the original intent of the specification. Hybrid ad hoc approaches may also import conflicting or ambiguous definitions from different standard families.

Some parts of SensorML uses XLink annotations to embed “model reference to an ontological description” in the sensor description (e.g. swe:phenomenon). These use cases are a possible source of confusion because they answer to requirements which can potentially be better addressed through new approaches based on semantic web technologies.

For example, to handle all the annotations requirements identified for RDFa in an XML context, a simple approach would be to add a new “style” to XLink for RDFa as an extension to the current XLink specification. For organisations like OGC who already use XLink and maintain a large number of XML schemas, this approach would have two advantages.

- To limit the impact on existing schemas to changes in the XLink schema,
- To provide a mechanism to isolate semantic XLink snippets from normal ones.

This upgrade of XLink should not be done without a careful consideration of the present usage of XLink in OGC standards and also in other standards like SVG[21].

5.6 Failure risk analysis

Combining legacy and opportunistic mashups will require robust and mashable validation tools to prevent and diagnose failures. Opportunistic mashups depends on external resources which may disappear or evolve without notice, especially mashable services and semantic resources, so the risks of failure are greater and more diverse than in other environments.

In a multi-layered mashup environment, it is important to support validation at every possible step of integration and to leverage the validation methods which are specific to each family of standards: XML, HTML and RDF individually. In this context, it is very important to check the availability of validators and their ability to check the content (markup validators) as well as the added annotations or links to remote resources and also the flexibility and robustness of these tools.

The Unicorn[22] (Universal Conformance Observation and Report Notation) project at the W3C is a *validator mashup* combining a HTML validator, a CSS validator and a HTML link checker. Extending this approach to the other families of the W3C[23] and OGC standards used in the type of mashups discussed above would be very useful.

6 Conclusion

There are multiple semantic enablement techniques which can be used in geospatial and semantic standards for legacy and opportunistic mashups. For the insertion of semantics links in XML content formatted according to OGC standards, the less disruptive approach identified in this review may be to add a new style to the existing XLink specification transposing all the RDFa attributes and processing rules defined for the HTML context.

The hRESTs-in-RDFa annotation format is preferred for the annotation of RESTful services. The arguments

[20] http://www.w3.org/TR/wsdl20/

[21] http://www.w3.org/Graphics/SVG/

[22] http://www.w3.org/QA/Tools/Unicorn/

[23] W3C specifications and validators are listed in http://www.w3.org/QA/TheMatrix

formerly raised (Graf 2007) to prefer Microformats to RDFa to add semantic annotations or links to HTML have been invalidated by the W3C decision to make RDFa available in HTML5. The analysis presented above shows that solutions based on Microformats prevent the implementation of generic lifting services with scripting languages such as XSL Transformations, XQuery or XSPARQL or with javascript libraries like rdfQuery which plays an essential role in opportunistic mashups.

The SAWSDL specification should also be upgraded to offer the same possibility for the user to select the service ontology.

Finally, in complex mashups, the risk of failure is greater and the validation methods are different for standards belonging to the XML, HTML and RDF families. There should be a limited number of methods to combine these standards together to lower the cost of development of new markup validators and link checkers. If possible, these new validation services should also be mashable to simplify the creation of more integrated validation services.

7 References

Adida, B., Birbeck, M., McCarron, S. and Pemberton, S. (2008): RDFa in XHTML: Syntax and Processing A collection of attributes and processing rules for extending XHTML to support RDF W3C Recommendation 14 October 2008, Available from http://www.w3.org/TR/rdfa-syntax, Accessed 17 Sep 2009.

Akthar, W., Kopecky, J., Krennwallner, T. and Polleres, A. (2008): XSPARQL: Traveling between the XML and RDF worlds – and avoiding the XSLT pilgrimage. *Proc. of 5th European Semantic Web Conference, ESWC 2008,* Tenerife, Spain, LNCS **5021**:432-447. Springer.

Al-Muhammed, M. J., Embley, D. W.,Liddle, S. W., and Tijerino, Y. A, (2007): Bringing Web Principles to Services: Ontology-BasedWeb Services. *Proc. of IEEE Congress on Services (Services 2007)*, 73-80.

Benjamins, V.R., Davies, J., Baeza-Yates, R., Mika, P. Zaragoza, H., Greaves, M., Gómez-Pérez, J.M., Contreras, J., Domingue, J. and Fensel D. (2008): Near-Term Prospects for Semantic Technologies, *IEEE Intelligent Systems*, **23**(1):76-88, Jan./Feb. 2008.

Berrueta, D., Polo, L., Fernández, S., Cantera, J. M. and Jiménez M. (2009): MyMobileWeb Deliverable D.5.4.1 Semantic extensions for IDEAL, 20 February, 2009, Available from http://forge.morfeo-project.org/wiki_en/index.php/Semantic_annotations_for_IDEAL, Accessed 17 Sep 2009

Birbeck, M. and McCarron, S. (2009): CURIE Syntax 1.0 A syntax for expressing Compact URIs W3C Candidate Recommendation 16 January 2009, Available from http://www.w3.org/TR/curie, Accessed 17 Sep 2009

Bizer, C., Heath, T., Ayers, D. and Raimond, Y. (2007): Interlinking open data on the web. *Proc. 4th European Semantic Web Conference*, Innsbruck, Austria.

Botts, M. and Robin, A. (2007): OpenGIS Sensor Model Language (SensorML), OGC 07-000, Open Geospatial Consortium, July 2007, Available from http://www.opengeospatial.org/standards/sensorml, Accessed 17 Sep 2009

Clark, K. G., Feigenbaum, L. and Torres, E. (2008): SPARQL Protocol for RDF W3C Rec. 15 January 2008, Available from http://www.w3.org/TR/rdf-sparql-protocol/ Accessed 17 Sep 2009

Connolly, D. (2007): Gleaning Resource Descriptions from Dialects of Languages (GRDDL). W3C Rec., W3C, 11 September 2007, Available from http://www.w3.org/TR/grddl/, Accessed 17 Sep 2009.

