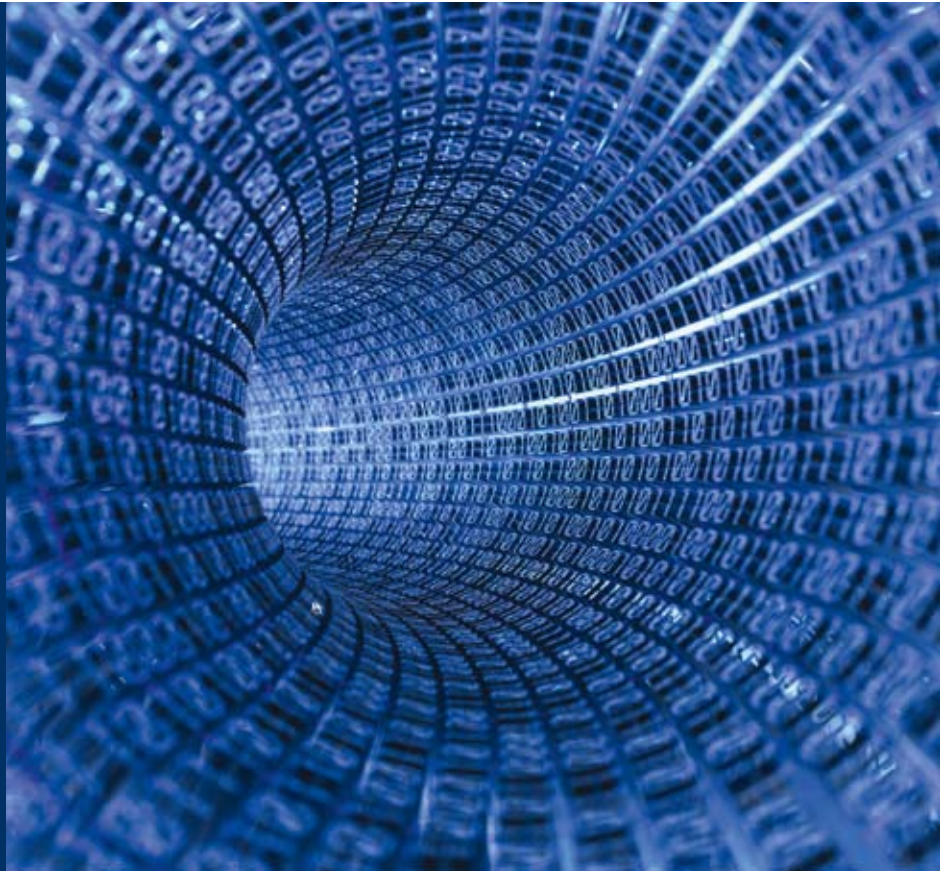


## Why Linked Data for data.gov.uk?

Jeni Tennison looks at how linked data will help to open up government data and identifies the challenges ahead.



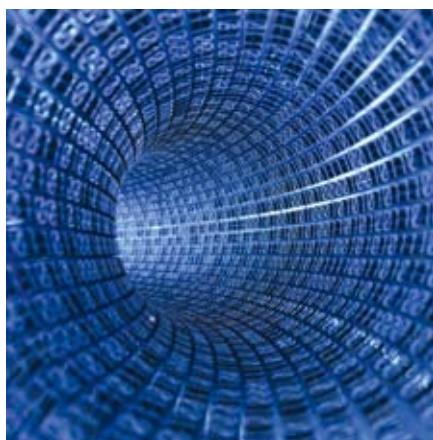
February 2010

[www.tso.co.uk](http://www.tso.co.uk)



Jeni Tennison is an independent consultant currently contracted to TSO. She specialises in XML-related and semantic web technologies with forays into client-side development. She trained as a knowledge engineer, gaining a PhD in collaborative ontology development. Since becoming a consultant she has worked in a wide variety of areas, including journal publishing, medieval manuscripts, legislation and financial services. She is author of several books including “Beginning XSLT 2.0” (Apress, 2005). Jeni has worked with TSO and OPSI on applying RDFa to the London Gazette to open up the data for reuse and on APIs for publishing legislation. She is currently advising on the linked data aspects of data.gov.uk.

[www.jenitennison.com](http://www.jenitennison.com)



## Why Linked Data for data.gov.uk?

**data.gov.uk** was finally launched to the public in January (still in beta, but now a more public beta than the beta that it's been in for the last few months). It's a great step forward, and everyone involved should be proud of both the amount of data that's been made available and the website itself, which was developed rapidly by a small team based on open source software (and at low cost).

