Publication Details

Author: Renato Iannella
Email: ri@semanticidentity.com
Copyright: Semantic Identity 2010-14

Acknowledgements

The information contained in this report represents analysis of publicly available sources believed to be reliable as of the date of this whitepaper. This document is for informational purposes only. Semantic Identity shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof, expressed or implied, in this document.

Thanks to Michael Hausenblas (DERI) for feedback on drafts of this document.

© 2010-2014 Semantic Identity. All Rights Reserved.

About Semantic Identity

Semantic Identity provides independent consulting, thought leadership, technical advice, research, and strategic direction for emerging ICT technologies. We focus our expertise around; Information Architecture, Semantic Web; Social Media; Policy & Privacy; Information Modeling; Rights Management; Metadata and Identifiers and Data Analytics with a philosophy based around Web Science.

http://semanticidentity.com
# Table of Contents

1. Executive Summary .................................................................................................................. 1
2. The Semantic Web ........................................................................................................................ 2
   2.1. Semantic Web Themes .............................................................................................................. 2
       2.1.1. Linked Data ...................................................................................................................... 2
       2.1.2. Vocabularies ................................................................................................................... 3
       2.1.3. Query .............................................................................................................................. 3
       2.1.4. Inference ........................................................................................................................ 3
       2.1.5. Vertical Applications ....................................................................................................... 3
   2.2. The Technology Map .............................................................................................................. 4
       2.2.1. RDF .............................................................................................................................. 4
       2.2.2. OWL .............................................................................................................................. 7
       2.2.3. SPARQL .......................................................................................................................... 8
       2.2.4. RIF ................................................................................................................................... 9
       2.2.5. GRDDL ..........................................................................................................................10
       2.2.6. POWDER .......................................................................................................................10
       2.2.7. SKOS .............................................................................................................................11
       2.2.8. RDB2RDF .....................................................................................................................12
   2.3. Additional Semantic Web Technologies ...............................................................................12
   2.4. Maturity Analysis ..................................................................................................................13
   2.5. Semantic Web Case Study ...................................................................................................14
3. Semantic Web Architectures .................................................................................................16
   3.1. Semantic Web Tools ............................................................................................................18
4. Scenarios .....................................................................................................................................20
   4.1. Commercial ITS Systems ....................................................................................................20
   4.2. ITS Enterprise Architectures ...............................................................................................21
   4.3. Amber Alert ........................................................................................................................23
5. Conclusion ...............................................................................................................................26
   5.1. Summary .............................................................................................................................26
   5.2. Future ....................................................................................................................................27
6. References ...............................................................................................................................28
1. Executive Summary

The Semantic Web combines a set of new technologies with grounded knowledge representation techniques to address the needs of more formal information modelling and reasoning for web-based services. The Semantic Web is used for many purposes from a standardised way to markup metadata to describe digital resources to a new growing movement favouring the open and shared expression of common ontologies.

The benefits of the Semantic Web include:

- Consistent mechanisms to model information from simple vocabularies to complex ontologies.
- A formal model approach ensures information reasoning outcomes.
- Data linking opportunities aimed at supporting better user experiences, and hence, improved business outcomes.
- A groundswell of activity in the development of open-source tools to exploit Semantic Web technologies and information
- Standardised by the W3C indicating global consensus and open royalty-free specifications.

This white-paper provides a high-level summary of Semantic Web technologies to help better understand the impact that these technologies will have on wider enterprise information architectures. The Semantic Web will play a larger role in new enterprise rollouts in the future, albeit initially with some trepidation, but with more confidence as the longer term benefits are realised with improved outcomes for the enterprise and end users.

The Semantic Web’s heritage has evolved from many years work in the “Knowledge Representation” domain and this is an important scoping feature. That is, the Semantic Web should not be seen as a new all-embracing technology that solves all current and future ICT problems. It is not a Business Processing Language, nor a Grid infrastructure, nor a Cloud, but can provide new ways to help these domains in information management. In many cases it also reuses familiar concepts with a new twist. For example, “ontologies” for “data dictionaries” and “semantic model” for “data model”. The twist is that the Semantic Web provides a new set of technologies and formal methodological model to undertake these tasks consistently and across the full information management spectrum.

The formal aspects of the Semantic Web provide a key long-term benefit for the enterprise. Large organisation are inefficient when it comes to data integration, sharing, and reuse and often duplicate data with unforeseen consequences. The formal modelling and reasoning pedigree of the Semantic Web reduces this impact by allowing information entities to be clearly defined, openly linked, integrated, and extended by stakeholders for greater decision support.

The key messages for the Semantic Web Architect include:

- Semantic Web technologies provide a strong base for long term stability.
- Enterprise Architectures can be improved by adopting Semantic Web techniques in the information and data layers.
- Business improvements are likely outcomes by early adoption of an emerging technology that fit the business goals of the organisation.
2. **The Semantic Web**

The term “Semantic Web” refers to W3C’s vision of technologies that enable people to create data stores on the Web, build vocabularies, and write rules for handling data empowered by technologies such as RDF, SPARQL, OWL, and SKOS [1]. The term became popular after Tim Berners-Lee’s semantic web article in Scientific American over a decade ago [2]. In this article he described the Semantic Web as expanding the current web with better ways to express meaning, represent knowledge and define ontologies. Since then, a full set of new technologies has been developed by the W3C to fulfil this promise of a smarter web. Some see the Semantic Web as being the cornerstone for “Web 3.0”.

### 2.1. Semantic Web Themes

Today, the W3C positions the Semantic Web into a number of areas:

- Linked Data
- Vocabularies
- Query
- Inference
- Vertical Applications

Let’s briefly look at these five areas with this simple scenario:

“A traffic accident at the corner of Garden St and Central Ave in Eveleigh, Sydney”

#### 2.1.1. Linked Data

The web is full of data and in order for the Semantic Web to operate fully, the “web of data” needs to be organised in such a way that it is available in a standard format and relationships between data are visible. Linked Data is the ability to make explicit links between data and applications may exploit the extra knowledge from other data sources by virtue of integrating information from several data sources, thereby providing a much better user experience. Linked data is fundamental to the Semantic Web and works based on the use of unique identifiers (URIs) for entities.

Figure 2.1 shows the growing community of Linked Data sources that form the Cloud of interlinked data. The power of the links provides consistency across data sources and exposes relationships between data that can lead to greater knowledge creation and user experiences.

