Creating and using Linked Open Data

Developments in the 'bigger picture' linked and open data eco-system around international development will mostly emerge from smaller-scale projects by organizations and individuals to create and use linked data. This section takes a more practical turn, summarizing considerations that may be required at different stages of these projects.

We have seen that creating and publishing linked data involves making a range of decisions. A number of good specialist texts exist concerning these decisions, including the openly available Linked Data: Evolving the Web into a Global Data Space (Heath & Bizer 2011) which covers many or the architectural decisions required concerning how data will be hosted and made discoverable on the web, and Semantic Web for the Working Ontologist (Allemang & Hendler 2008) which explores in depth issues concerned with logical modeling of data. This section does not aim to offer a comprehensive technical account of creating linked data, but instead provides an overview of the process and considerations that arise: considerations that frequently require both technical and policy responses.

In the short-term, few systems will natively manage their data and information as linked data. Just as you may publish information on web pages after originally creating it in a word processor, linked data will often be published on the web, converted from where it was originally stored in databases, spreadsheets or other tools. The five star aproach to publishing data online, put forward by Tim Berners-Lee, advocates an approach of publishing data in standard common formats first (for example, Excel and then the more open Comma Separated Values (CSV) format for spreadsheet files), and then following up with publishing data as linked data.

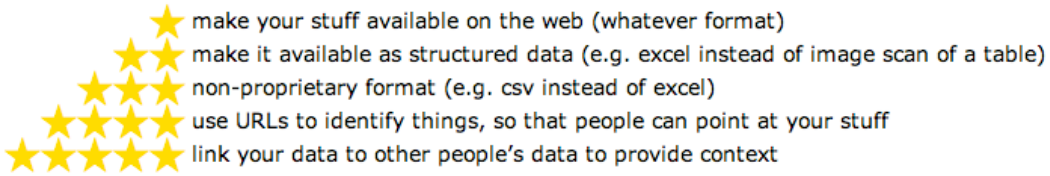


Image Credit: (Summers 2010)

In the section that follows we will look at a number of layers of the linked data stack. Rather than work through it top-to-bottom, our path follows the rough order in which an organization looking to share data as linked data might work through the relevant issues. This highlights the nature of much work with linked data: it's not so much a linear process of stacking up layers – as a job of putting the jigsaw pieces together, trying bit-by-bit to bring the whole picture into view.

Identify what data can be opened and prioritize

The Development Data remix of the Open Data Manual highlights the wide range of data that development organizations might hold[1]. In our pilot projects we've explored publishing data ranging from high-level statistics about countries, down to micro-data survey observations from a longitudinal study. In the process we discovered that, to put the data being published in context, it was also helpful to publish additional datasets, such as the lists of questions used in our surveys, and details of the people involved in collecting data. Linked data advocates commonly encourage organizations to start publishing open linked data by sharing internal taxonomies and code-lists on the web, and mapping these to existing linked data taxonomies. These can provide a 'backbone' for links to be made when other datasets are published. It is also important to think about the particular problems that publishing as linked data may solve: linked data is particularly strong at supporting data integration. For some datasets, simply publishing spreadsheets as open data may suffice in the short term.

In our pilots we had to think about how we could make data available under open licenses[2] without risk to individuals who might be identified in a dataset (we anonymised survey data for example, and for some data chose to only publish summary statistics), and without breaking the terms of other agreements that were already in place. Sometimes data is stored in proprietary systems or managed under terms that restrict its openness. A long-term plan for linked open data might involve adding considerations around openly licensed and easily-to-export data to the procurement or renewal process for database systems in the future[3], or adding data sharing as a requirement in contracts. Sometimes information that could be made available as data will not yet be collected in structured forms and there will be work to be done to identify how to collect data in more structured forms, or extract structured data from various sources of information[4].

---

[1] Available from http://bit.ly/odcc11manual

[2] In fact, in our pilots we never quite got to explicitly specifying the license under which our data was opened, as this turned out to raise a range of organizational issues. Starting a parallel track of work early on to address licensing issues for your data, and looking at examples like the Open Government License (http://www.nationalarchives.gov.uk/doc/open-government-licence/) and Open Database Licenses (http://www.opendatacommons.org/licenses/odbl/) to work out the sort of licensing you might aim for may turn out to be beneficial.

