Information Architecture for Linked Open Data

April 2017

SEMANTIC IDENTITY
OBSERVE REASON IMAGINE
1. Linked Open Data

The age of the Semantic Web and Linked Open Data (LOD) has the potential to create a new breed of information system that delivers enhanced data features to a wider audience of users. LOD promotes the publication of data in open formats for easier accessibility by machines and visualisation for human users.

LOD is premised on the principles of using URIs to name things, using HTTP to retrieve data, and using RDF to encode/model entities. A key feature is to provide as much linking between entities as possible in the data, and since URIs are used almost exclusively, then a larger web of machine-interoperable and inter-related data is naturally formed.

LOD has been very successful in some sectors (such as government, smart cities), a significant number of community projects, and is becoming more important in search engine services. The number of repositories of LOD grows each year, as shown in the "LOD Cloud" with each node representing thousands, if not millions, of LOD records [1]. The growing and inter-related web is clearly shown in the LOD Cloud as well as the linkages between LOD datasets.

LOD is also a business strategy to exploit datasets. Opening up data for reuse by others promotes better and effective collaboration and supports innovative new data sharing opportunities. LOD can also inspire greater customer engagement by opening and sharing data to third parties to build new data services. Government are seen as a key data source for LOD as they collect and manage a vast array of data that benefits the public, but typically is under-utilised.

LOD also needs to respect privacy and should never publish any personal identifiable data without explicit and robust access control mechanisms. In addition, LOD is not always "free" as in open to anyone to reuse for any purpose, so some licensing conditions need to be expressed and honoured by third parties.
Tim Berners-Lee proposed a 5-Star rating systems for LOD as a way to encourage "good linked data" [15]. These include:

- ★ Available on the web (whatever format) but with an open licence
- ★★★ Available as machine-readable structured data (eg excel instead of image scan of a table)
- ★★★★ As (2) plus non-proprietary format (e.g. CSV instead of excel)
- ★★★★★ All the above plus, Use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff
- ★★★★★★ All the above, plus link your data to other people’s data to provide context

These are very useful guiding principles and enables the LOD provider to target different states in their evolution in publishing datasets.

However, these principles do not focus on the actual data itself, only how the data is made available and what formats. For example, you can treat RDF as a serialisation rather than a modeling approach and end up with significant amounts of "5-Star" open data but they may have very little "semantics" to reason with as the underlying information model may simply be based on structure only.

As we will see in the following sections, many examples of open data today are aimed at 3-star LOD - typically as Comma Separated Values (CSV) - as this seems to be the lowest friction point for vast amounts of data with low technology requirements. The challenge then is ensure that the data is also not modelled at this lowest-point, that is, a flat list of columns in a spreadsheet.

The current mantra is to support and publish LOD (such as the 5-Star principles) and that, in itself, will deliver advanced outcomes. However, this will not happen effectively if the underlying Information Architecture does not reflect the target semantic information model in which such advanced services can be supported.

2. Information Architecture

Information Architecture (IA) has always been the poorer cousin in enterprise architecture frameworks, such as The Open Group Architecture Framework (TOGAF) [2] and the Object Management Group - Model Driven Architecture (OMA-MDA) [3]. The reason is that the other enterprise viewpoints are mostly more objective in that they cover the computational, application, technology, implementation, and engineering layers. Or they are more visionary, such as the enterprise or business layers.

IA deals with, and focusses on, features the system must perform for end users, the data needed for these features, and the processing of this data (and related data) to inform or make business decisions. IA is the heart and brain of any information system and must address data modelling and data semantics as its core activity.

This is never easy as (typically) the information architect does not have the full picture of the end users desired requirements. Hence, the model and semantics represent a partially-closed world. This is partly the problem of end users not always
being aware of exactly what they want, a lack of understanding of what to do with linked open data, and designing a solution that is based on historical practices.

Enterprise architecture is often considered to be analogous to the building industry. Blueprints represent architectures for the different layers of a house (foundation, walls, utilities, roof, etc) and (usually) it all works. Typically because a home-owner has a good grasp of the features that a house provides. They know what, where, and how a laundry will work in the house. It won't be in the bedroom and won't drain water to the wrong waste pipes. These are all clear and obvious in the architectural drawings. But IA does not have the same capability. It is difficult to "see" what the data will do, where it lives, how it interacts, and if it is doing the "wrong" thing. That is, unfortunately, until it is too late.