![Figure 2.1 - Linking Open Data Project Cloud](http://lod-cloud.net)
In our scenario, we may identify “Garden St” with a unique identifier. This means that any other application or service may also “link” to Garden St and these applications know explicitly what this entity is about and, more importantly, that it is the same entity. For example, the Sydney City Council plans to resurface “Garden Street” with new bitumen and even though they call it slightly differently, they use a unique identifier so that other authorities know exactly which entity is being referred to.

2.1.2. Vocabularies

Vocabularies define the concepts and relationships used to describe and represent an application domain. Vocabularies are used to classify the concepts, represent possible relationships, and define constraints on using those entities. Typically, vocabularies can range from very simple (eg a dozen entities) to be very large and complex (eg tens of thousands of entities). There is no clear division between the what is referred to as “vocabularies” and “ontologies”. The current trend is to use the word “ontology” for large, complex, formal collections of entities, and “vocabulary” for smaller cases.

In our scenario, we may classify “Garden St” and “Central Ave” as instances of types (classes) of “Roads” and that “Street” and “Avenue” are all subtypes of Road. Additionally we may define a “Corner” where at least two roads intersect.

2.1.3. Query

The Semantic Web needs programatic mechanisms to retrieve all the data, just like relational databases or XML need specific query languages. The Semantic Web, typically represented using the RDF model, needs its own query language and services.

In our scenario, applications may wish to find out all current traffic incidents in Eveleigh or how many accidents have occurred on Garden St. The Semantic Web defines are way we can express these queries and the return the results to the client application.

2.1.4. Inference

Inference on the Semantic Web can be characterised by the automatic discovery of new relationships. Since the Semantic Web data is formally modelled with Ontologies via a set of named relationships, inferences can be extracted or explicitly added or returned from a query. The Semantic Web also supports inference measures from rule sets to discover and generate new relationships based on existing ones.

In our scenario, even though we did not explicitly state this in our original data, we can infer that since Garden St is a type of “Street”, which is a subtype of “Road”, then Garden St is also a “Road”. This is classic object-oriented inferencing. More complex inference mechanisms are extremely application specific, for example, if I know that a “party” event will be held on “Garden St”, then I could infer that this is a “street party”.

2.1.5. Vertical Applications

Vertical applications areas are used to explore how Semantic Web technologies can help improve operations, efficiencies, and provide better user experiences. The two most prominent areas for Semantic Web adoption are Health Care/Life Sciences and e-Government sectors as these sectors provide valuable use cases, feedback, and deployment scenarios used to improve the range
of Semantic Web technologies. Of course, the Semantic Web is a generic set of technologies and can be applied to any sector/community.

In our scenario, there would a number of different government agencies involved in responding to the traffic accident. State-based Emergency Services (eg Police, Ambulance) and Local Council services (eg to repair the damaged traffic lights) may all respond. The same information messages need to be transmitted and understood by these different agencies. In some cases, the message maybe mapped to local vocabularies or entities translated to alternatives (eg Eveleigh maybe translated to a postcode, or the geospatial coordinates of the road intersection maybe preferred for some agencies).

2.2. The Technology Map

The Semantic Web has evolved into a number of specific technologies that provide the support for the semantic services. Figure 2.2 shows the “Semantic Web Layer Cake” which clearly shows the central role the core technologies (eg RDF, OWL) play in the overall web architecture.

We now briefly review and highlight the key features of the Semantic Web family of technologies. This is not an exhaustive review, but aims to show the core capabilities and positions the technology in the overall web architecture.

2.2.1. RDF

The Resource Description Framework (RDF) [3] was the first and is the fundamental technology of the Semantic Web. It defines the underlying information model based on triples that underpins all semantic web mechanisms. The RDF model is very simple and yet powerful. It describes Resources through a series of Statements. Each statement is a Property and Value that is about a Resource.

The RDF triple connects the Resource to the Property and Value. More formally these are Subject, Property, and Object, respectively. The other key ingredient of the RDF Model is that Subject and Property are always identified with URIs, and Object via a string Literal or a URI. The RDF triples form directed, labelled graphs. Additionally, the RDF statements can be represented in machine interoperable formats, such as XML, Turtle, and RDFa, the latter for embedding RDF in HTML web pages.

Figure 2.3 shows a (simplified) RDF graph for our scenario in which all the oval nodes are Resources and the connecting lines are Properties. The Objects are also Resources (what the line points to) in this example, but could have been string Literals.
As mentioned previously, RDF exploits URI identifiers for Resources, Properties, and in many cases Objects. This can be clearly seen in the XML representation of Figure 2.3 below.

```
<rdf:Description rdf:about="http://road.com/incidents/2010/06/15/00000001111222">
  <x:inLocation rdf:nodeID="Intersection"/>
</rdf:Description>

<rdf:Description rdf:nodeID="Intersection">
  <x:hasRoad rdf:resource="http://road.com/road/garden/st/01"/>
  <x:hasRoad rdf:resource="http://road.com/road/central/av/01"/>
</rdf:Description>

<rdf:Description rdf:about="http://road.com/road/garden/st/01">
</rdf:Description>
```

You can now clearly start to see how Linked Data works with so many unique identifiers for common entities. For example, if some other application was to refer to “http://road.com/road/garden/st/01” then both applications are referring to the same unique entity (Garden St in Eveleigh, Sydney).

RDF also supports a compact syntax called RDFa [45]. RDFa provides a set of XHTML attributes to augment visual data with machine-readable data. It’s aim is to embed RDF data into Web Pages without the need to repeat the data twice; once for the human to visually read, and another for the machine to parse and understand. The example below shows a Web page snippet of the simple scenario expressed in both human-displayable HTML and machine-readable RDFa.

```
<div about="http://road.com/incidents/2010/06/15/00000001111222" typeof="x:TrafficAccident">
  A traffic accident at the corner of
  <a rel="x:hasRoad" href="http://road.com/road/garden/st/01">Garden St</a> and
  <a rel="x:hasRoad" href="http://road.com/road/central/av/01">Central Ave</a>
</div>
```
Another significant part of RDF is its Schema language. RDF Schema allows for greater specification of semantics for information models. Specifically, the Classes (ie types), Properties, and constraints.