[3] If there is increased interest in linked data, some providers are likely to start offering direct linked data exports from their software. Critical attention needs to be given to whether the ways in which 'default' exports formats from commercial systems will fit with the aims and goals of building a pro-development linked data eco-system. For example, providers of financial systems may have modeled a default export of linked data for it's comparability with domestic spend data, but not modeled it using vocabularies which maximize the possibility of stakeholders in countries where funds are spent accessing, exploring and understanding it.

[4] Again, a lot of considerations need to be taken into account. Frequently grass-roots and frontline staff end up constrained by top-down imposed data

What to publish
It is unlikely that an organization will be able to publish all it's data as open and linked data right away. Choices about what to publish can have a range of motivations:

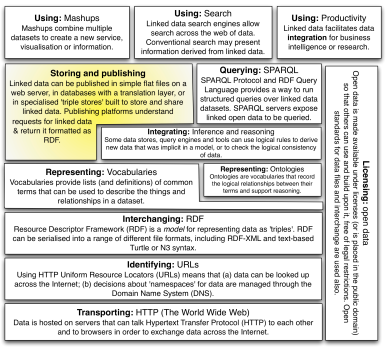
- **Exploring the potential of linked data for the organization;**
- **Being a first mover in order to 'define' concepts or act as a linkage point;**
- **To maximize the re-use of particular datasets;**
- **To contribute to collaborative building of a linked data eco-system;**
- **To integrate data with another organization already publishing this sort of data;**

Prioritizing the data that challenge, rather than contributes to, disparities of power between well-resourced agencies, and the smaller organizations, local communities and individuals that development is intending to support, is one possible guiding principle for development organizations. As this papers companion working paper (ICT for or against development? An introduction to the ongoing case of Web3) outlines, far too often ICTs have been used (both intentionally and many times unintentionally) to enhance the position of already powerful development organizations, rather than ICTs being used as developmental tools that shift the balance of power in favor of the poor and marginalized.

requirements in order to serve the needs of centralized databases – and the desire for structured information can dramatically affect processes, incentives and outcomes in an organization. Both data entry forms, and alternative approaches to generate structure from unstructured data (e.g. Natural Language Processing) will make certain assumptions and have certain normative bias implicit in them. The flexibility of linked open data publishing, the possibility of publishing data locally, modeled for both local and central needs, and mapping datasets together for central use afterwards, may offer a new set of opportunities for designing more democratic data collection systems.

Delivering data: choosing the platform

We have already noted that RDF is a data model, not a file format. RDF data can be made available in a variety of ways.



- **You can publish a file on your web server that contains RDF/XML** (or any other serialization of RDF). You can identify things in that file using the # fragment identifier.

  For example, the 2010 Global Hunger Index is published as the default file at **http://data.ifpri.org.uk/rdf/ghi/2010/qb/** and we use URIs like http://data.ifpri.org/rdf/ghi/2010/qb/#observation-1990-AL to refer to things in the dataset.

  This way of publishing requires no special technology on the web servers where the data is published, but we do need to take care to keep the files in the same place and keep them updated, as other people might start using URIs pointing to them for publishing data.

- **You can encode RDF data in web pages using RDFa.**
  RDFa uses special mark-up added to the templates of a web page to expose structured data and to allow linked data identifiers to be used within a web page. If you have structured information in a website Content Management System that you want to publish as linked data, and you can edit the page templates, then RDFa can offer a route to do this without needing new tools and platforms.

  Note, that the addresses of page on your website then become identifiers for the things that those pages describe – and so keeping the structure of the website stable could be important in future[5]. You can have a page to identify each thing in your dataset, or you can use fragment identifiers to identify multiple things within one page.

- **You can create RDF mappings for existing databases**
  There are tools which allow relational databases to be mapped to RDF models and to serve up RDF data, or custom code could be written which to output RDF from an existing website content management system. There are open source libraries for creating RDF output in many programming languages.

- **You can store RDF data in a triple store**
  A triple store is a form of database optimized for storing RDF linked

[5] The UK Governments guide to 'Cool URIs' for the public sector (REF) recommends that when creating important sets of URIs (identifiers) that creators should be thinking about keeping them stable for at least 10 years – as others may come to rely on the identifiers you provide. This is particularly relevant for 'infrastructural' data like definitions of countries or categories of projects.

data.