One aspect of enterprise frameworks is the emphasis on a requirements-driven approach for the design of architectures. TOGAF recommends these steps in developing an enterprise information architecture (Phase C - Information Systems Architectures - Data Architecture in [2]):

- Select reference models, viewpoints, and tools
- Determine overall modeling process
- Identify required diagrams
- Develop baseline data architecture description
- Develop target data architecture description
- Perform gap analysis
- Define candidate roadmap components
- Resolve impacts across the architecture landscape
- Conduct formal stakeholder review
- Finalise the data architecture
- Create architecture definition document

This set of steps and deliverables are conducive to consistent IA outcomes and its iterative nature ensures opportunity for stakeholder review and input.

3. Example LOD Use Case

One of the most interesting LOD projects is describing the locations (and other features) of public toilets across countries, states, and cities. Many UK councils [4], Australia [5], and New York State [6] publish such data for example. This sounds like a simple task, but it is not. The UK experience has reported major issues with data format incompatibility, licensing of data, and lack of data updates [7].

The most poignant finding was; "Local authorities don't know whether to assign this task to their Open Data people, who don't know much about toilets, or to their toilets people, who don't understand much about data issues."

And herein lies the problem. Data people will see toilets as a bunch of data needing to be encoded and put into a database (with a sparql endpoint!), and local authorities
will look at it from a completely different management and user perspective. For example, they may have these types of viewpoints:

- **Camper** - I am travelling around in my camper van and need to discover public toilets in some of the most remote parts of the country.
- **Asset Manager** - My department maintains public toilets and manages the opening/closing and cleaning.
- **Policy Officer** - I am responsible for public policy and need to report on accessibility and ensuring gender equality in public services.
- **Event Coordinator** - I organise marathons and need to plan routes that can utilise public toilets.

These four roles would be atypical candidates for more expressive use cases (or user stories) and the generation of detailed requirements that would enable Information Architects to design data models for these purposes.

### 4. Comparing LOD Data Models

It is clear from looking at the current published data about toilets and their schemas (shown below) that there is a relatively flat structure in the data model, and a different focus in each country (eg more management and location in the UK, more toilet details and accessibility in Australia, and very little data in the USA).

<table>
<thead>
<tr>
<th>National Toilet Map (Australia)</th>
<th>Northumberland Council (UK)</th>
<th>New York City (USA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtractDate</td>
<td>OrganisationURI</td>
<td></td>
</tr>
<tr>
<td>OrganisationLabel</td>
<td>ServiceTypeURI</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>ServiceTypeLabel</td>
<td></td>
</tr>
<tr>
<td>LocationText</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>LocationText</td>
<td>Location</td>
</tr>
<tr>
<td>Town</td>
<td>Borough</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Postcode</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoordinateReferenceSystem</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Latitude</td>
<td>GeoX</td>
<td></td>
</tr>
<tr>
<td>Longitude</td>
<td>GeoY</td>
<td></td>
</tr>
<tr>
<td>GeoPointLicencingURL</td>
<td>GeoAreaURI</td>
<td></td>
</tr>
<tr>
<td>GeoAreaLabel</td>
<td>GeoAreaLabel</td>
<td></td>
</tr>
<tr>
<td>InfoURL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>Category</td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DumpPoint</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FacilityType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AccessLimited</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What is clear from the existing data schemas is that there is no underlying "data model" as the focus has been on capturing a set of structural properties. There is also no clearly defined "semantic entities" in the structures, such as places or things of interest. In addition, some of the key properties join two separate concepts (such as gender and accessibility).

Some properties should actually be values of properties (e.g., MLAK, and RADARKeyNeeded are types of keys used to open locks) and are obviously not clearly defined by using uncommon acronyms. So, semantically, the data structures are poorly modelled and documented.
Many of the values for each property (especially in the Australian case) are booleans. In most other cases, the values are all text literals. This leaves little room for (open) linked data to play any role as no URI linkages can be formed.

In addition, there does not seem to be any reuse of concepts/terms/properties from any existing data models or vocabularies that are available in LOD vocabularies, such as Linked Open Vocabularies [17].