From our simple scenario, we can declare that the “Garden St” Resource is of class “Street” which is a subclass of “Road”. Similarly, “Central Ave” Resource is of class “Avenue” which is also a subclass of “Road”. Of course, in our modelling process, we have decided that “Street” and “Avenue” are significant entities that warrant their own class. Alternatively, the could both be of class “Road” (which they indirectly are).

In addition to the Classes, we can also define the specific Properties that make up the characteristics of those classes of things. For example, we can define “length” as a Property for all Road classes. RDF also provides a vocabulary for describing how properties and classes are intended to be used together in RDF data. These include Range (used to indicate that the values of a particular property are instances of a designated class) and Domain (used to indicate that a particular property applies to a designated class). Figure 2.4 shows the expanded RDF Schema model for our simple scenario.

RDF has a number of other features, including:

- Containers and Collections to group Resources
- Reification to describe RDF statements themselves
- Structured values to support describing relations between more than two Resources
- Defining SubProperties of Properties

For further technical information, see the RDF Primer [4].
2.2.2. **OWL**

RDF provides a core set of primitives for semantic modelling, but does not address some of the more advanced requirements. The Web Ontology Language [5], informally OWL, is a declarative knowledge representation language for the Semantic Web with formally defined meaning for creating ontologies. OWL is a ‘monotonic’ language meaning that new information added does not negate any existing statements, it simply adds to the knowledge pool. Also, OWL uses many of, and builds on, the RDF and RDF Schema constructs previously discussed.

An OWL ontology is essentially a collection of basic “pieces of knowledge.” Statements, such as in RDF, that are made in an ontology are called axioms in OWL, and the ontology asserts that its axioms are true. OWL axioms refer to objects of the world and describe their category (“Garden St is a Road”) or saying something about their relation (“Garden St and Central Ave intersect”). All the atomic constituents of statements, be they objects (Garden St), categories (Road) or relations (Intersection) are called entities. OWL denotes:

- objects as individuals,
- categories as classes, and
- relations as properties.

Properties in OWL are further denoted as:

- Object properties relate objects to objects (eg Intersection between Garden St and Central Ave).
- Datatype properties assign data values to objects (eg Length of Garden St).
- Annotation properties are used to encode meta-information about the ontology itself.

Another feature of OWL is that entities can be combined into new expressions using constructors. For example, the atomic classes “Road” and “Accident” could be combined conjunctively to describe the class of “Traffic Accidents”. This new entity would be described by an OWL class expression that then could be used in other statements.

OWL supports Classes and Class hierarchies (similar to RDF), including the ability to specify two Classes are equivalent, and adds additional capabilities, such as:

- Class Disjointness - specifically stating that an individual can be a member of one class but excluded from another. For example, we could say that all Road and Street individuals are disjoint, meaning that an individual cannot be a member of both classes.
- Object Properties - describes the relationship between two individuals (via a property). These are one-way relationships and the property name typically is expressed to capture this. For example, “inSuburb” has only one unique intuitive reading. You can also explicitly state that when two individuals are not connected via this property. For example, you can state that Yeronga is not a Suburb of Sydney.
- Property Hierarchies - allows for when property relationships implies reciprocal equivalence. For example, you can state that the inSuburb property has an equivalent hasSuburb sub-property.
- Domain and Range Restrictions for Object Properties (similar to RDF).
• Equality and Inequality of Individuals - since there is no assumption that the names of individuals need to be unique, OWL supports expressing that two individuals are the same, or not. For example, we can say that “Garden St” is the same as “Garden Street”, and that “Garden St” is not the same as “Garden Sth”. Note: this is a key feature to support Linked Data with OWL’s sameAs property.

• Datatypes - like RDF, OWL reuses the XML Schema datatypes. There are also additional constructs to express that an individual cannot have a specific data value (e.g., Length of Garden St is not 10km) and that datatype properties for specific Classes can only have a specific datatype. For example, the Length property is a positive integer for the domain of Road individuals.

OWL also includes a number of advanced relationships for complex class and property restrictions, including:

• Class intersections

• Class unions

• Class complements (logical negation)

• Existential Quantification (defines a class as the set of all individuals that are connected via a particular property to another individual which is an instance of a certain class)

• Universal Quantification (defines a class of individuals for which all related individuals must be instances of a given class)

• Property Cardinality Restrictions

• Enumeration of Individuals

• Inverse, Symmetric, and Asymmetric Properties

• Reflexive and Irreflexive Properties

• Functional and Inverse Functional Properties

• Transitive Properties (interlinks two individuals A and C whenever it interlinks A with B and B with C for some individual B)

OWL is a very expressive and complex language, both for users and computationally. As a result, OWL has defined a number of Profiles that scope the capabilities, and hence reasoning ability, into approachable subsets of OWL sufficient for a variety of applications. The direct model-theoretic semantics provides a meaning for OWL 2 in a Description Logic (DL) style and the RDF-Based Semantics is based on viewing OWL 2 ontologies as RDF graphs.

For further technical information, see the OWL Primer [6].

2.2.3. SPARQL

The Semantic Web, typically represented using the RDF data format, requires a specific query language to make it possible to send queries and receive results. This is provided by the SPARQL query language and the accompanying semantics and protocols [7]. SPARQL queries are similar in syntax to SQL but are based on the RDF triple models and provide patterns
against such relationships in which some resource references are variables. A SPARQL engine would return the resources that match these patterns for all triples.

For example, consider the below SPARQL query in which we ask for all the Road names in Eveleigh:

```sparql
PREFIX ns: <http://road.com/>
SELECT ?name
WHERE
{
  ?road ns:inSuburb "Eveleigh".
  ?road ns:hasName ?name.
}
```

The results would be:

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Garden St”</td>
</tr>
<tr>
<td>“Central Ave”</td>
</tr>
</tbody>
</table>

SPARQL offers a complete set of read-only query semantics for operators, regular expressions, pattern matching, ordering, etc. that are framed around the RDF triple model. Under current development are a SPARQL Update language, SPARQL HTTP bindings, and discovery mechanism for describing the capabilities of a SPARQL endpoint.

For further technical information, see the SPARQL Query Language [8].

### 2.2.4. RIF

The Rule Interchange Format (RIF) family of specifications [9] focuses on exchange rather than a single one-fits-all rule language but even rule exchange alone is a complex and daunting area. Rule systems fall into three categories; first-order, logic-programming, and action rules, and these paradigms share little syntax and semantics.

