

# SemWebbing the London Gazette

*Jeni Tennison*

The Stationery Office (TSO)

[jeni.tennison@tso.co.uk](mailto:jeni.tennison@tso.co.uk)

*John Sheridan*

Office of Public Sector Information (OPSI)

[john.sheridan@opsi.x.gsi.gov.uk](mailto:john.sheridan@opsi.x.gsi.gov.uk)

## Introduction

The web is changing everything - the economy, society, the relationship between the citizen and the state.

To understand these trends better, in 2007 the UK Government commissioned Ed Mayo and Tom Steinberg to conduct The Power of Information review [MAY07]. They set out a simple vision, “that citizens, consumers and government can create, reuse and distribute information in ways that add maximum value.” It is a vision that has won the enthusiastic support of Ministers and is increasingly driving the government’s view of the web.

Governments have an important role as we move towards the web of linked data because they are amongst the largest primary producers of information. The UK Government recognises the transformative capability of the web and its potential to impact on almost every area of our lives. The web provides a platform and semantic technologies provide the standards and capabilities for creating a new type of information infrastructure – one that will underpin ever more businesses and communities. Authoritative sources such as official information carry enormous weight and significance in this environment. A key challenge for the Office of Public Sector Information (OPSI) is to understand how to make public sector information available in ways that enable and maximise its reuse.

Most people are unfamiliar with The London Gazette. “Published by Authority” since 1665, the London Gazette is the UK Government’s Official Journal and Newspaper of Record. It was set up to provide King Charles II with news while he and his court were in Oxford avoiding the Great Plague, and it has been in operation ever since. With a new edition every day, today’s London Gazette contains a huge amount of information — statutory notices about decisions and changes at a local and national level. It is an authoritative source of semi-structured and potentially very rich information. The London Gazette is a natural candidate for the Government to semantically enable, to unlock the reuse potential of the information it contains.

The aims of this project are two fold:

- to address the practical challenges of publishing public sector information in ways which maximise its reuse, and
- to give the London Gazette a new role, as a vehicle for the government to serve semantically enabled official information. Anytime a piece of legislation says that information must be published in the London Gazette, it will be in effect ensuring that information is made publicly available, in a consistent way and in a reusable form.

In the rest of this section, we’ll outline the way the London Gazette is currently published online, and the semantic technologies we decided to use. The rest of the paper will focus on the kinds of reuse we envisioned and the hurdles that we encountered.

## Existing Website

The London Gazette and its partner publications are currently published by the The Stationery Office (TSO) and available online at <http://www.london-gazette.gov.uk/>. This website offers a great deal of functionality to the various users of the London Gazette, including users who wish to place notices, browse through recent notices or search for historical notices.