5. Semantic Modelling for LOD

Conceptual Data Model

If we first took a more conceptual view (and not rush into a list of properties, which is a common mistake) we should develop a high-level data model that represents the key requirements of the stakeholders. Figure 1 shows an example.

In our example, we will use the four roles described above which need support for discovery (camper), management (asset manager), compliance (policy officer), and planning (event coordinator). Each of the concepts in the model help towards meeting these priorities, for example:

- Discovery: Location, Availability
- Management: Administration, Location, Services, Availability
- Compliance: Gender, Accessibility
- Planning: Location, Availability, Administration, Services

An enterprise design process (such as TOGAF) would then recommend an iterative formal stakeholder review process of all architecture outputs. In this (simple) case, the
conceptual design model would be documented (with explanatory narrative, scenarios, etc) and reviewed and feedback used to produce any updates.

**Logical Data Model**

Developing the logical data model now requires greater attention to details as well as addressing stakeholder requirements and business drivers, such as supporting LOD. The conceptual model provide a framework for the logical model to dive deep into each entity and design detailed data models.

The Toilet entity is one that is unique in this domain. Describing toilets in the current state has focussed on boolean values (ie are there toilets or not). A better option (for all uses; discovery, management, compliance, and planning) is to indicate numerically their existence, as this also infers a boolean state.

A good way to model this is to look at the extreme case, since if we can model that with flexibility, then we should be able to model lesser cases. A good LOD strategy is not to define toilet types, genders, and accessibility - but to only define how they are related (in our context) and to reuse existing vocabularies (that could be further refined in the future).

We can model a Toilet Facility as having a number of Toilets that each can accommodate a fixed number of occupants. Each Toilet is designated a gender and accessibility information (eg, type, transfer sides, access key required, etc), as shown in Figure 2.

![Figure 2 - Toilet Data Example](image)

The Location entity is one where there are many existing data vocabularies and reuse should be straightforward (and there is no need to re-invent more ways to describe locations). Location will include street-like addresses as well as more accurate geospatial coordinates. Examples include the vCard Ontology [8] and OASIS...
Extensible Address Language [9]. For linked data, the best option would be to use the geoURI to enable direct links to geospatial long/lat coordinates [10].

The Availability entity requires the opening hours to be quite specific. Not only the hours per individual day, but also special days when it could be closed (e.g., public holidays). Text-based values (as in the examples in section 6) would make it difficult for systems to parse and query the data. The GoodRelations vocabulary has a vocabulary for opening hours that could be reused [11], as well as a new property to indicate exceptions when the facility is closed. The concept of Availability would also include if payments are required and/or access keys.

The Services entity would express the additional features of the Toilet facility, such as Baby Change, Showers, Sharps Disposal etc. The current boolean approach does not scalable nor does it support the open data approach. The better logical design is to explicitly represent the service as the object of the Service Entity. (Semantically, each service is a subclass of Service.) This approach would also enable additional properties about each service to be expressed, for example, if the Shower has hot and cold water.

For interoperability, the challenge would be to find existing vocabularies for these explicit services. DBPedia is one source where these could be defined [12]. There are URIs for:

- Shower: <http://dbpedia.org/resource/Shower>
- Sharp Containers: <http://dbpedia.org/resource/Sharps_container>
- Drinking Water: <http://dbpedia.org/resource/Drinking_Water>

Otherwise, if there are no suitable candidates, the term identifiers should be defined by managed URIs and published widely for others to reuse.

The Administration entity would express details on who is responsible for the facility and may include contact details. Again, reusing the vCard Ontology would be sufficient [8].

**IA Definition Document**

The IA design process and outcome will need to be fixated in the Information Architecture Definition Document for stakeholder review. The most appropriate visual modeling process and tools would dictate the development of the artefact.

The Unified Modelling Language (UML) is a common example for data models as it is independent from any implementation detail and has an industry-accepted and consistent visualisation [13]. One of the drawbacks of the Semantic Web is that there is no visual consistency in the tools that develop RDF/OWL ontologies, and stakeholders should not be making decisions from reviewing hundreds of lines of Turtle code.

Figure 3 is an example UML class diagram for the IA document (this is not complete and shows only a sample of the class attributes). Of course, it would be narrated with all the architecture and design decisions made by the IA team. For example, the
relationships between classes (such as Accessibility is related to one Toilet) and the reuse of vocabularies for some classes and attributes.