RIF is a family of languages, called dialects, with rigorously specified syntax and semantics designed to be uniform and extensible. The core idea behind rule exchange through RIF is that different rule systems will provide syntactic mappings from their native languages to RIF dialects and back. These mappings are required to be preserve their original semantics so that systems can exchange data through a suitable dialect for which they both support.

Obviously RIF rules should be able to interface with RDF and OWL ontologies even though they are languages with dissimilar syntaxes and semantics. The “RIF-RDF and OWL Compatibility” specification defines how RIF uses its frame syntax to map onto RDF triples and joint semantics are defined for the combination.

For RDF, there is a mapping between the subject/predicate/object triple and RIF frame formulas so that whenever the triple is satisfied, the corresponding RIF frame formula (s[p’ -> o’]) is satisfied, and vice versa.

For example, consider the RDF graph that contains the following triples:

```rdf
ns:GardenSt ns:inSuburb ns:Eveleigh.
```
Then consider the RIF document that contains the rule:

\[
\text{Forall } ?x \ ?y \ ?z (\ ?x[\text{ns:jurisdictionOf} \rightarrow \ ?z] \rightarrow \\
\text{And}(\ ?x[\text{ns:inSuburb} \rightarrow \ ?y] \ ?y[\text{ns:inCity} \rightarrow \ ?z]))
\]

which says that whenever some x is the the Suburb of some y and y is in the City of some z, then x is in the Jurisdiction of z. From this, the RIF frame formula :GardenSt :jurisdictionOf :Sydney, as well as the RDF triple :GardenSt :jurisdictionOf :Sydney, are outcomes of this RIF rule.

For further information and links to the family of specifications, see the RIF Overview [10].

### 2.2.5. GRDDL

The Gleaning Resource Descriptions from Dialects of Languages (GRDDL) specification introduces markup based on existing standards for declaring that an XML document includes data compatible with the RDF and for linking to algorithms (typically represented in XSLT), for extracting this data from the document [11].

GRDDL provides a set of mechanisms for bootstrapping RDF content from XML. GRDDL does this by via transformation algorithms for XML markup (such as XHTML) to allow the formulation of RDF statements from HTML pages. GRDDL defines a number of common transformations, as well as the ability to define your own.

In the HTML example below, a GRDDL transformation has been added to “glean” the title of the document.

```html
<html xmlns="http://www.w3.org/1999/xhtml"
     xmlns:grddl='http://www.w3.org/2003/g/data-view#'
     grddl:transformation="glean_title.xsl http://www.w3.org/2001/sw/grddl-wg/td/getAuthor.xsl" >
    <head>
      <title>Road Traffic Updates</title>
    </head>
    ...
</html>
```

Which will then produce this RDF/XML output:

```xml
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <rdf:Description rdf:about=""> 
    <dc:title>Road Traffic Updates</dc:title>
  </rdf:Description>
</rdf:RDF>
```

For further technical information, see the GRDDL Primer [12].

### 2.2.6. POWDER

The Protocol for Web Description Resources (POWDER) specification supports the provision of description resources and also a way to apply them to groups of online resources and for the authentication of those descriptions [13]. There are two versions of POWDER, one simple version expressed in XML, and the other that harnesses the Semantic Web languages (POWDER-S). A Description Resource is a claim from someone making a statement about a
given resource, or group of resources. Since a Description Resource may be published by anyone, anywhere, to describe anything, the end consumer may wish to ask the author more details about the claim in order to establish trust over the description.

POWDER is useful from a content providers point-of-view as it is an easy and inexpensive way to describe many resources at once (such as content on a Web site), is capable to support authentication to increase the trust level of your data, and opens up your data to the Semantic Web services.

The example snippet below shows how to add simple POWDER description resources to describe all resources at the given URI.

```xml
<d>
  <iriset>
    <includehosts>http://road.com</includehosts>
  </iriset>
  <descriptorset>
    <displaytext>Road Traffic Information</displaytext>
    <displayicon src="http://road.com/icon.png" />
    <dc:copyright>road.com pty ltd</dc:copyright>
    <geo:jurisdiction>QLD</geo:jurisdiction>
  </descriptorset>
</d>
```

The next example snippet below shows how to add semantics to an ontology to include information leading to greater trust outcomes (using POWDER-S).

```xml
<owl:Ontology rdf:about="http://road.com">
  <powder:issued>2010-01-01T00:00:00</powder:issued>
</owl:Ontology>
```

For further technical information, see the POWDER Primer [14].

### 2.2.7. SKOS

The Simple Knowledge Organization System (SKOS) is a common data model for sharing and linking knowledge organization systems, such as thesauri, taxonomies, classification schemes and subject heading systems. SKOS explicitly captures much of the similarity across these systems to enable data sharing across applications [15]. The goal of SKOS is not to replace existing ontology languages, but enable vocabularies to be mapped and shared for greater interoperability.

The fundamental element of the SKOS vocabulary is the “concept”, which is an abstract entity. A collection of concepts can be grouped into a “Concept Scheme”. SKOS also supports:

- Labelling properties with preferred, alternative and hidden terms.
- Broader and Narrower properties to enable the representation of hierarchical links, including explicit transitive relationships.
- Related properties to enable the representation of associative (non-hierarchical) links.
- Mapping properties to indicate how concepts compare.
- Ordering of concepts.
• Documentary properties to capture notes, definitions, examples, etc.

The following example shows how a number of SKOS concepts have been defined with definitions and preferred labels. The Street and Avenue concepts are defined as having the broader concept of Road. The City concept has the narrower concept Suburb and is related to the Jurisdiction concept. The Intersection concept has been mapped to two external vocabularies as an exact match to Junction and close match to Crossroad.

```xml
ns:road rdf:type skos:Concept;
skos:definition "specially prepared surface that vehicles can use";
skos:prefLabel "Rd".

ns:street rdf:type skos:Concept;
skos:prefLabel "St";
skos:broader ns:road.

ns:avenue rdf:type skos:Concept;
skos:prefLabel "Ave";
skos:broader ns:road.

ns:city rdf:type skos:Concept;
skos:narrower ns:suburb.
skos:related ns:jurisdiction

ns:intersection rdf:type skos:Concept;
skos:prefLabel "Intersection";
skos:exactMatch ns2:junction.
skos:closeMatch ns3:crossroad.
```

For further technical information, see the SKOS Primer [16].