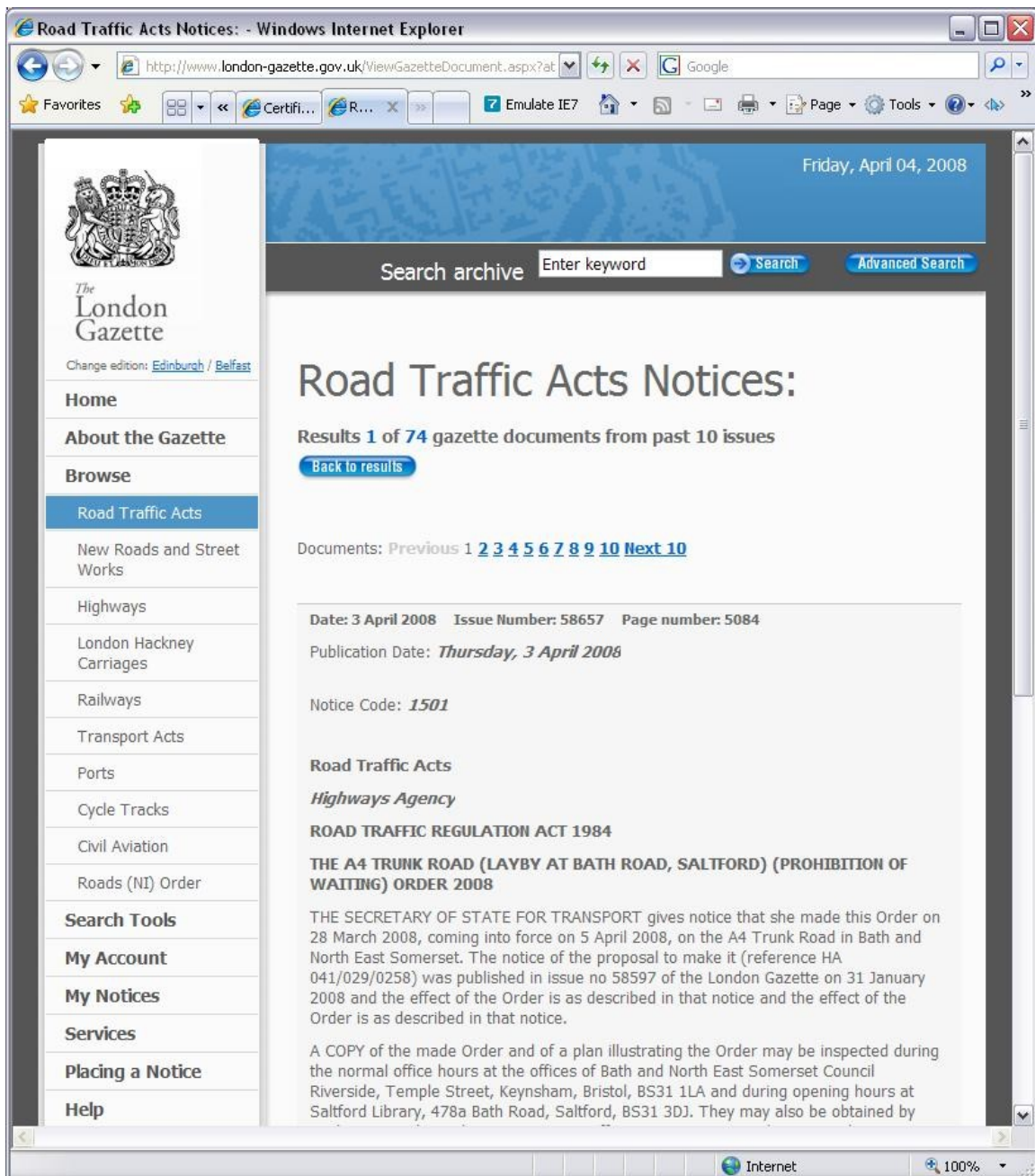
For our purposes, the most important part of the site are the Gazette Notices. Notices are created in a number of ways. They can be posted, faxed or emailed to the London Gazette as plain text. They can be created using a form on the web. They can be submitted using a Word template or directly as XML. Whichever

form is used, the end result is an XML document in the Submission Markup Language.

This XML document is transformed into both HTML and PDF for display on the website. The screenshot in Figure 1 shows an example notice.

[notice.jpg](#)

Figure 1: An example Road Traffic Acts Notice on the Gazettes website.



The website offers a great deal of useful functionality, and it is well-designed and accessible for end-users. However, an important part of our story is that this website does not perfectly adhere to best practice: its URLs expose the back-end technology; its pages claim to be XHTML but are not well-formed. Not only are we unable to start again from scratch, but our starting point is far removed from the standards-based, RESTful ideal.

This is not unusual. As we'll see later in this paper, when you introduce semantic technologies these seemingly theoretical issues become very practical problems.

## Semantic Web

The semantic web is a long-standing dream for many members of the web community. The Resource Description Framework (RDF) was originally standardised in 1999 [*LASS99*], over nine years ago, with the aim of exposing machine-understandable data on the web. (That Recommendation has since been superseded by a suite of documents on RDF; the easiest starting point is the RDF Primer [*MAN04*].)

RDF allows you to make *statements* about *resources*, which can be real-world things like people or locations or web-things such as HTML documents. A statement consists of a *subject* resource, a *predicate* which is a property or relationship of that subject, and either an *object* resource or a literal value such as a number or date. This fundamental data model can be used to facilitate light-weight domain-specific mash-ups or larger-scale applications that construct triple stores across multiple domains.