*This is a first step on a long road.*

One of the features of the UK Government's approach to freeing data is the emphasis on using **linked data**. What I don't think has really been articulated is either what that means or why we should take this approach. From what I've seen, developers seem to think:

- Linked data is a synonym for turning everything into RDF and putting it in one big triplestore, equivalent to making one big database of government data and therefore prone to exactly the same, well-known and understood problems that government has with creating huge databases
- Linked data requires everyone to agree to the same model and vocabulary, which means huge efforts in standardisation and ends up with something that suits no one
- The UK government will be releasing all their data as linked data immediately, and in no other way
- The UK government has been seduced into using linked data by academics who don't understand anything about how the web or the real world works
- The UK government has been seduced into using linked data by big businesses who stand to make a pretty penny providing services to departments that are forced to publish their data in this way.

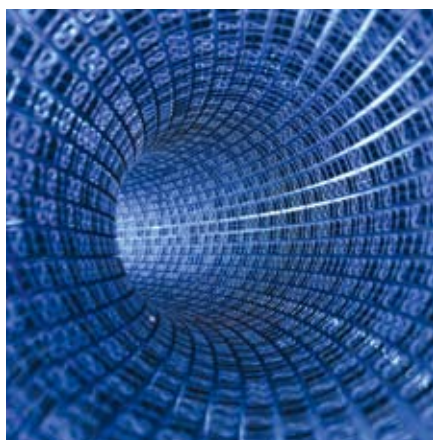
None of these are true. In fact, the UK government is committed to publishing data as linked data because they are convinced it is the **best approach available for publishing data in a hugely diverse and distributed environment, in a gradual and sustainable way.**

*Why?*

Because linked data is just a term for how to publish data on the web while working *with* the web. And the web is the best architecture we know for publishing information in a hugely diverse and distributed environment, in a gradual and sustainable way.

If you're a web developer, you already know that the best APIs are **RESTful APIs**. That argument has been won. It means:

- Using (HTTP) URIs to identify resources: naming *things* with URIs rather than actions on those things (which are carried out using the standard set of HTTP verbs)
- Recognising the distinction between resources and representations of those resources: the same URI might return a different representation of the resource, such as HTML or XML or JSON
- Returning self-descriptive messages: being able to process representations in a manner that is obvious from the mime type
- Hypermedia as the engine of application state: being able to locate additional resources through the use of (typed) links.



Linked data is about following these rules for publishing data. It is about using URIs to identify things, providing information at the end of those URIs that is self-descriptive, and linking those things to other things through typed links.

One of the features of this approach is that it doesn't require any big bangs. No one planned the web: sat down and mapped out each page and its precise relations to every other page in advance. It grew, and evolved, and continues to grow and evolve every day. It grows through individuals and institutions publishing information for their own reasons and linking to other people who have published information for their own reasons, and because we have some fundamental standards that clients and servers understand, it All Just Works.

### RDF and SPARQL are crucial standards

Did you notice how I slipped in the “because we have some fundamental standards that clients and servers understand”? One standard is obviously HTTP: that controls how clients and servers can talk to each other: it allows clients to request pages and servers to respond. Another standard is HTML: that enables browsers to display information in ways that people can understand it, and (crucially) has a known set of semantics that browsers can use to tell when something is a link, which people can navigate to find more information.

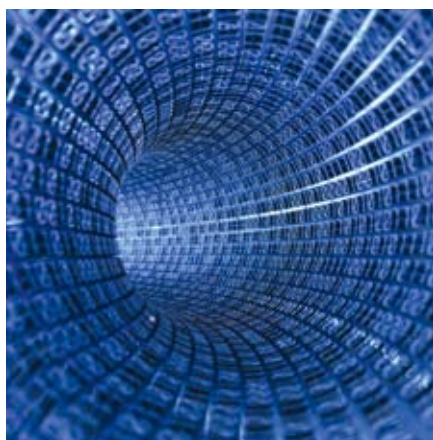
For linked data, there are two crucial standards: RDF and SPARQL. Yes, I know what you're thinking, because believe me two years ago that would have been my reaction too, but let me explain why.

There's one way in which publishing data isn't like publishing documents: its model. Documents are made up of paragraphs and headings and lists and tables and so on. Data is made up of... what? Well, at its most basic, it's *things* that have *properties* which have *values*. We might call the things *objects* or *entities*, and call some of the properties *relations*. We might even call them *records* with *columns* and *values* and *foreign keys*. But however you term them, for better or worse, we do tend to think about data in this way: *thing, property, value*.

So if we are going to publish data on the web, we need a standard way of expressing the data so that a client receiving the data can work out what's a *thing*, what's a *property*, what's a *value*. **And, because this is the web, what's a link.** This is the fundamental standard we need, and this is what RDF gives.