2.2.8. RDB2RDF

The goal of RDB2RDF is to standardise a language for mapping relational data and relational database schemas into RDF and OWL - tentatively called the RDB2RDF Mapping Language (R2RML) [17]. Given that a significant amount of data resides in relational databases, it makes sense to provide a view of that data for Semantic Web tools and services to manipulate. Hence the need to provide an RDF mapping to RDB schemas. The RDB2RDF working group have not completed any final specifications as of yet, focussing initially in the requirements and use cases to drive the future development.

For further information, see the RDB2RDF Use Cases [18].

2.3. Additional Semantic Web Technologies

The above technologies are a core part of the W3C Semantic Web Activity. In the wider community, numerous technologies and services have been developed using the core Semantic Web languages.

• Dublin Core - a popular set of properties to describe documents and related resources, such as images and movies [19].

• DBpedia - publishes the structured data extracted from Wikipedia in RDF/OWL thus allowing Semantic Web agents to provide inferencing and advanced querying over the
Wikipedia-derived dataset and facilitating interlinking, re-use and extension in other data-sources [20].

- vCard RDF - maps the popular vCard standard into RDF enabling semantic web applications [21].
- FOAF - Friend of a Friend uses RDF to describe the relationships people have to other people and the things around them [22].
- GoodRelations - an e-commerce vocabulary for expressing product information, prices, payment options, etc. It also allows expressing demand, and tendering / request for quotation data. GoodRelations has been adopted by BestBuy, Yahoo, OpenLink Software, O’Reilly Media, the Book Mashup, and others [23].
- SIOC - the Semantically-Interlinked Online Communities vocabulary of terms and relationships that model social web data spaces, such as discussion forums, blogs, feed subscriptions, mailing lists, shared bookmarks, and image galleries [24].
- Sindice - is a search engine for ontologies, documents, terms and data published in Semantic Web formats. Sindice employs a system of crawlers to discover RDF documents and embedded RDF content [25].
- Basic Geo Vocabulary - is an RDF vocabulary that provides the Semantic Web community with a namespace for representing latitude/longitude and other information about spatially-located things, using WGS84 as a reference datum [30].
- Notation 3 (N3) - is a Semantic Web inspired compact syntax aimed to optimise the expression of data and logic in the same language. N3 allows RDF to be expressed, allow rules to be integrated smoothly with RDF, allows quoting so that statements about statements can be made, and aims to be as readable, natural, and symmetrical as possible [44].

In addition to vocabularies and software tools, the Semantic Web community also provides many online free services to help build and foster the adoption and Semantic Web technologies:

- Semantic Overflow (http://www.semanticoverflow.com/) is a online Question and Answer service where many experts will respond to any Semantic Web related questions with the best answers rewarded.
- Linked Open Data Around the Clock (http://latc-project.eu/) has assembled and maintains a library of open source Linked Data tools and provide a data source inventory to the community supporting both institutions as well as individuals with tutorials and best practices concerning Linked Data publication and consumption.

2.4. Maturity Analysis

For the layperson, the Semantic Web is still in its infancy. The range of new technologies have not yet matured to the point that the typical web user could understand their purpose and features and very little friendly tools are available to mask their complexity. For example, OWL’s transitivity mechanism is a powerful feature for the professional ontologist, but not meant for the typical end user.

The Semantic Web is not a new Web and will not reach the same user impact as technologies like HTML did for typical web users. The Semantic Web is a professional web aimed at the
enterprise market where there is a business case to improve the way information is managed and decisions made. Some Semantic Web techniques, such as Linked Data, will be more end user focussed, but the majority of Semantic Web technologies, such as OWL and RIF, are aimed at back-end processes to increase interoperability and transparency of information modeling and reasoning.

A number of industry analyst reports are now beginning to recognise Semantic Web technologies and companies. Even though these reports do not explicitly mention the phrase “Semantic Web”, they do mention companies, such as Sinequa and Expert System as niche players, as well as the market leaders, such as Google and Microsoft [26].

The PEW report [27] is one the few industry reports that directly challenges the Semantic Web agenda. Some 895 industry people (including 371 “experts”) responded to the questionnaire to predict the likely progress toward achieving the goals of the semantic web by the year 2020. There was a slight majority (47%) who thought the Semantic Web would not meet its objectives, versus 41% who thought it would. Among the experts, the numbers were 52% against and 38% for.

The major theme that emerges from the report is:

“This too many complicated things have to fall into place for the semantic web to be fully realized. The idea is a noble one and gives the technology community something to shoot for. But there is too much variation among people and cultures and economic competitors to allow for such a grand endeavour to come to fruition”

Although it sounds less than promising, what the view highlights is that it is critically important to understand your business needs first before embarking on adopting semantic web technologies.

However, another Technology Radar report [28] which shows emerging trends that affect the market today places “RDF” in the “assess” phase meaning it is “worth exploring with the goal of understanding how it will effect the technology impact dimensions of your enterprise”.

2.5. Semantic Web Case Study

A good example of this exploration was the recent use of Semantic Web technologies by the BBC for the World Cup Football 2010 coverage [29]. The system was powered by a high-performance dynamic semantic publishing framework based on a rich ontological domain model. The ontology described all the entities and relationships pertinent to the World Cup. For example, "Harry Kewell" is part of the "Australian Squad" and the "Australian Squad" competes in "Group D" of the "FIFA World Cup 2010".

The ontology also includes journalist-created content (eg stories, blogs, profiles, images, video and statistics) and how they were related to the World Cup entities. Hence, you can quickly search for all blogs about "Harry Kewell".

Figure 2.5 shows a high-level overview of the architectural components of this semantics-based framework. Metadata is captured using RDF representations and triple store technology and made persistent for querying with SPARQL. The driving goal was that the domain ontology allowed for smart mapping of journalist content to entities and queries and automatically derived inference statements.
For example, if the "Harry Kewell" entity was selected then entities such as "Australian Squad", and "Group D" are generated triples within the triple store. It was found that this inference capability was simpler and quicker than traditional SQL approaches and the inferred statements produced higher quality and greater breadth of content. Similarly, it was found that the RDF approach promoted more agile modeling over traditional relational schema modeling.