Many triple stores, and some tools for mapping relational databases to RDF also provide SPARQL interfaces for users to not only fetch linked data, but to also be able to query all the linked data an organization holds. Some triple stores provide tools for inferencing over data – logically processing it to identify implicit facts (for example, if the data states that a project took place in a province of Uganda, we can also assert the implicit fact that it took place in Uganda – and inferencing engine could draw out this fact with the right vocabularies and ontologies to draw upon.)

When using a triple store or mapping an existing database to RDF it's important to consider how it will respond to requests for data to URIs used in the dataset (when a computer looks up an idenfitier, or a human follows links to an identifier in their web browser). Some sort of front-end is often needed to connect the triple store and expose it at relevant URLs on the web. Usually you would have one URI for each thing in the dataset. The Cool URIs guidance outlines recommend approaches to choosing URIs. (REF)

Content negotiation is an approach whereby humans looking up a URL can be directed to a human-readable web page about the thing identified by that URL, whereas machines requesting RDF can be redirected to an RDF data file. DBPedia.org uses content negotiation. Human visitors to http://dbpedia.org/resource/Oxford are sent to http://dbpedia.org/page/Oxford whereas machines wanting data are redirected to: http://dbpedia.org/data/Oxford

A number of linked data providers have also recognized that, for many people interested in re-using data, RDF/XML or N3 are unfamiliar formats, and so some, such as data.gov.uk, are using a linked data API to return JSON, XML and CSV data on request also (based on content negotiation and file-type endings on the URL)

The right approach to use for publishing data may vary from dataset to dataset. Publishing data in flat files is probably the easiest way to get started with linked data, and it's possible to make linked data files using desktop software such as Google Refine with the RDF Extension[6]. A triple store can provide a way for information from across an organization to be integrated together. For example, if different departments make statements about a identified thing (e.g. a project) and store that data in the same triple store, it becomes possible to build up a more detailed picture of the thing – without needing to know about separate data files where information is held. Triple stores can be used to store linked data that is not open for integration inside the organization, as well as linked open data.

[6] http://lab.linkeddata.deri.ie/2010/grefine-rdf-extension/

Policy issues: new responsibilities?

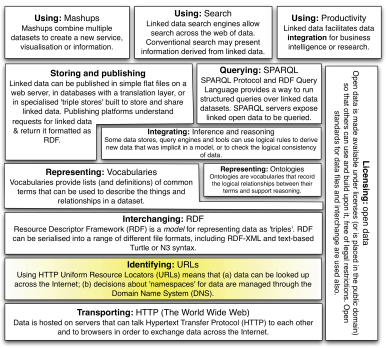
Whilst many organizations are used to establishing and maintain an Internet presence, linked data introduces a range of new responsibilities for stewardship of data resources. Whilst we're familiar with arriving at a website that has recently been redesigned, and where links are broken, requiring the user to search for the page they were looking for – publishing linked data responsibly may involve a commitment to keeping links available, or using web standards such as HTTP redirect codes, to ensure that links keep working even when content moves.

There is undoubtedly, at least at the present time, a higher technical barrier to entry to effectively publish linked data, and to get the maximum benefits from it through hosting it in a triple store or other tool that allows data to be queried. Development organizations will need to consider what responsibility they have to provide support or platforms that ensure beneficiaries of projects can participate as creators of content in the linked data eco-system.

Publishers toolbox

- Jena, Virtuoso Community Edition and Garlik 4Store are examples of open source triple stores;
- Talis Open Data Commons provide a free hosted triple store service for open licensed linked data;
- The Puelia Linked Data API can provide a front-end onto a RDF triple store, providing human readable, CSV, JSON and XML exports of data.
- Version 7 of the Drupal Content Management system uses RDFa to expose linked data within pages. Add on modules can be used to extend this capacity to provide advanced linked data modeling within the Content Management System.

Choosing identifiers



We have seen that the linked data model makes use of web addresses as identifiers for things. This is because a web address can be a 'global identifier'. Each web address is made up of parts:

- A **protocol** (generally http:// for linked data and web pages)
  which tells computers *how* to fetch data.

- A **domain name** (e.g. example.com) which computers look up in the global Domain Name System (DNS) registry to work out who the domain belongs to, and *where* the servers are to be found which they should fetch data from.

- A **filename** (e.g. /resource/Oxford) which is what computers will request from the server they have located using it's domain name.