RDF statements can be expressed in a number of ways, the most readable of which is using Turtle [*BECK07b*]. For example,

```
@base "http://www.london-gazette.gov.uk/" .
</issues/20070810/notices/321825>
  a g:RoadTrafficNotice ;
  g:published "2007-08-10"^^xsd:date ;
  g:issue </issues/20070810/> .
```

states that the resource `http://www.london-gazette.gov.uk/issues/20070810/notices/321825` is a Gazette Road Traffic Notice published on 10th August 2007 in the issue `http://www.london-gazette.gov.uk/issues/20070810`.

An RDF schema [BRI04] or ontology in the Web Ontology Language (OWL) [MCG04] can be used to describe classes of resources (such as Notices, and more specifically Road Traffic Notices in the Gazette) and the properties that they can have (such as `g:issue` or `g:published` for Gazette Notices). An ontology defines not only the names of classes and properties but the relationships between them, which helps both human readers and RDF reasoners draw conclusions based on RDF data.

RDF has been very successful in limited contexts, such as within individual organisations, but the last year has seen some major steps forward towards making a more connected semantic web a reality.

First, GRDDL became a Recommendation in September 2007 [CON07]. GRDDL provides a way of pointing from an XHTML (or, more generally, XML) document to an XSLT transformation that will turn the document into RDF/XML. An application that understands RDF can visit a web page and easily turn its human-readable contents into RDF statements.

Second, less importantly for our purposes but significant none the less, the SPARQL query language for RDF became a Recommendation in January 2008 [PRUD08]. This provides a standard way of querying existing RDF graphs and constructing new ones.

Third, RDFa has been further developed and formalised, reaching Last Call in February 2008 [ADID08]. Where GRDDL provides a general-purpose method of extracting RDF from a web page, RDFa provides a specific syntax for augmenting normal XHTML documents with semantic information.

The idea that your XHTML pages can serve a double purpose of providing human-readable information and computer-understandable data has been around for a few years, largely thanks to the microformats community [MIC08]. With GRDDL, microformats can be exposed to RDF-aware applications as RDF. However, we chose to use RDFa because:

- Microformats are deliberately limited to existing and widely adopted types of information, such as events and contact information. The Gazettes hold information of other sorts, such as references to legislation and Gazette-specific metadata, for which there are no microformats.

- The Gazettes are an official publication, and as such need to be fully accessible for people with visual impairments. Some of the microformats patterns

of use, in particular the use of the `title` attribute to hold computer-understandable data, are not consistent with this aim. By avoiding existing markup, RDFa avoids these problems.

·RDFa offers more flexibility for dealing with some of the messier markup that we have to deal with. For example, it allows us to talk about the date and location of a meeting without having an element that wraps that date and location together.

## If We Build It...

Adding semantic information to a website requires a certain leap of faith: a belief that making data available to the world will enable others to construct new and innovative applications that they would otherwise not have been able to build.

Our initial focus has been on four types of notices:

·**Transport** notices, which include information about when and where road works are planned to take place, roads closed, new roads built, and other transport-related information such as parking charges and airport charges.

·**Planning** notices, which include the stopping up of roads to enable building work to take place, changes of the use of land, the introduction and expansion of conservation areas, and so on.

·**Environment** notices, which describe applications for permits to do things that may have an environmental impact, such as quarries or food factories.

·**Water** notices, which include information about waste water being discharged into the sewer system, or treated water onto land, and other water-related notices.

We imagine users of the Gazette website getting an immediate benefit if they use plug-ins such as Operator [*KAP08*], Piggy Bank [*SIM08*] or Fuzzbot [*FUZZ08*]. For example, this might give users the ability to bring up the context menu and simply add a meeting to their calendar, or a person to their address book.

We imagine web-based applications that enable people to subscribe to feeds that supply them with notices that concern areas that they're interested in, or that

keep members of the public informed about the environmental impact of particular companies they want to monitor.

We imagine applications that mash-up information gleaned from the notices about road-works with maps that show where the road-works will take place.

We imagine policy-makers analysing Gazette notices to see the impact of legislation on the areas in which construction takes place, or the use of water resources over time.

But the most welcome reuse scenarios are those we haven't imagined.

## Problems and Solutions

In this section, we'll discuss the issues we faced when adding semantic information to the London Gazette website and how we addressed them.