The BBC is now looking at expanding the capabilities of this Semantic Web inspired system to possibly cover all sports, more geographically aware assets, and alignment with news stories. They do see their semantic path as a key driver for the London 2012 Olympics.
3. Semantic Web Architectures

An Enterprise Architecture is a rigorous and complete description of an organisation and how it is decomposed into sub-systems, how the sub-systems are related and dependent, the terminology used, the external environment, and the principles and goals of the organisation. An Enterprise Architecture presents an organisation with the unique circumstances to identify opportunities for improvement and to better meet its overall objectives.

The most common layers of an Enterprise Architecture include:

- Business - strategic documents, business processes, capabilities.
- Information - metadata, information flow, data models.
- Applications - software, functional systems, application interfaces.
- Technology - middleware platforms, network infrastructure, programming languages.

To date, most Enterprise Architectures have not utilised nor deployed, in any systemic way, Semantic Web technologies.

An example of an Enterprise Architecture is from the Australian Government Information Management Office (AGIMO) in which they define five layers of the Australian Government Architecture Reference Model [31] as shown in Figure 3.1.

Each of the layers represent and is further expanded into a new reference model. For example, the Data Reference Model, shown in Figure 3.2, shows it’s three sub-systems:

- Data Description - a means to uniformly describe data.
- Data Context - the categorisation of data according to taxonomies.
- Data Sharing - access, query and exchange of data between parties.

The Data Reference Model further describes and decomposes the three subsystems into an Abstract Model, and importantly, how they interact and are related, as shown in Figure 3.3.
The Data Description area of the abstract model identifies the various data types and their interrelationships. The focus of this area is the identification of entities and the designation of the information describing them. The Data Context area of the abstract model identifies the structures used for resources in the form of a set of terms collectively called ‘categorisation’ or ‘classification’ taxonomies. The reference model, at this point, then refers to the Semantic Web with “implementation of Taxonomies could take the form of XML Topic Maps, Web Ontology Language (OWL) hierarchies or ISO11179 Classification schemes”. Finally, the Data Sharing area of the abstract model conveys an architectural pattern for the sharing and exchange of data with examples to support their business needs.

Even with a cursory analysis, after looking at the entities in Figure 3.3, we can immediately see where some of the Semantic Web technologies could form part of the core features of the Data Reference Abstract Model:

- Data Description: RDF
- Data Context: OWL
- Data Sharing: SPARQL

In this case, there is a promising opportunity to overlay Semantic Web technologies onto the Australian Government's enterprise architecture.

However, in general, there has been little comprehensive, and large-scale, attempts to create Enterprise Architectures utilising Semantic Web technologies as the foundation information model and technology platform. There are some small pockets of activity in this space, ranging from individuals (eg the "Layered Semantic Enterprise Architecture" [32]) to niche companies.
promoting the Semantic Web as part of their architecture platform (e.g., “Towards Executable Enterprise Models” from TopQuadrant [33]).

3.1. Semantic Web Tools

In order to work with Semantic Web technologies, tool support is essential. There is a large range of tools currently available to cater for RDF triple stores, inference/reasoning engines, RDF generators/converters/validators, SPARQL databases, search engines, content management systems, semantic web browsers, development environments, and semantic wikis. For an exhaustive list and links, see [34, 35].

The most widely used RDF/OWL editor is Protégé, a free open-source editing framework developed at Stanford University. It has an open plugin structure for the easy integration of special-purpose ontology editing components, as shown in Figure 3.4. Other open-source systems include SWOOP and NeOn-Toolkit, and commercial editors include TopQuadrant’s TopBraid Composer.

There are a number of RDF triple store systems including the open source Mulgara scalable RDF database written entirely in Java, 4Store a MySQL based triple store, Openlink Virtuoso an innovative enterprise grade multi-model data server, and ARC a flexible RDF system for PHP practitioners. Commercial systems include the Talis Platform, a hosted data service providing content and triple storage, along with various auxiliary services including change management, access control, index/search and SPARQL.

There are several reasoners available under open source including HermiT an open-source Java based OWL 2 DL reasoner, FaCT++ an OWL DL Reasoner implemented in C++, and Pellet is an open-source Java-based OWL DL reasoner.

There are a number of online validators, such as the W3C RDF Validator, Vapour the linked data validator, and the WonderWeb OWL-DL validator.

RDF creation tools include the Ontos API which is a public web service which returns semantic metadata in standard RDF-based formats for any text input. Ontos recognises entities and relations between them using natural language processing techniques. Open Calais is a web service that automatically attaches semantic metadata to a document. Using natural language processing and machine learning, Calais categorises and links the document with entities, facts, and events with the results stored centrally and returned as RDF statements.
A few search engines have been specifically developed to crawl and index semantic web data (primarily RDF) including Sindice, Swoogle, and Watson.

There are a number of plug-in tools for web browsers to support additional Semantic Web functionality. OWLSight is an OWL ontology browser that runs in current web browsers, SIOC Browser is a RDF browser for exploring information expressed in RDF and the SIOC ontology in particular, and Semantic Radar displays a status bar icon to indicate presence of Semantic Web (RDF) data in a web page.
4. **Scenarios**

Semantic Web technologies clearly thrive in complex domains with rich ontological needs and diverse stakeholders. Intelligent Transport Systems (ITS) is such a domain related to the application of a wide range of ICT technologies to manage the safety of the transport systems and enhance the travellers experience. Any ITS enterprise architecture would need to include and support services such as:

- Traffic Management
- Freight and Fleet Management
- Public Transport and Traveller Information
- Emergency Management and Safety Services
- Toll Payments
- Weather and Environment Information

This is a broad range of services and would typically involve many stakeholders across Government levels and industry providers. Clearly, this scenario would benefit from shared semantics across all these stakeholders in providing these services in the most effective and efficient manner. Each stakeholder would typically have their own ICT system to support these services so mechanisms are required to ensure optimal service collaboration.

4.1. **Commercial ITS Systems**

Any commercial ICT system for ITS would need to communicate and expose its semantics to other stakeholders, as this is a core requirement for interoperability.