Linked data uses the governance structures around the Domain Name System to handle questions of assigning control of identifiers to organizations or individuals. If you own a domain name you can control what data is returned when someone looks up (dereferences) identifiers pointing to things in that domain. In linked data, each domain name (and each subdirectory or file) can provide it's own 'namespace' where identifiers are created. This can be contrasted with global identifiers like ISBN numbers on books, where there is one 'namespace' and one global registry of books, with it's own more formal governance structures.

What URIs can I use?

You can use any URIs you like to identify things in your dataset. They don't have to be URIs that you control the domain name or server for, nor do they have to be URIs which return linked data when looked up. A dataset can contain statements about these URIs locally. However, you can only choose what data is returned when a URI is dereferenced if you control that URI.

For example, I can create an RDF file and put it on the IKM Emergent servers which states:

> @prefix ikm: <http://ikmemergent.net/def/>.
>
> Ikm:infomediariesProject ikm:workedWith <http://younglives.org.uk/id>.
> <http://younglives.org.uk/id> rdfs:label "Young Lives Project".

If you look up http://younglives.org.uk/id you won't find any RDF data, but anyone reading my file will know that the thing I'm referring to using the identifier <http://younglives.org.uk> can be given the label "Young Lives Project". Even though I don't 'own' the younglives.org.uk domain I can still make statements about this 'thing' in my files, and anyone accessing my linked data will be able to choose to pay attention to these statements or not. However, if the Young Lives Project did choose to start publishing RDF data

in future, then they could make extra statements at http://younglives.org.uk/id that would expand what any linked data applications which dereferences this URI could discover.

Two extensions of the RDF model commonly supported by tools for accessing and querying linked data, RDFS (REF) and OWL (REF), provide some special properties that are useful when choosing what URI to use to identify things. The rdfs:seeAlso predicate allows you to use a URI in your own domain, but tell human or computer agents accessing it that more information about that resource is available at some other URI. The owl:sameAs predicate allows you to state that systems should consider your URI for a thing as equivalent to someone else's URI for a thing – and any facts they find linked to your URI, or to the sameAs URI can be asserted to be about either. Allemang & Hendler (2008) explain the consequence of using rdfs and owl in more detail. For our purposes it is suffice to say that: you can refer to a thing using a third party URI, or you can create a URI in your own domain and control (allowing you greater capacity to annotate the thing identified) but assert that it is the equivalent of a third-party URI.

Cool URIs and language

The Cool URI guidance at REF provides some suggested good practice for deciding on what URIs to create in your own dataset. It includes suggestions such as using URIs that are easy to guess, and having a hierarchical structure to URIs.

One issue that can arise when creating multi-lingual linked data is which language to use for URIs. For example, if you are providing a list of countries, should you use: the English name as part of the URI (easier to guess for English speakers); an ISO country code (standard across languages, easy to guess for some computers); or should you create a URI in each language and assert they are 'sameAs' some additional URI with a country code, and/or with each other. This third option may seem to be the most appealing in terms of promoting linguistic equality, by it raises some practical difficulties – as someone using these URIs would have to dereference them to discover the equivalence – and there is a risk it could lead to a duplication and fragmentation in data.

External URIs

A lot of the benefits of linked data come when you identify things in your dataset using third-party URIs.

For example, instead of using your own identifiers for a country, when you link against the Food and Agriculture Organisation's (FAO) geographical ontology (http://www.fao.org/countryprofiles/geoinfo/geopolitical/resource/) you find that: (a) you are using an identifier that many other people may be using in their datasets, and so it will be easier to identify where you hold data about the same things; (b) when you look up (dereference) an identifier in the FAO ontology, you will find they provide detailed additional information about countries, including their 'codes' in other code schemes such as ISO codes, or their identifiers in key linked data hubs like dbpedia.org. Your application,

or applications querying your data, can now choose to integrate all this information.

However, in this example, a number of additional consequences can flow from the decision of which URIs to use:

Firstly, you establish which third-party datasets it will be easiest to integrate your dataset with.If you use exactly the same identifier as a third-party dataset, then you can mix your datasets together in a triple store or RDF aware tool and instantly have integrated data. If you link against a source such as the FAO geopolitical ontology which provides useful mapping information (e.g. ISO codes), then you give your applications access to the information they need to integrate with a dataset that uses such codes, but, the integration is likely to require some addition work, either in how queries over the data are constructed, or in using reasoning tools which look for implicit connections in the dataset and add them to a triple store.
Given this additional effort may require time, skills, software or equipment in some cases, choices of identifiers may impact significantly on how data gets used, who it is used by, and who the burdens of integration effort fall upon.