RDF is actually a model rather than a syntax. It's a bit like the split between the DOM and HTML or XHTML. The DOM tells the browser how to render the page: the HTML or XHTML is just a syntax which the browser is able to convert into a DOM that it displays. We could imagine browsers converting wiki syntax into a DOM. Or creating a DOM based on XML and XSLT, which of course they all do.

So, RDF is like the DOM, with varying representations of RDF (XML-based, text-based, JSON-based, even HTML-based) that can be used to pass to the client the underlying model of *things* and *properties* and *values* (some of which are *links*). What the client does then is its business: clients that retrieve data aren't browsers — they're not all going to display the data, use the same parts of the data, or otherwise process it in the same way — but they can pull out the *things*, *properties* and *values*, and know which are *links*, and this data structure will often, with a good RDF library, map on to some natural structure within whatever programming language is being used, and make the programmer's job easier.



## Vocabularies will evolve

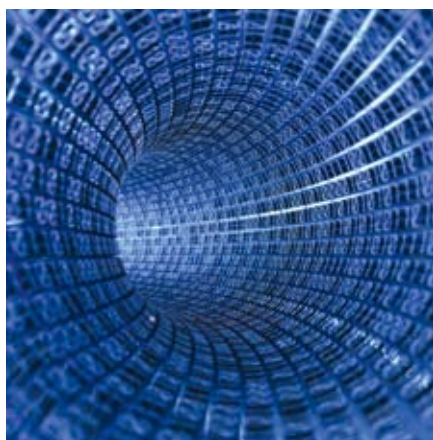
What we don't want to have to define are standard ways of expressing *particular* data (such as data about a school) because different individuals and organisations will have completely different ways of thinking about a particular thing. A school itself will have information about uniform and open days; **OFSTED** about performance; **Edubase** about administration and pupil numbers; the PTA about after-school activities. Expecting everyone to adopt a particular standard vocabulary for describing a school is as futile as expecting everyone to adopt exactly the same page layout within their web pages, and exactly the same class names in their CSS.

But we don't want to rule out opportunistic alignments where individuals or organisations, for whatever reason, *do* want to use the same vocabularies. Look at what's happened with classes in HTML. There is absolutely no constraint on what classes people use in their HTML. But there are clusters of web pages that use some of the same classes. Websites that use **microformats**. Websites that adopt a particular **CSS framework**. Importantly, though, even where some classes are shared, it doesn't mean that *all* classes are shared: adoption of a particular microformat or CSS framework doesn't limit the rest of the page.

RDF has this balance between allowing individuals and organisations complete freedom in how they describe their information and the opportunity to share and reuse parts of vocabularies in a mix-and-match way. This is so important in a government context because (with all due respect to civil servants) we *really* want to avoid a situation where we have to get lots of civil servants from multiple agencies into the same room to come up with the single government-approved way of describing a school. We can all imagine how long that would take.

The other thing about RDF that really helps here is that it's easy to align vocabularies if you want to, post-hoc. **RDFS** and **OWL** define properties that you can use to assert that this property is really the same as that property, or that anything with a value for this property has the same value for that other property. This lowers the risk for organisations who are starting to publish using RDF, because it means that if a new vocabulary comes along they can opportunistically match their existing vocabulary with the new one. It enables organisations to tweak existing vocabularies to suit their purposes, by creating specialised versions of established properties.

So the linked data web is designed to grow and evolve in exactly the same way as the human web has grown and evolved. It grows through people adding links to existing data. It grows through people creating their own vocabularies. And it evolves as links break and reform, and vocabularies combine and diverge. It is complex and messy and self-organising.



## The importance of URIs

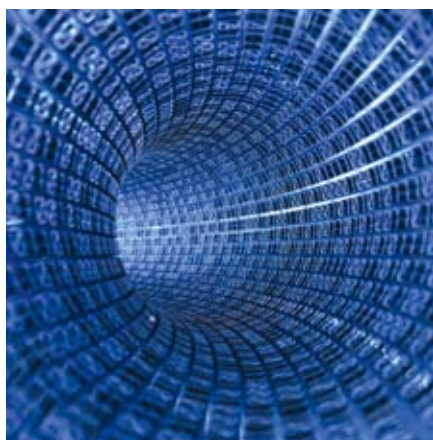
The cornerstone of the great, messy, web is the URI. URIs have two important roles:

- **They identify things** – If two sets of data use the same URI then it's dead easy to work out when they are talking about the same thing, for example to bring together the information published by a school with its OFSTED report with its pupil census. Spread this around to five, 10, 20 datasets from different places all using the same identifier for the school, and you have huge pool of information. And the great thing about RDF (because they also use URIs to identify properties) is that those datasets can be combined automatically without worrying about clashes, rather than through painstaking developer effort.
- **They provide somewhere to look for information** – This is the point of using HTTP URIs, because that look-up is as simple as retrieving a document from the web. This enables programmatic, on-demand, access to the information. Developers don't have to download huge database dumps when all they are interested in is a small fraction of that data.