Often missing from IA documents is any detailed definitions of the properties. There is an assumption that the name of the property is sufficient to not only define it, but to also describes its usage. Use of a more systemic approach to documenting data elements such as ISO11179 is highly recommend [14].

As an example, for one of the properties in the model:

<table>
<thead>
<tr>
<th>Data Property</th>
<th>Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>The position of the grab rails in relation to the toilet</td>
</tr>
<tr>
<td>Obligation</td>
<td>Optional</td>
</tr>
<tr>
<td>Data Type</td>
<td>Enumerated String</td>
</tr>
<tr>
<td>Classification</td>
<td>&quot;Left&quot;, &quot;Right&quot;</td>
</tr>
<tr>
<td>Note</td>
<td>If both sides are supported, then the property should be expressed twice.</td>
</tr>
</tbody>
</table>

**Implementable Data**

At some point, this data needs to exist in formats required by the stakeholders. The best method is to map the data model into different serialisations. Each mapping will make decisions (tradeoffs) that are necessary for that format.
In the LOD world, there is a dual need to support the least common denominator (e.g., CSV exchanged via spreadsheets) and RDF semantics (in Turtle via SPARQL endpoints).

For CSV, the information model classes and attributes would need to be mapped to columns (with unique names). For more formal semantics, an RDF (and/or OWL) Ontology would need to be created.

The most common (and emerging) format to support is JSON-LD for its acceptance by developers and support for linked open data [16]. A JSON-LD context (schema) can also be created from a suitable RDF/OWL Ontology.

6. Data Examples

Current State

The current serialisations of Public Toilet typically show a flat structure and little use of URIs for property values. The example below is from the Australian data set in XML.

```
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="https://toiletmap.gov.au/ file:./ToiletDetails-V_1_1.xsd"
 Status="Verified" Latitude="-37.82620718" Longitude="140.7879912"
 ToiletURL="https://toiletmap.gov.au/toilet/4989" LastUpdateDate="2008-02-04Z">
 <Name>Lady Nelson Information Centre</Name>
 <Address1>Princess Highway</Address1>
 <Town>Mount Gambier</Town>
 <State>South Australia</State>
 <Postcode>5290</Postcode>
 <GeneralDetails>
  <Male>true</Male>
  <Female>true</Female>
  <DumpPoint>false</DumpPoint>
  <FacilityType>Other</FacilityType>
  <ToiletType>Sewerage</ToiletType>
  <AccessLimited>false</AccessLimited>
  <PaymentRequired>false</PaymentRequired>
  <KeyRequired>false</KeyRequired>
  <Parking>false</Parking>
 </GeneralDetails>
 <AccessibilityDetails>
  <AccessibleMale>true</AccessibleMale>
  <AccessibleFemale>true</AccessibleFemale>
  <AccessibleUnisex>false</AccessibleUnisex>
  <MLAK>false</MLAK>
  <ParkingAccessible>false</ParkingAccessible>
  <Ambulant>false</Ambulant>
  <LHTransfer>false</LHTransfer>
  <RHTransfer>false</RHTransfer>
  <AdultChange>false</AdultChange>
 </AccessibilityDetails>
```

Semantic Identity
As is shown, there is an abundant use of properties using boolean values (15 in total) and which all seem mandatory. There is also potential confusion in that it is unclear which of the L/RHTransfer properties applies to which accessible toilet. Most other fields are text literals that do not support any linkages to related concepts/terms. There is only one property (icon) that actually uses a URI, and this is for an image, not for any LOD linkages.