Sometimes there may be two or more possible sets of identifiers to use for a thing, with some datasets using one set, and others using the other, and no existing mapping between them. In these cases, if the mapping between terms is non-trivial, your choice of identifiers can place you within a particular community of datasets that can only be connected when an investment is made in mapping to integrate the two.

Secondly, you may influence other's use of URIs, setting informal standards through your data publishing. There are strong network effects when it comes to choice of URIs. If you are publishing a significant dataset and choose to use a particular set of URIs, others who come along to publish after you may follow your choice. Linked data doesn't have formal standard setting processes, so precedents function as informal standard setting.

Thirdly, you decide who you are delegating authority over particular concepts to. This delegation of control can happen on two levels:

(1) Often URIs will follow established standards devised by offline systems. For example, FAO's country ontology of URIs only contains countries that FAO, as a UN body, has chosen to recognize. If you want to refer to a country that FAO doesn't recognize, you won't find an FAO URI. The choice of URI can involve a commitment to following a particular institutions view of the world;

(2) You delegate control, to some extent, over defining the thing referred to the owner of the domain of the URI. For example, FAO could choose to start making new assertions about countries in their dataset which did not fit with your understanding of a country. Or another third-party you were linking to could completely change, or cease to provide, the URIs you were using.

Neither of these issues are insurmountable. You can create your own URIs for concepts that a third-party does not have coverage for; and you can choose not to trust particular third-party data in your applications, or to update your dataset to use alternative URIs in future if a third-party ceases to provide useful data. However, if most of the entities in your dataset are linked to third party URIs, but a small proportion are not, there is a risk these could become 'second class citizens' in your data or could be missed out in queries which assume everything is linked to the third-party URIs.

One phrase that came out of the IKM Linked Data workshop to capture the decisions involved in choosing URIs was that, to gain full benefit from linked data, we must face the "Economics of integration" or the "Politics of delegation" – pointing to the need to either spend time and effort creating your own URIs and mapping these to diverse other URIs sets (as the FAO Ontology does), or to delegate control to third-parties, making explicit or implicit choices about which concepts can be easily used in a dataset, and how those concepts are defined[7].

Policy issues: providing identifiers

What identifiers does your organization need to provide? Are you responsible for managing particular taxonomies, vocabularies or thesauri used by others? Are there particular project identifiers that you hold the authoritative data source for?

Coming up with a plan to make your identifiers available as linked data on the web, and to map these to other identifiers can be a good starting point for publishing linked data. The priorities you set for mapping together different identifier sets will affect the sorts of ways your data (and related data) can be queried and used.

When you decide to re-use a set of identifiers that someone else has provided (linking against external URIs) think carefully about the commitments you are making: will you limit your capacity to describe the world as you understand it in your data by delegating control of defining certain concepts to a third party, or do you gain extra connections by linking against third-party terms?

It is likely that many of the identifiers you might create will be based on 'top down' definitions of things. Thinking about how you also provide space for bottom-up definition of terms, and creation of identifiers will be important too. Whilst Allemang & Hendler (2008) argue that in a web of data in theory Anyone can say Anything about Anything (they describe this as the AAA rule),

[7] You won't, for example, find Kosovo in the FAO ontology (though you do find a term for Socialist_Federal_Republic_of_Yugoslavia_the). Institutional policies, in this case UN policies, feed into the politics of the URI set. And the politics of the URI set could, without attention being paid, impact upon the relative visibility of knowledge from different communities within a web of data.

the ability to effectively create re-usable identifiers, rather than re-using identifiers from others, requires control of a domain name or a stable places on the web to publish information.

Publishers toolbox: finding identifiers

There are a number of places to look to find possible URI sets to draw upon:

- **DBPedia.org** contains identifiers for concepts that have a page on Wikipedia in a number of languages. It often provides useful data, and has the same advantages and limitations of Wikipedia as a user-generated resource[8].

- **SameAs.org** allows you to enter an existing URI (for example, a URI from DBPedia) and to find alternative suggestions.