For example, the DYNAC Advanced Traffic Management System [36] software support many of the features listed above. Figure 4.1 shows a window from its Decision Support Manager module in which an incident is being reported. As can be seen, there are a number of predefined Event Types (eg “Substation”) and Classifications (eg “Transformer Failure”, “Ground Fault”, etc) for the reported incident. These would need to be understood and compatible with external systems that the incident is reported to. Clearly, such terms would be from a common vocabulary defined and shared by the stakeholders prior to deployment. Semantic Web technologies would provide the key mechanisms for this sharing of semantics.
Another service supported by ITS systems is real-time travel advisories to enhance the safety of motorists. Some of the information required for these would be current weather reports from the responsible meteorological authorities. The Australian Bureau of Meteorology issues weather observations every 10 minutes, for example, see [37] for the latest weather observations for Sydney Harbour which are also available in programming-language specific machine-readable format (a snippet is shown below). This data could be fed live into an ITS system to assist in the creation of travel advisories for Sydney Harbour Bridge drivers if certain thresholds are met.

```json
{
  "name": "Sydney Harbour",
  "history_product": "IDN60901",
  "local_date_time_full": "20100707123000",
  "air_temp": 13.2,
  "gust_kmh": 26,
  "gust_kt": 14,
  "rel_hum": null,
  "wind_dir": "WSW",
  "wind_spd_kmh": 22,
  "wind_spd_kt": 12
}
```

This data could also be easily translated into RDF syntax. More importantly, the semantics of the properties need to be clear and shared. For example, “gust_kmh” and “gust_kt” represent the Wind Gust in both km/h and knots. This is probably not the best way to model these properties as the information is duplicated and the datatype is part of the property name. Nevertheless, the data still could be mapped into the “ITS Semantic Information Architecture” and used to inform the real-time travel time advisories module.

### 4.2. ITS Enterprise Architectures

More formal Enterprise Architectures have been developed for ITS. The US Department of Transport (DOT) ITS Architecture [38] includes the following logical functions:

- Manage Traffic
- Manage Commercial Vehicles
- Provide Vehicle Monitoring and Control
- Manage Transit
- Manage Emergency Services
- Provide Driver and Traveler Services
- Provide Electronic Payment Services
- Manage Archived Data
- Manage Maintenance and Construction

The functional specification is expressed in graphical form as Data Flow Diagrams (DFDs) and in more detailed textual form as Process Specifications (P-Specs) and data flow descriptions. The DFDs provide a convenient view of the structure of the requirements and they consist of lines showing data flows and circles that represent either a P-Spec, or a lower level DFD. The P-specs...
provide the core of the requirements in that they describe how data that flows into the logical architecture is transformed either for use elsewhere, or for output from the overall Enterprise Architecture.

The DOT ITS Enterprise Architecture is extremely detailed and intricate, reflecting the complexity of the domain it covers. Figure 4.2 shows the Manage Incidents DFD which is a sub-part of the Manage Traffic DFD. These processes manage the classification of incidents and implement responses when they actually occur. The key points about this DFD include:

- Incident inputs available from many sources, e.g. emergency services, event promoters, etc.;
- Data from traffic sensors analysed for indications of possible incidents;
- Traffic operations personnel can update incident data with other inputs, e.g. telephone reports;
- Incident inputs from construction and maintenance organisation must be confirmed;
- When an incident occurs predefined mitigation strategy is automatically implemented;
- Predefined incident strategies can be set up in advance by the traffic operations personnel;
• Incident status provided as input to the predictive traffic model and to vehicle route selection criteria;
• Information on incidents is available to the media.
• Provide disaster response and evacuation support in the form of coordination with other traffic management centres, traffic operations personnel and emergency management.

These functions of the ITS Architecture would be invoked for our original simple scenario; “A traffic accident at the corner of Garden St and Central Ave in Eveleigh, Sydney”. For example, the large circle in Figure 4.2 is “Process Specification 1.3.2 - Review and Manage Incident Data” which has one of it many output flows entitled “roadway detours and closures for traffic” to ascertain if traffic diversions are needed for the incident in Eveleigh.

4.3. **Amber Alert**

One feature of ITS systems is its ability to control variable traffic signs to inform the traveller of conditions and incidents ahead. These facilities are also being used for Amber Alerts to help in the case of kidnapped children. Realtime information about the incident (such as the car make, colour, registration number) can be sent to variable traffic signs in the local area to help locate the child from the general public. There are common guidelines for such incidents [39] for Transport Authorities to follow and typically the source information will come from trusted authorities (such as Police).

One new approach [40] has been to send Amber Alerts as part of the Common Alerting Protocol (CAP) [41], as shown in the below snippet.

```xml
<alert xmlns="urn:oasis:names:tc:emergency:cap:1.2">
  <identifier>urn:nsw:police:au:alert:3212121</identifier>
  <sender>sydney@nsw.police.gov.au</sender>
  <sent>2010-06-06T11:30:00</sent>
  <status>Actual</status>
  <msgType>Alert</msgType>
  <scope>Public</scope>
  <info>
    <category>Rescue</category>
    <event>Child Abduction</event>
    <urgency>Immediate</urgency>
    <severity>Severe</severity>
    <certainty>Observed</certainty>
    <headline>AMBER ALERT</headline>
    <description>VICTIM: JANE DOE. FEMALE. 110CM. BLONDE HAIR. WEARING RED SHORTS BLUE JEANS. SUSPECT VEHICLE: GREEN FIAT 500. REGO# NSW 555-555</description>
    <contact>Crime Stoppers 1800 333 000</contact>
    <area>Eveleigh, NSW, Australia</area>
  </info>
</alert>
```

CAP defines an XML format for interoperability in alerting and public warning systems. The intention is to promote consistency in the information produced by all kinds of sensor and alerting systems, thereby reducing confusion and helping to get crucial warning information to the public.
faster. CAP messages carry message identifiers; information about the sender and the time sent; message status, type and scope; and the event category, urgency, severity and certainty. In addition, the messages can carry other optional information, such as instructions for the recipients and a description of the target area. CAP has had good early uptake in the US and is emerging as the common information standard for general incident messages.

CAP defines value semantics for critical properties such as Severity: Extreme, Severe, Moderate, Minor, and Unknown. Such properties are used in the decision making process (both manually and automatically) so it is critically important that their semantics are clear, concise, and commonly understood for actionable purposes.