But we know that of course sometimes developers *do* want to download huge database dumps. So we need URIs for those dumps, and ways to associate metadata with them, and ways to search them. Adopting linked data doesn't preclude providing sets of data in larger lumps. In fact, what's needed are ways of creating those larger datasets by bringing together the more granular linked data into lists and graphs; this is essentially what SPARQL does.

We also know that there's a trade-off to be made between the power of URIs and the simplicity of using short, unqualified names, particularly when it comes to naming schema-level entities such as properties or classes. Most mashups that we see at the moment bring together just a few datasets, making it easy for developers to scan for naming clashes, or examine values to work out whether a particular property contains a link or not. This is the 80% of the use of data on the web that can be addressed by the 20% solution of the kind of JSON and plain old XML you see in most APIs.

But publishing with RDF can be the basis of these kinds of simple APIs, and still address the hard 20% that we will encounter quickly as we mash more data together. Anyone working with data knows that the main challenge of making data available in an easily accessible way is cleaning, tidying, modelling and restructuring. If that's done into RDF then creating simple JSON, XML and even CSV is really easy. Creating middle-ware that will make the creation of these basic APIs really easy must be the top priority of this linked data effort.



## Reality Check – there are still challenges ahead

Just as in the early days of the human web, we face huge challenges simply getting tooling to a level where it's easy (really easy) for government departments and local authorities to publish data as RDF and for the consumers of the data to use it. We have some patterns for publishing linked data, but, as in the early days of the human web, there's still a lot we don't know about the best way to make data usable by third parties.

It's worth noting that the main challenges we face are ones that are common to all attempts to make data both open and reusable. How do we easily create structured and reusable data from presentation-oriented Excel or (worse) PDFs? How do we handle changes over time, and record the provenance of the information that we provide? How do we represent statistical hypercubes? Or location information? These are things that we will only learn by trying things out.

In the end, though, the best evidence we have for how the web of linked data will progress is the evidence of how things were for the human web. It is hard to be an early adopter, both for social reasons and technological reasons. Nothing will happen overnight, but gradually there will be network effects: more shared URIs, more shared vocabularies, making it both easier to adopt and more beneficial for everyone.

Is this a kind of faith? Maybe. I believe in the web.

*The content of this white paper is reproduced with permission from Jeni Tennison.*

## About TSO

TSO (The Stationery Office) is the leading provider of information management and publishing solutions to the public sector. We are the largest publisher by volume, publishing more than 8,000 titles a year in print and digital formats. Our experts help to create, structure, capture, transform and deliver some of the most important government information. TSO provides services, consultancy and infrastructure to deliver all aspects of the information lifecycle to the highest standards for our clients.

TSO has been at the forefront of working with public sector clients to open up published data. We create tools and processes to allow data to be created in a structured way; enhance data using text engineering techniques; convert data into formats to publish as linked data on the web and provide and host web environments that allow both humans and machines to access the data.

To find out how TSO can help your organisation to publish open linked data visit our website:

[www.tso.co.uk/opendata](http://www.tso.co.uk/opendata)

Sourced by TSO and published on [www.tso.co.uk](http://www.tso.co.uk)

Our White Paper series should not be taken as constituting advice of any sort and no liability is accepted for any loss resulting from use of or reliance on its content. While every effort is made to ensure the accuracy and reliability of the information, TSO cannot accept responsibility for errors, omissions or inaccuracies.

### CONFIDENTIALITY STATEMENT

The contents of this document together with all other information, data, materials, specifications or other related documents provided by TSO (together "materials") shall be treated at all times by the recipient as the confidential and proprietary information of TSO. The recipient shall not disclose any such materials to any third parties without the express, advance written approval of TSO. Where such express approval is granted by TSO, the recipient shall ensure that all third parties to whom disclosure is made shall keep any such materials confidential and shall not disclose them or any part of them to any other person. All intellectual property rights in the materials provided by TSO are and shall remain the property of TSO, or its third party licensors, and are protected by copyright.

© 2008 Williams Lea Group

### DISCLAIMER

This document may be incomplete without reference to any oral briefing provided by TSO, reflects current conditions and TSO's views as of this date and is subject to correction or change at any time. Although the information contained in this document is believed to be accurate in all material respects, neither TSO nor any of TSO's advisers, agents, officers or employees accepts responsibility or liability for or makes any promise, representation, statement or expression of opinion or warranty, express or implied, with respect to the accuracy or completeness of the content of this document (to the extent permissible by law) unless and save to the extent that such promise, representation, statement or expression of opinion or warranty is later expressly incorporated into a legally binding contract.



Part of the Williams Lea Group