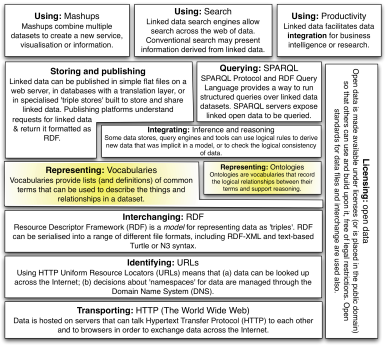
- **Sindice.com** is a linked data search engine. Search for terms and then look to see if they suggest the existence of a URI set.

- **The Linked Data Cloud diagram** (http://lod-cloud.net) provides a clickable map of large linked data sources. Exploring these, the identifiers they provide and use can prove useful.

You can look up different identifiers and explore the data they return (and thus what commitments they make about the world) to inform your choice of the right identifiers to use.

[8] For example, coverage and quality varies significantly.

Modeling data



In a spreadsheet, modeling your data involved deciding which columns you will have, and what will go in each column (for example, do you put address all in one column, or have separate columns for each of 'Street name', 'Town' and 'Postcode'/'Zipcode'). In a relational database you also need to think about how you will spread information across different tables, and how to set up relationships between tables (for example, you could have a controlled list of Countries, and link address records to these using a foreign key rather than including a free-text field for country).

Modeling choices often involve trade-offs between a number of factors:

- **How easy it is to enter the data/convert existing data into the proposed format** – e.g., are all the addresses you have formatted so they can easily be split up at the commas into separate parts.

- **How much flexibility you want in the data in future** – e.g. right now you might only need addresses for mail-merge, but in future you might want to sort your data by town, or to use postcodes to map the data.

- **How easy or complicated are queries against the data will be** – e.g. in a spreadsheet, do you have to pull values out of different columns for a mail-merge or to generate reports; or in a relational database if data is normalized with towns and countries in their own tables, how many join statements do you need to query the data, and how much does a user need to know about the structure of the data to write a query against it.

- **Cost** – on big, complex or heavily used datasets complicated queries can be expensive in terms of the amount of computer power they require.

Modeling linked data involves many of the same processes as modeling tabular or relational data: identifying how to split up fields, and how to articulate relationships between them. However, it also involves making choices about how your data model will link into a wider web of data. As the middle term in a RDF triple (the predicate: you can think of it as equivalent to a field names, column heading or property if you are more familiar with these concepts) is an identifier just like any of the other identifiers in your dataset, it can also be a link to a third party term. Or you can introduce your own terms, but relate them to third party terms and classes of thing using a range of relationships (rdfs:subClassOf; rdfs:subPropertyOf; and rdf:type to name a few). The full details of RDF data modeling are beyond this short paper. In the following section we will focus on considerations around choosing existing models, but those looking to move beyond this are encouraged to take the time to explore an RDF modeling text in more depth.

An RDF vocabulary is a collection of terms, each with it's own URI, that you can use to model your data. Ontologies are specialized vocabularies that use RDF standards and extensions like RDFS and OWL to record the relationship between the terms in the ontology, and to allow logical inferences about the data represented using them to be drawn. We will use the term 'vocabulary' to refer to both vocabulary and ontology. Just as you can choose to create your own identifiers for things, or you can choose to mix-and-match identifiers from other sources, you can 'model' your data using terms from one or more existing vocabularies, or you can invent your own vocabularies, or even extend an existing one[9]. When you re-use a widely adopted third-party vocabulary party you increase the chance (a) of it being easier to combine your data with other data using that uses the same vocabulary, and (b) that there will be applications that already exist that know how to display or work with your data. However, just like identifiers, vocabularies can be mapped together – allowing some flexibility in the use of local conventions for representing data, but, through using 'reasoning' software and mapping between vocabularies, allowing data to remain compatible with other datasets outside of the organization.

Tim Berners-Lee, inventor of the World Wide Web, and advocate for linked data, is reported to have originally considered the title 'Philosophical engineering' for the discipline of understanding the web and linked data web now called 'web science'. The term 'philosophical engineering' may in many ways have been more appropriate, as data modeling for linked data can quickly start encountering complex philosophical questions about identity, time and what properties things have. Because the linked data model theoretically makes it possible to have a constant regress of definitions, and for additional content and annotations to be attached to most things, finding the boundaries of what you model can be difficult. Often a 'best-efforts' approach, pragmatically modeling as much as you can, but focusing attention on areas where the most value is to be gained from detailed modeling, will need to be adopted.