A typical scenario may see a Transport Authorities required to support the IEEE 1512.1 Standard for Traffic Incident Management Message Sets for Use by Emergency Management Centers [42] and also be a consumer of CAP messages from external agencies. IEEE 1512.1 also defines value semantics for critical properties. They have a similar property to CAP Severity with the name “Event-incident-severity” as defined in the below snippet:

```xml
<xsd:simpleType name="Event-incident-severity" >
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="noAdditionalInformation"/>
            <xsd:enumeration value="otherAdditionalInformation"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="minor"/>
            <xsd:enumeration value="major"/>
            <xsd:enumeration value="naturalDisaster"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:simpleType>
```

This IEEE property could be related as a “close match” to the CAP property under the SKOS framework. There are some exact matches in value semantics such as “minor” and “unknown”, but the other values in both properties do make it difficult to perform mappings and will require some innovative application-level semantics as there are some in-built semantics to values such as “naturalDisaster”.

Figure 4.3 shows a more detailed mapping using RDF, OWL and SKOS primarily. In this example, we assume the Transport Authority has an existing Severity vocabulary of:

- Critical
- Serious
- Medium
- Minimal

The modelling and mapping to global vocabularies would inform the enterprise information architecture as to how well the business can address external inputs, where the missing semantics are, and a consistent path to address these. In this example, it is clear the enterprise information architecture needs to be updated to address the business needs of dealing with CAP and IEEE messages from stakeholders.
FIGURE 4.3 - SEVERITY SEMANTIC MODEL
5. Conclusion

5.1. Summary

The Semantic Web is a collection of technologies for knowledge representation and sharing of semantics through mechanisms like Linked Data. W3C has created the technology base with RDF and OWL as generic languages for modelling complex information relationships. Communities representing domains-of-interest are building on these base technologies and creating simple vocabularies and advanced ontologies to help provide consistency across organisations as well as between organisations.

The Semantic Web offers plenty of opportunities for Enterprise Architectures. Given one of the core principles of Enterprise Architectures is a shared vision of an organisation, such a vision needs to be expressed in clear, common, and precise semantics. RDF and OWL provide the formal ability to provide this objective.

The Semantic Web can help drive Enterprise Architecture design by:

- Improved information consistency with comprehensive ontology support, mapping, and rules (ie RDF, OWL, SKOS, RIF)
- Simplifying the effort for data provisioning with interoperable standards (ie SPARQL, POWDER)
- Providing legacy support for data sources (ie GRDDL, RDB2RDF)

However, the Semantic Web should not be seen as a new panacea for Enterprise Architectures. Consideration for some of its deficiencies (not by design) must be addressed. For example, the Semantic Web is not typically seen as a “Business Processing/Modelling Language” as there are other stronger candidates for this role. Nevertheless, the Semantic Web can still play a role in consistently defining the entities used in such workflow languages, and RIF could be investigated as a rule-based alternative.

The best approach for adopting Semantic Web technologies is to start small and bottom-up. New initiatives and projects should utilise Semantic Web methods and technologies, and then provide interfaces to legacy systems. For example:

- Undertake a “Terminology Review” across the organisation identifying all the entities used and, more importantly, where they conflict and need alignment.
- Similarly, undertake an “External Terminology Review” of all incoming information and data from outside sources.
- Model the two sets in appropriate Semantic Web tools and investigate alignment.
- Prove the new ontologies (in RDF/OWL) in a small project (utilising open-source tools) and the benefit to the wider organisation.
- Championing and share the ontologies across the organisations and external stakeholders.

The Semantic Web offers many choices to organisations and the best and most appropriate parts should be adopted that align with the business interests of the enterprise.
5.2. Future

Like many ICT technologies before, predicting the future is simultaneously precarious and auspicious. The Semantic Web is no different and has both been criticised for being too complex and "academic" and showered with praise for its new “exciting” technologies that drive web-based user experiences.

The potential future path of the Semantic Web may include:

- Redefined RDF language (now being discussed [43]).
- Merge the RDF and OWL languages into one comprehensive language with various levels of semantic support.
- Support a range of simple (eg "microdata") and complex syntaxes seamlessly.
- Support greater Vocabulary/Ontology management.

The last point is one of the "killer opportunities" for the Semantic Web, and will test how mature and useful the range of technologies are in reaching out to the wider community. A recent report [46] indicated that “vocabulary development was still seen very much as a niche activity undertaken by those with some degree of prior specialist experience” and there was a “pressing need to provide more stable, robust and persistent hosting for vocabularies” and a “key requirement for driving vocabulary usage is the availability of vocabulary discovery services”.

It will be a long journey before the Semantic Web becomes a core and essential Web technology in all enterprises and sectors, perhaps up to 10 years in some cases. However in some cases, such as Health and Life Sciences, the business need and technology match has proven clear and beneficial early results [47], and this is a good indicator that the technology set has wider potential. Nevertheless, adopting the principles fundamental to the Semantic Web, where there are complex data sets and a prominent value of highly structured ontologies, will place an enterprise in a good position for improved information management and stakeholder interoperability in the long term.
6. References

[6] OWL 2 Web Ontology Language Primer <http://www.w3.org/TR/owl2-primer/>
[8] SPARQL 1.1 Query Language <http://www.w3.org/TR/sparql11-query/>
[10] RIF Overview <http://www.w3.org/TR/rif-overview/>
[12] GRDDL Primer <http://www.w3.org/TR/grddl-primer/>
[14] POWDER Primer <http://www.w3.org/TR/powder-primer/>
[16] SKOS Primer <http://www.w3.org/TR/skos-primer/>
[18] RDB2RDF Use Cases <http://www.w3.org/TR/rdb2rdf-ucr/>
[20] DBpedia Website <http://dbpedia.org/>
[23] GoodRelations eCommerce Ontology <http://purl.org/goodrelations/>
[30] Basic Geo (WGS84 lat/long) Vocabulary <http://www.w3.org/2003/01/geo/>
Towards Executable Enterprise Models: Building Semantic Enterprise Architecture Solutions with TopBraid Suite™

Semantic Web Tools

Semantic Web Development Tools

DYNAC Advanced Traffic Management System

Bureau of Meteorology Latest Weather Observations for Sydney Harbour

US Department of Transportation National ITS Architecture Version 6.1

AMBER, Emergency, and Travel Time Messaging Guidance for Transportation Agencies.

Publish critical public warnings on the Web, IBM Developerworks.

OASIS Common Alerting Protocol version 1.2


W3C Workshop — RDF Next Steps Workshop Report, 2010

Notation 3.

RDFa Primer

Dagstuhl Perspectives Workshop Report: Semantic Web Reflections and Future Directions, 2010

Semantic Web and eHealth Whitepaper, Semantic Identity, January 2013