### URLs

RDF requires that each resource that we want to make statements about has an identifier, and that each time we talk about that resource, we use the same identifier. These identifiers are URLs, partly because URLs provide an easy way of subdividing the space of possible identifiers, and partly because URLs are resolvable: an application can make an HTTP GET request to a URL and retrieve something useful in return.

Using RDF brings with it two requirements for URLs:

- Every resource should have a single, standalone, identifying URL. Using the same URL whenever we refer to a particular resource enables RDF-based reasoners to pull statements together: if two statements have the same subject URL then a reasoner knows they concern the same thing. (The `owl:sameAs` property provides a way of saying that two identifiers refer to the same resource, but it is good to aim for a single URL per resource within the space of URLs that is controlled by a single organisation.)

- The URLs used in RDF should be permanent. Statements that are made about resources will persist indefinitely into the future, in systems over which we have no control. It should always be the case that GETting the URL will return a meaningful resource. This implies that the URLs used must be free from short-

term considerations such as the back-end content management system used to return the resource.

The principles of unique and permanent URLs have been recognised as good practice for a long time, but they become vitally important when the semantic web is introduced.

This posed a big challenge for the Gazette website, which currently uses URLs such as:

```
http://www.londongazette.gov.uk/ViewGazetteDocument.aspx?atdocid=4753641&GeoType=London&categorydocids=112&lastissuecount=10
```

The existing URL scheme:

- exposes the back end technology: the `atdocid` request parameter relates to an identifier in the underlying content management system
- only works in combination with cookies: cookie-based session information determines the search context you see when browsing
- provides multiple URLs for a single notice

The current recommended best practice for RDF identifiers [SAU08] is to distinguish between resource identifiers and document URLs. Resource identifiers are used as identifiers within RDF graphs, while document URLs are used on the web to retrieve documents. When an application makes a request to the resource identifier, the server should return a 303 Redirect status code which directs the application to a document URL that represents that resource. At that point, content negotiation can take place, so that the application can request RDF/XML or HTML or another form of representation of that document.

We therefore introduced identifier URLs such as

```
http://www.london-gazette.gov.uk/issues/20070810/notices/321825
```

to retrieve notice number 321825 from the 10<sup>th</sup> August 2007 issue of the London Gazette. Requesting this resource results in a 303 redirection to

```
http://www.london-gazette.gov.uk/issues/20070810/notices/321825.htm
```

which is the XHTML page containing RDFa annotations.

We took a simple approach to implementing these URLs, namely to introduce a redirection layer between a new set of permanent URLs and the URLs used by the back end. Since we use Microsoft IIS, this was implemented using the Ionics ISAPI redirect filter [IIRF08], but can be done just as easily with Apache's `mod_rewrite`. Redirection rules return a 303 Redirect status for identifier URLs and perform internal redirection to the existing URL for each resource. Thus, behind the scenes, a request to this second URL above is mapped to:

```
http://www.gazettes-online.gov.uk/ViewGazetteDocument.aspx
?src=search&atdocid=4441414&GeoType=London
```

As we discovered, there are disadvantages to using a basic redirection layer.

First, you cannot completely avoid changing the back-end system. In our case, the back end only supported accessing notices by the internal document identifier used by the content management system. When we introduced URLs that instead used the notice number, we needed to also introduce mechanisms for mapping notice numbers to document identifiers.

Second, mismatches between the significant portions of the identifier URLs and the back end URLs mean that the server responses aren't always accurate. In our case, notice numbers are in fact unique, and in the mapping from document URL to internal URL above, only the notice number is significant. We used hierarchical URLs based on issue date and notice number because they are more structured, readable and "hackable" than a flatter URL based only on notice number. Requesting a notice from an issue that doesn't contain that notice, such as

```
http://www.london-gazette.gov.uk/issues/20071010/notices/3
21825
```

should return a 404 Not Found status, but the redirection layer doesn't have enough information to tell that there's no such notice, and the back end doesn't know to limit the search results to only that issue.

## Ontologies

Using RDF entails using ontologies of some sort: deciding what the important resources are within the application and what their properties are. There are already a number of ontologies in existence, some large, some small, some formal, some loose.