A note on language

The current RDF standard uses language tags to allow different language versions of literal text to be flagged up. For example:

> <http://dbpedia.org/resource/South_Africa/> rdfs:title "Republic of South Africa"@en
>
> <http://dbpedia.org/resource/South_Africa/> rdfs:title "Republiek van Suid-Afrika"@af

can both be asserted in a dataset, leaving it up to the application using the data to select between the @en and @af tagged language representations.

[9] See http://richard.cyganiak.de/blog/2011/02/top-100-most-popular-rdf-namespace-prefixes/ for a list of popular vocabularies (sometimes referred to as namespaces) based on queries to the prefix.cc service.

However, the SPARQL query language does not currently support searching by language tags – so, as the Lingvoj.org site, providing language identifiers as linked data resources, points out, it's not possible to make use of language tags to query linked data to ask questions like:

- "Can I find native speakers of Bengali in Berlin?"
- "Which books by Victor Hugo are translated in Arabic?"
- "Is this software documented in Chinese?"

Where retrieval of content by language is important, then attention to modeling this will be required.

Policy issues: choosing and creating vocabularies and models

Modeling decisions are decisions about what sort of data you are aligning your own data with.

Modeling decisions may need to be reviewed as the web of data evolves. New conventions may be established, and you will need to consider whether you should update your data model to fit in with them.

You may need to generate a new vocabulary (or set of conventions about which vocabularies to use) for your data publishing. If there are other people in need of the same sort of vocabulary, consider how you can establish a collaborative process of vocabulary development. Vocabularies often have a community of users around them. Some widely adopted vocabularies like Dublin Core (DC) meta-data standards have formal governance structures for deciding what terms they will include and articulating publishing best practices. Others have more informal 'open source' and lightweight collaborative structures based around mailing lists. Face-to-face meetings often play an important part in establishing the outline of a new vocabulary, or refining a draft vocabulary. If you are involved in generating new vocabularies, consider how to make sure the process is inclusive of a wide range of stakeholders. (Allow time for modeling decisions; involve users as well as technologists; find participative processes for making modeling decisions).

Publishers toolbox: finding vocabularies

How do you find vocabularies and ontologies to use to publish your data? Again the process often involves looking at what others are doing. Sometimes if no one is yet publishing data similar to yours as linked data you will need to decide whether to create a whole new vocabulary, mix or match from others, or modify one or more existing vocabularies.

- **Schemapedia.com** – includes over 250 vocabularies (or schema) covering a range of topics.

- **Swoogle** (http://swoogle.umbc.edu/) indexes and searches a large number of ontologies and can be useful to find example files with – although many vocabularies you will find are only used in limited

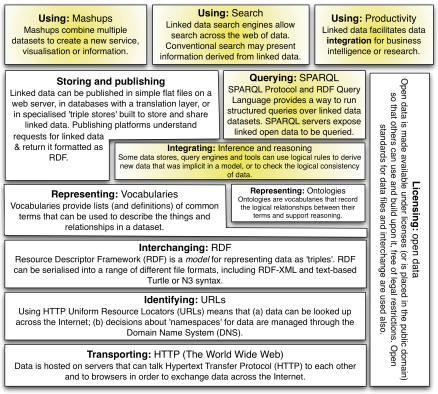
contexts, so check a vocabulary has a community of users around it before selecting.

- **VocabularyMarket** (http://www.w3.org/wiki/VocabularyMarket) is a page maintained by the World Wide Web Consortium listing common vocabularies and search services.

Sometimes there will be a clear data model to use, but an RDF vocabulary for it may not yet be defined (for example, the IATI Standard defines how to publish Aid flow information as XML, but an RDF vocabulary for sharing IATI data has not yet been agreed). Developing a temporary model based on a non-RDF standard can be an effective way to get started with modeling data where a linked data example to copy is not yet available.

Using Linked Data

Whether you've gone through the process of chosing identifiers, finding models and publishing your own linked data, or you have found linked data resources you want to draw upon – at some point the question has to arise: how do we actually work with this linked data?



This section looks briefly at some of the uses that can be made of linked data, and the tools required to enable that use.

Inference

We have mentioned 'reasoning' a number of times before. Because of the logical model underlying RDF it is possible for tools to infer implicit data held in a linked data file, or that can be worked out when a number of linked data files are combined.