An example from the UK data is shown below which uses the RDF/XML serialisation.

```xml
<dsbase:_4b2v-65nr rdf:about="http://opendata.camden.gov.uk/resource/_4b2v-65nr/32">
  <socrata:rowId>32</socrata:rowId>
  <socrata:member rdf:resource="http://opendata.camden.gov.uk/resource/_4b2v-65nr/"/>
  <ds:name>Kilburn Library Centre</ds:name>
  <ds:opening_hours>Mon-Thu 10am-7pm, Fri 10am-5pm, Sat 11am-5pm, Sun closed.</ds:opening_hours>
  <ds:building_number>12-22</ds:building_number>
  <ds:street>Kilburn High Road</ds:street>
  <ds:postcode>NW6 5UH</ds:postcode>
  <ds:telephone>020 7974 4444 (Contact Camden call centre)</ds:telephone>
  <ds:ward_code>E05000140</ds:ward_code>
  <ds:ward_name>Kilburn</ds:ward_name>
  <ds:easting>525552</ds:easting>
  <ds:northing>183494</ds:northing>
  <ds:latitude>51.536398</ds:latitude>
  <ds:longitude>-0.191174</ds:longitude>
  <ds:location><geo:SpatialThing>
    <usps:deliveryAddress/>
    <usps:cityName/>
    <geo:lat>51.536396</geo:lat>
    <geo:lon>-0.191174</geo:lon>
    <usps:stateAbbr/>
    <usps:zipCode/>
  </geo:SpatialThing>
</ds:location>
</dsbase:_4b2v-65nr>
```
Again, there is little evidence of a real-world model underlying the serialisations, and all data properties are text literals, hindering any LOD opportunities, and requiring complex parsing rules, such as for the opening_hours data property. (The example also includes the foaf:document property which actually only exists in the FOAF ontology as a class, not a property.)

Future State

As an example of what the future state serialisations may look like, the below shows an JSON-LD encoding based on a mapping from the UML Data Model in Figure 3. The goal is to reuse vocabularies (eg vCard, GoodRelations), assign types to resources, and use URLs (eg from DBPedia) as the objects of properties as much as possible.

```
{  
  "@context":
    {  "t": "http://public-toilet.info/",
    "vc": "http://www.w3.org/2006/vcard/ns#",
    "db": "http://dbpedia.org/resource/",
    "gr": "http://purl.org/goodrelations/v1#"
  },
  "@type": "t:ToiletFacility",
  "t:hasName": "Yeronga Parklands",
  "t:hasLocation":
    {  "@type": "vc:Address",
    "vc:street-address": "Brisbane Corso",
    "vc:locality": "Yeronga",
    "vc:postal-code": "QLD",
    "vc:country-name": "AU"
    },
    { "vc:hasGeo": "geo:-27.505678,153.014656"
    }
  },
  "hasAvailability": [  
    {  "@type": "gr:OpeningHoursSpecification",
    "gr:opens": "08:00",
    "gr:closes": "20:00",
    "gr:hasOpeningHoursDayOfWeek": ["gr:Monday", "gr:Tuesday", "gr:Wednesday", "gr:Thursday",
    "gr:Friday", "gr:Saturday", "gr:Sunday"
    ],
    { "t:hasClosedDay": "db:Anzac_Day"
    }
  ],
  "t:hasToilet": [  
    {  "t:hasOccupancy": "5",
    "t:hasGender": "db:Female",
    "t:hasAccessibility":
      {  "@type": "t:Ambulant",
      "t:hasTransferSide": "left"
      }
    },
    { "t:hasOccupancy": "1",
    "t:hasGender": "db:Unisex",
    "t:hasAccessibility":
      {  "@type": "t:Ambulant",
      }  
    }
  ],
```

Semantic Identity
The above JSON-LD serialisation could also be represented as columns in a CSV document (lowering the barrier to entry for LOD) but based on a semantic data model.

7. Conclusion

Developing a robust and comprehensive Information Architecture for an enterprise information system provides the necessary shared understanding for the target solution. The Semantic Web and LOD offer additional advantages (such as sharing, reasoning and reuse) to such solutions. Current approaches have not yet reached the level of maturity expected for enterprise solutions.

There are key design issues for LOD information architectures:

- Create conceptual models of the key entities
- Create formalised information models
- Define properties as first-class entities
- Reuse existing information models and vocabularies
- Discover and use URIs from other datasets
- Plan for linkages to your data

There is no doubt that sharing data via LOD has many beneficial outcomes for both the publisher and consumer. Enterprise-ready information architectures for LOD must approach designing such solutions for open communities with the target state (5-Star LOD) as the goal.

References


Information Architecture For Linked Open Data


Publication Details

Author: Renato Iannella
Email: ri@semanticidentity.com
Date: April 2017
ISBN: 1 74064 804 8
Copyright: © 2017 Semantic Identity. All Rights Reserved.

The information contained in this report represents analysis of publicly available sources believed to be reliable as of the date of this white paper. 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 David Wood (Ephox) for feedback on drafts of this document.

http://semanticidentity.com