This project requires ontologies that cover a range of things:

- the structure of the Gazette: Editions, Issues, Notices, Supplements, and so on
- the legislation referenced by notices and the sections within that legislation
- organisations that might post notices or be the subject of notice, such as companies, government departments and local councils
- people, places and events

The main lessons we learned here were:

·**Small is beautiful.** Smaller ontologies are easier to maintain and easier to reuse. For example, we created an ontology for legislation that could be used in other applications that need to talk about legislation.

·**Reuse is great if the ontology fits.** The Ordnance Survey ontologies [*OS08*] helped us enormously by providing information about administrative areas within the UK, which we reused in our definitions of administrative bodies such as local councils.

·**If you can't reuse an ontology as-is, at least tie it in.** Gazette notices sometimes talk about people, for example providing contact information so that members of the public can get in touch with those in charge of a planning application. We looked at using FOAF [*FOAF08*] or vCard in RDF [*IAN01*], both of which only partially fitted with what we needed to do. So we created our own ontology for people, but used `owl:equivalentClass` and `owl:equivalentProperty` to tie in our new ontology with those existing ones.

·**If your data is in XML, mirror that markup language.** Transformations between XML and RDF versions are infinitely easier if the classes and properties in your OWL ontology mirror the semantics and terminology of the elements and attributes you have in your XML schema.

·**You have to stop somewhere.** There's a great temptation, when creating an ontology, to keep adding classes and properties: to attempt to describe the world. It's important to think about what level of description is going to be useful for the people reusing your data.

We mostly used Protégé [*PROT08*] to create our ontologies, but some had to be edited by hand (particularly to introduce `owl:equivalentClass` and `owl:equivalentProperty`). Our councils ontology was created based on the Ordnance Survey Administrative Geography ontology using an XSLT transformation.

## RDFa

RDFa is based on XHTML. Applications that extract RDFa, including GRDDL transformations, require that pages be (at least) well-formed in order to work. Like many other documents found on the web, the Gazette pages claimed validity by having a `<!DOCTYPE>` declaration but were neither well-formed nor valid.

The Gazette web pages are created using ASP. Some parts, such as the footer, come from static documents created by human editors. The main content is generated from XML using XSLT. The rest of the page is written within the ASP itself. We therefore needed to look at three levels:

- The ASP and XSLT code needed to create valid XHTML.
- The editors who created the static content needed to be trained and given support tools to ensure that what they created was valid. This included introducing a validation-on-check-in, so that it was impossible to add invalid static XHTML to the pages.
- The code needed to be integrated such that the page was valid as a whole.

Excepting the addition of the RDFa itself, four other changes were required in our web pages to support RDFa:

- We had to add namespace declarations for the base URIs of the ontologies that we wanted to use, including the XML Schema datatype base URI.
- These namespace declarations aren't included in the basic XHTML DTD, so our documents were then invalid. We had to create our own DTD and reference it from the `<!DOCTYPE>` declaration of the page. (It should be possible to use an internal subset rather than a separate DTD; unfortunately browsers tend to display ' ]>' at the top of the page if you do so.)

·A `<base>` element needed to be added giving the identifier URL for the notice; this acts as the initial object during RDFa parsing. Relative URLs in the rest of the page needed to be updated to account for the change in base URL.

·We needed to support GRDDL so that applications could generate a RDF/XML version of the document. We added `profile="http://www.w3.org/2003/g/data-view"` to the `<head>` element and included a `<link rel="transformation">` pointing to the RDFa-to-RDF/XML transformation we wrote.

The RDFa itself is inserted in two places within the pages: in the `<head>` and in the `<body>`.

General metadata about a notice, such as the issue it's in, is placed within the `<head>` of the document in standard `<meta>` and `<link>` elements such as

```
<meta property="dc:title" content="London Gazette: Issue
58475 dated 10 October 2007: Notice 364061" />
```

and

```
<link rel="gr:issue" href="/issues/2007-10-10" />
```

These follow the normal patterns for metadata in XHTML, except that the `property` and `rel` attributes hold CURIEs (compact URIs) which are resolved during RDFa processing into full URIs based on the namespace declarations in the document. (In RDFa, the `href` attribute can hold a CURIE or a URL; we used CURIEs when referring to terms in known vocabularies, and URLs for more general purpose URLs because they are recognisable to non-RDFa processors.)