Some triple stores and code-libraries for working with RDF support inference, but it is usually an option that needs to be explicitly chosen when writing queries or setting up code. When configured correctly, inferene-aware tools will allow you to access, query and work-with implicit triples as well as those you explicitly recorded in their input files.

Inference can be used to combine disparate knowledge bases, or to help map together datasets. Many of the benefits of linked data for data integration will involve some use of inference.

Querying - SPARQL

SPARQL is the primary query language for working with linked data. It has some similarities to the SQL Structure Query Language that is used to explore relational databases. Using SPARQL queries you can find out what is contained within a triple store, and you can write queries that retrieve a table of data (SELECT queries) or that retrieve new RDF graph structures (CONSTRUCT queries). DESRIBE queries can be used to find out about an entity contained within a triple store.

Just as to write an SQL query you need to know the structure of the tables in a relational database, to write a SPARQL query you need to know something about the structure of the RDF data you are querying. However, because of the self-describing nature of RDF data it should be possible to uncover the structure of data through running a number of queries, and where data is well annotated, to discover detailed descriptions of what is contained within an RDF store of linked data.

Most implementations of SPARQL only query the data held within a specific triple store, so to query across multiple datasets you either need to combine them in the same data store, or to use a query engine which will follow links to external URIs and pull that data temporarily or permanently into it's data store.

There is no way to run SPARQL queries against the whole web of data at

once, only against those parts of it you have copied into local data stores.

Linked data client tools

Some linked data tools will help you take advantage of the web of data by following links to other data sources on demand and making third-party data accessible to your application.

For example, the Graphite PHP Library[10] provides a simple way to fetch additional data about an external URI on demand and to display it on a web page.

When you integrate linked data from other sources at 'run-time' you will need to think about how to either ensure you only integrate trusted data, or you make clear which data is from you, and which is third-party data.

Mash-ups

A mash-up combines data from different sources to create new insights – usually via visual presentation of data.

Linked data can be a powerful enabler of mash-ups. For example, in the Young Lives pilot project we had statistics on a number of countries, and we were using Geonames.org identifiers to refer to countries. Using a SPARQL query engine (Virtuoso) which was able to fetch external data at run-time, without having recording co-ordinates in our own dataset we were able to create queries which fetched geographic co-ordinates for each country and put our statistics on a map.

Identifying the mash-ups that may be possible with your data involves exploring what data third-parties you are linking against provide, and identifying ways to either write queries that will draw on this, or developing tools that fetch this data into your applications.

Search

Linked data can be used to improve search applications. Resources like dbpedia.org and other lists of terms may provide translations or related terms that can be used to answer search questions. The FAO have done extenstive work to explore the use of vocabularies and ontologies to support multilingual search. Most search tools will need to be specifically customized to take advantage of linked data.

Your linked data may also be used by search engines, both specialist linked data search tools like Sindice.com, and mainstream search engines like Google.com (which currently focus on using microdata and RDFa).

Data integration

Policy issues: using linked data

[10] http://graphite.ecs.soton.ac.uk/

Whilst this paper has predominantly focused on creating linked data, it is important to think about where you can take advantage of linked data as a consumer.

Applications can be configured to follow links and to seek out third-party data, but generally you still need to actively discover and choose third-party which third-party data you should draw upon.

Whilst linked data can make integrating different datasets easier, it does not solve all the problems – tools and support are still needed to making effective use of it.

Developers toolkit

Consumer tools for working with linked data are still in their early stages, and many open source tools were generated during short-term research projects and so many not be actively maintained. However, there are a wide range of tools, and most programming languages have some sorts of open source libraries for working with linked data. Below are just a few possible tools:

- **Graphite** (http://graphite.ecs.soton.ac.uk/) is a simple PHP library for working with linked data.

- **The SPARQL Proxy** (http://data-gov.tw.rpi.edu/ws/sparqlproxy.php) provided by the Tetherless World Constellation provides a useful tool for fetching back CSV and other file formats from SPARQL queries if your own SPARQL tools do not support this.

Going further

There is much more to be said about creating and using linked data. The tools for working with linked data are developing rapidly, as is the community of users. Through this section we hope readers will have been able to see in more detail some of the judgments involved in creating linked data, and to discover useful resources for projects starting their exploration of linked open data. This second of the report will be made available as part of the Open Data for Development manual being created by Open For Change, and will, we hope, be able to be updated as the field develops.