DeRose, S., Maler, E. and Orchard, D. (2001): *XML Linking Language (XLink) Version 1.0* W3C Recommendation 27 June 2001, Available from http://www.w3.org/TR/xlink/, Accessed 17 Sep 2009.

Di Lorenzo, G., Hacid, H., Paik, H., and Benatallah, B. (2009): Data integration in mashups. *SIGMOD Rec.* **38**(1):59-66, Jun 2009.

Farrell, J. and Lausen, H. (2007): *Semantic Annotations for WSDL and XML Schema* W3C Rec., August 2007, Available from http://www.w3.org/TR/sawsdl/, Accessed 17 Sep 2009.

Gamble M. T. and Gamble R. (2008): Monoliths to Mashups: Increasing Opportunistic Assets, *IEEE Software*, **25**(6):71-79, Nov/Dec 2009.

Goel, K., Guha, R. V. and O. Hansson (2009): Introducing Rich Snippets, Google Webmaster Central Blog, 2009-05-12, Google, Available from http://googlewebmastercentral.blogspot.com/, Accessed 17 Sep 2009

Graf, A. (2007): RDFa v.s. Microformats DERI Technical Report 2007-04-10, Available from http://www.sti-innsbruck.at/results/publications/, Accessed 17 Sep 2009

Hadley, M.J. (2009): Web Application Description Language (WADL) Sun Microsystems Inc. February 2, 2009, Available from https://wadl.dev.java.net/, Accessed 17 Sep 2009.

Henson, C. A., Pschorr, J. K., Sheth, A. P. and Thirunarayan K. (2009): SemSOS: Semantic sensor Observation Service. *Proc. of International Symposium on Collaborative Technologies and Systems 2009*, 44-53.

Hickson, I. (2009): HTML5 Draft Standard, 3 November 2009, Available from http://whatwg.org/html5, Accessed 13 Nov 2009.

Jarrar, M. and Dikaiakos, M. D. (2009): A Data Mashup Language For The Data Web. *Proc. of Linked Data on the Web (LDOW2009) Workshop at WWW2009*, Madrid, Spain, ACM.

Kopecký J. (2007): Web Services Description Language (WSDL) Version 2.0: RDF Mapping W3C Working Group Note 26 June 2007, Available from http://www.w3.org/TR/wsdl20-rdf, Accessed 17 Sep 2009

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

Kopecký, J., Gomadam, K. and Vitvar, T. (2008): hRESTs: An HTML Microformat for Describing RESTful Web Services. *Proc. 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, wi-iat **1**:619-625.

Kopecký, J., Vitvar, T., Fensel, D. and Gomadam, K. (2009): D12v0.1 hRESTs & MicroWSMO CMS WG Working Draft 10 March 2009, Available from http://cms-wg.sti2.org/TR/d12/v0.1, Accessed 17 Sep 2009

Koschmider, A., Torres, V. and Pelechano, V. (2009): Elucidating the Mashup Hype: Definition, Challenges, Methodical Guide and Tools for Mashups. Proc. *of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web* in conjunction with the 18th International World Wide Web Conference, Madrid, Spain.

Lathem, J., Gomadam, K. and Sheth, A. (2007): SA-REST and (S)mashups: Adding Semantics to RESTful Services. Proc. *IEEE Int'l Conf. Semantic Computing*, 469–476, IEEE CS Press.

Lemmens, R., de By, R., Gould, M., Wytzisk, A., Granell, C., and van Oosterom, P. (2007): Enhancing Geo-Service Chaining through Deep Service Descriptions, *Transactions in GIS*, **11**(6):849-871, December 2007

Le Phuoc, D., Polleres, A., Morbidoni, C., Hauswirth, M. and Tummarello, G. (2009): Rapid semantic web mashup development through semantic web pipes. *Proc. of the 18th World Wide Web Conference (WWW2009)*, Madrid, Spain.

Le Phuoc, D. and Hauswirth, M. (2009): Linked open data in sensor data mashups. *Proc. of the 2nd International Workshop on Semantic Sensor Networks (SSN09)*, Chantilly, VA, USA, CEUR-WS Proceedings **522**.

Portele C. (2007): OpenGIS® *Geography Markup Language (GML) Encoding Standard* version 3.2.1 OGC 07-036 Open Geospatial Consortium 2007-08-27

Prud'hommeaux, E and Seaborne, A. (2008): SPARQL Query Language for RDF W3C Rec. 15 January 2008, Available from http://www.w3.org/TR/rdf-sparql-query/, Accessed 17 Sep 2009

Sahoo, S. S., Sheth A. and Henson, C. (2008): Semantic Provenance for eScience: Managing the Deluge of Scientific Data, *IEEE Internet Computing*, **12**(4):46-54, July/Aug. 2008.

Sheth, A. P., Gomadam, K. and Lathem, J. (2007), "SA-REST: Semantically Interoperable and Easier-to-Use Services and Mashups," IEEE Internet Computing, **11**(6):91-94, Nov./Dec. 2007.

Vaccari, L. Shvaiko, P. and Marchese, M. (2009): A geo-service semantic integration in Spatial Data Infrastructures. *International Journal of Spatial Data Infrastructures Research (IJSDIR)*, **4**:24-51.

Zhao, H. and Doshi, P. (2009): Toward Automated RESTful Web Service Compositions. *Proc. of the 2009 IEEE International Conference on Web Services (ICWS 2009)*, 189-196, IEEE Computer Society.