One pattern we found useful here was the definition of a blank node for properties such as `dc:subject` which have both a value (`rdf:value`) and a vocabulary (`dcam:memberOf`). The following code illustrates this pattern:

```
<link rel="dc:subject" href="[_:subj]" />
<link about="[_:subj]" rel="dcam:memberOf"
href="[egms:IPSV]" />
<link about="[_:subj]" rel="rdf:value"
href="[ipsv:RoadTransport]" />
```

which produces the triples:

```
<> dc:subject [
```

```
    dcam:memberOf egms:IPSV ;
    rdf:value ipsv:RoadTransport ] .
```

The RDFa in the main body of the page is used to annotate existing content rather than to add metadata. Plain data is marked up by adding property attributes to XHTML elements (most usually `<span>`):

```
<span property="g:hasNoticeNumber"
  datatype="xsd:string">421373</span>
```

Where the information is in a human-readable form, we can provide a computer-readable version using the `content` attribute:

```
<span property="g:hasPublicationDate" datatype="xsd:date"
  content="2007-12-28">Friday, 28 December 2007</span>
```

The final pattern that we use a lot is illustrated here:

```
<span rel="g:hasAuthority">
  The <span about="[org:CivilAviationAuthority]"
    instanceof="g:Authority org:PublicCorporation"

    property="g:isKnownAs">
    Civil Aviation Authority</span>
</span>
```

This creates the triples:

```
<> g:hasAuthority org:CivilAviationAuthority .
org:CivilAviationAuthority
  a g:Authority ;
  a org:PublicCorporation ;
  g:isKnownAs "Civil Aviation Authority" .
```

In this pattern, the `rel` attribute on the outer `<span>` creates an "incomplete triple" with a subject (the notice) and a property (`g:hasAuthority`) but without an object. This triple is completed by the resource that's the subject of the nested `<span>`, in this case `org:CivilAviationAuthority`. This pattern is particularly useful when the RDFa is automatically generated, as it separates the concerns of creating the relationship from creating information about the resources mentioned in the text. It also mirrors the "striped" style of markup you see in RDF/XML.

The W3C validator [*W3C08*] can be used to check the validity of XHTML+RDFa documents but only looks at whether the correct attributes are used, not whether they are used in a way that makes sense.

Checking that the correct triples were generated was more complicated, particularly as RDFa has been a moving target, with no reference implementation available. We eventually simply used our GRDDL transformation, using the Triplr service [*BECK07a*] to generate Turtle [*BECK07b*] which we could check by eye. We expect this situation to improve when RDFa reaches Recommendation status.

## Adding Semantics

Notice content is stored in the Gazette Submission Markup Language. This XML is transformed using XSLT to create the XHTML+RDFa that is served on the Gazette website.

We have had to extend the mark-up language we're using in the light of the way in which RDFa exposes semantic information.

The current mark-up language has focused on marking up structures within a few very formalised notice types. The notice types we're looking at now are less structured, so some elements that we used to allow only at a top level need to be allowed inline as well. We have to also shift from semantic mark-up that is useful for determining presentation towards mark-up that might be useful for people who reuse the Gazette data.

For example, the following text is found in a notice about conservation areas:

`Nos 15-35 (odds), Nos 30, 52, 68 View Road`

This address list is far removed from the kind of addresses we find in vCard, say, and it requires specialised markup if we are to expose the underlying data in a reusable way.

The current version of the Submission Markup Language can mark up people's names, addresses and dates, but is unable to say that a person has a particular address or that a meeting is to be held on a given date at a particular place. In other words, the elements are currently isolated and need to be tied together, either explicitly or by their proximity, to create the kind of assertions we want to make in the RDFa.

An illustration of this problem is shown in this sentence:

```
Any objections must be in writing and sent to
<PersonName>Mr Waziri Sudi</PersonName>,
<PersonRole>Conservation Officer</PersonRole>,
<Address>Urban Regeneration & Housing, Development
Services - Development Planning, Town Hall, St Helens,
WA10 1HP</Address>.
```

We would like to expose this in RDFa as something like:

```
Any objections must be in writing and sent to <span
about="[_:MrWaziriSudi]" property="p:hasName">Mr Waziri
Sudi</span>, <span about="[_:MrWaziriSudi]"
property="p:hasRole">Conservation Officer</span>, <span
about="[_:MrWaziriSudi]" property="p:hasAddress">Urban
Regeneration & Housing, Development Services - Development
Planning, Town Hall, St Helens, WA10 1HP</span>.
```

to generate the triples:

```
[_:MrWaziriSudi]
  p:hasName "Mr Waziri Sudi" ;
  p:hasRole "Conservation Officer" ;
  p:hasAddress "Urban Regeneration & Housing, Development
Services - Development Planning, Town Hall, St Helens,
WA10 1HP" .
```

but there is currently nothing in the XML, except for proximity, that links the person's name, their role and their address together. (Assessing proximity might require assessing whether markup is within the same *sentence*: a construct that isn't exposed in the markup.)

In addition to issues with the markup language, the semantic markup in the XML documents themselves can vary considerably:

- Authors do not mark up everything that could be marked up. For example, authors will very rarely mark up people's names or addresses, partly because there is no benefit to them of doing so and partly because any such markup is effectively lost (at least to human readers) when the material is presented in HTML or PDF forms. So in many cases the semantic markup is simply not present in the documents.

- When creating documents with a WYSIWYG editor such as Word, authors use styles to achieve a certain look rather than because the semantics of the style

are appropriate to the content. (This is known as tag abuse.) So in other cases, the semantic markup is wrong.

The text of the documents themselves can contain mistakes or simple variations that mean that even when the markup is present and correct, the content is wrong or non-standard. For example, "Royal Borough of Kensington and Chelsea" might appear as "Royal Borough of Kensington & Chelsea" or "Royal Borough of Kensignton [sic] and Chelsea".

We are therefore currently examining options for validating the markup more strictly than previously, and of automating the mark up of notice content. Many of the semantics that we want to expose either adhere to standard patterns (such as dates or postcodes) or come from known vocabularies (such as the administrative areas listed in the Ordnance Survey ontologies, or the legislation listed on the OPSI website). We are therefore hopeful that we'll be able to mark up quite a lot without needing full natural language processing.

## Rights

The Semantic Web is all about reusing information. It is important to think about who owns what information and how best to enable others to legitimately reuse it. Copyright and database rights arise automatically when a work is created or the necessary investment is made in a database. We have to think about rights to give confidence and assurance to reusers – even when the material is being made available for free. With our “maximise reuse” objective in mind, two issues are of particular note:

1. What rights does the person or organisation that created the original content (i.e. placed the notice) have over the work they have created?

This varies depending on how the notice was constructed. From a copyright perspective, filling in fields in a form on the web -- which is then used by the application to construct a notice based on a template – is different from writing and supplying the full text of the notice. In the former case much if not all the creative work lies in the design of the template, so the poster of the notice might own no rights; in the latter case the person placing the notice has authored a creative work and thus is the copyright holder. Onward reuse of that information is governed by the Terms and Conditions associated with placing the notice.

2. Does adding semantic mark-up create a new database right in the material?

A database right is different from copyright. In the context of rights, a database is defined as being a collection of data or other materials arranged in a systematic or methodical way so that items are individually accessible. It is not enough to collect and store data to create database right. For a database right to apply there must be a substantial investment in obtaining, verifying or presenting the contents of the database. We think that by adding semantic mark-up to existing information we are creating a new database right. Given that semantic enrichment may become increasingly common we have decided to seek legal advice from The Treasury Solicitor's Department (The Government's Lawyers) as it is an interesting and important consequence of our work. Practically, we need to resolve the question of what rights are in the data so we can construct an appropriate licence permitting its reuse.

## Future Work

A beta version of the work described in this paper is available on the Gazettes website at <http://www.london-gazette.gov.uk/>.

Our immediate next steps are to improve the semantic mark-up of the notices addressed in this project and extend the approach to more notice types. On the basis of what we've done so far, we believe RDFa provides a good approach for serving data from the London Gazette to the Semantic Web. We also hope to drive the use of web standards on the Gazette website generally by improving XHTML validity and using the URLs we've developed as identifiers for navigation throughout the site.

The long-term strategy for the London Gazette is to make it a highly re-usable information asset via the web -- part of the nation's core information infrastructure. Future work will increasingly shift the London Gazette from a paper-based official publication into an online commodity that we hope will form the basis of many and varied applications as part of the web of linked data.

This project also takes forward the UK Government's general thinking about practical approaches for enabling the reuse of public sector information. By showing how it can be done, we hope to inspire others to follow us down this road.

# Acknowledgements

The project reported in this paper is the product of a lot of work from other people at TSO and OPSI, with advice from the University of Southampton. We would especially like to thank those at TSO and OPSI who have helped in reviewing this paper, namely Paul Appleby, Robin Brattel, Shaun Bigg and Tim Padfield..

## Bibliography

- [ADIDo8] Adida, B., Birbeck, M., McCarron, S. & Pemberton, S. (2008) *RDFa in XHTML: Syntax and Processing*. W3C Last Call Working Draft. [URL: <http://www.w3.org/TR/rdfa-syntax/>]
- [BECKo7a] Beckett, D. (2007) *Triplr*. [URL: <http://triplr.org/>]
- [BECKo7b] Beckett, D. (2007) *Turtle – Terse RDF Triple Language*. [URL: <http://www.dajobe.org/2004/01/turtle/>]
- [BRIO4] Brickley, D. & Guha, R.V. (eds). (2004) *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation. [URL: <http://www.w3.org/TR/rdf-schema/>]
- [CONo7] Connolly, D. (ed) (2007) *Gleaning Resource Descriptions from Dialects of Languages (GRDDL)*. W3C Recommendation. [URL: <http://www.w3.org/TR/grddl/>]
- [FOAFo8] The Friend of a Friend (FOAF) project [URL: <http://www.foaf-project.org/>]
- [FUZZo8] Fuzzbot [URL: <http://rdfa.digitalbazaar.com/fuzzbot/>]
- [IANo1] Ianella, R. (2001) *Representing vCard Objects in RDF/XML*. W3C Note. [URL: <http://www.w3.org/TR/vcard-rdf>]
- [IIRFo8] Ionics Isapi Rewrite Filter [URL: <http://www.codeplex.com/IIRF>]
- [KAPo8] Kaply, M. (2008) *Operator*. Mozilla Plug-in. [URL: <https://addons.mozilla.org/en-US/firefox/addon/4106>]

- [LASS99] Lassila, O. & Swick, R. (eds) (1999) *Resource Description Framework (RDF) Model and Syntax Specification*. W3C Recommendation [URL: <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>]
- [MANo4] Manola, F. & Miller, E. (eds) (2004) *RDF Primer*. W3C Recommendation [URL: <http://www.w3.org/TR/rdf-primer/>]
- [MAYo7] Mayo, E. & Steinberg, T. (2007) *The Power of Information: an independent review*. The Cabinet Office. [URL: [http://www.cabinetoffice.gov.uk/upload/assets/www.cabinetoffice.gov.uk/strategy/power\\_information.pdf](http://www.cabinetoffice.gov.uk/upload/assets/www.cabinetoffice.gov.uk/strategy/power_information.pdf)]
- [MCGo4] McGuinness, D.L. & van Harmelen, F. (eds) (2004) *OWL Web Ontology Language Overview*. W3C Recommendation [URL: <http://www.w3.org/TR/owl-features/>]
- [MICo8] Microformats [URL: <http://microformats.org/>]
- [OSo8] Ordnance Survey Ontologies [URL: <http://www.ordnancesurvey.co.uk/oswebsite/ontology/>]
- [PROTo8] The Protégé Ontology Editor and Knowledge Acquisition System [URL: <http://protege.stanford.edu/>]
- [PRUDo8] Prud'hommeaux, E. & Seaborne, A. (eds) (2008) *SPARQL Query Language for RDF*. W3C Recommendation. [URL: <http://www.w3.org/TR/rdf-sparql-query/>]
- [SAUo8] Sauermann, L. & Cyganiak, R. (eds) (2008) *Cool URLs for the Semantic Web*. W3C Working Draft. [URL: <http://www.w3.org/TR/cooluris/>]
- [SIMo8] Piggy Bank – SIMILE [URL: [http://simile.mit.edu/wiki/Piggy\\_Bank](http://simile.mit.edu/wiki/Piggy_Bank)]
- [W3Co8] The W3C Markup Validation Service [URL: <http://validator.w3.org/>]