„nn

University of Glasgow

VIA VERITAS VITA

Gray, N. (2015) RDF, the Semantic Web, Jordan, Jordan and Jordan. In: Moss, M. and Currall, J. (eds.) *Transition to the Digital*. Facet Publishing.

http://eprints.gla.ac.uk/101484/

RDF, the Semantic Web, Jordan, Jordan and Jordan

Norman Gray[1]

24 August 2014

> *DRAFT chapter to appear in 'Transition to the Digital', Michael Moss and James Currall (eds.) (2015?). This collection is addressed to archivists and library professionals, and so has a slight focus on implications implications for them. This chapter is nonetheless intended to be a more-or-less generic introduction to the Semantic Web and RDF, which isn't specific to that domain. Copyright Norman Gray 2014.*

This chapter is about the key novelties of the Semantic Web – the novel ideas, and the novel opportunities. But we will discuss these digital novelties in the context of the Semantic Web's *continuities* with other features of the information world.

Our most obvious antecedent is not that old – the (non-semantic) Web didn't exist before the 90s – and we will learn about the very close technical overlap between the Semantic Web and the 'textual Web' of our now-usual experience. The Semantic Web is, closer than a cousin, the sibling of the textual web.

Other antecedents have a history as old as the first library index. The Semantic Web has, we might say, a 'logical wing' and an 'information wing'. These are not primarily distinguished by their technical or organisational features, but by the largely disjoint research questions they address, and by their motivations. While the 'logical wing' is characterised by a concern for formal logic and its implementations, rich in the theory of computing science[1], the 'information wing', with a sturdily pragmatic focus, can be regarded as continuous with the information-organising goals of the world of library science, sharing its aspiration to systematize and share information, and its acknowledgement that such sharing is always approximate and never unmediated, and that one must aim for a balance between faithfulness to sources, and what is actually usable by the information's actual audience.

Below, we will start with a broad introduction to the Semantic Web. From there we can move briskly on to practice, and the question of where, how and whether the semantic web might appear in technological fact. Our goal is to indicate the continuities with the textual web, and thus to indicate the novelties of the Semantic Web, and so suggest why they are important. After a brief parenthesis on 'Web 2.0' (in Sect. 2), we describe 'linked data' in Sect. 3.

1 What is the Semantic Web?

The Semantic Web is simple in summary:

> *The Semantic Web is the emerging next stage of the web, designed to transmit machine-processable meaning, through a logical framework*

[1] Physics and Astronomy, University of Glasgow, UK; `http://nxg.me.uk`

[1] This strand can with at least a little justice be regarded as the most recent last hurrah of AI, a tradition pronounced dead more often than *South Park's* Kenny.

> *named RDF, enhanced by machine inference based on OWL and other ontologies*.

Though I assert that this modest statement is plausible, and the outcome desirable, the reader may be disinclined to agree, on the grounds that the statement is on the face of things gobbledegook. Over the course of this chapter, I intend to explain each component of this remark, step by step, in the hope that it is a short hop from there to plausibility.

1.1 The Semantic Web...

First, a general lament.

The term 'semantic web' is an unfortunate one, since it makes the topic sound much more arcane than it really is. It is arguably simple: the next stage in a long-term vision of the web, originally formulated by Tim Berners-Lee, and elaborated by the World Wide Web Consortium (W3C) which he was instrumental in founding.

1.2 ...is the emerging next stage of the web...

Before we can fully understand the 'semantic web', we need a clear idea of what the 'World Wide Web' is, even though, nowadays, such a question may seem as odd as asking what 'air' is.

The web is *remarkably* homogeneous: it consists of *one* protocol, *one* bit of glue, and markup.

Everything on the web is connected by the Hypertext Transport Protocol (HTTP) [1], and if you look at a web page, update a podcast, talk on Jabber, or download videos, on a computer or a mobile phone, the bytes come to your computer via HTTP. Indeed, a debatable but plausible definition of the web is as the set of things reachable through an HTTP request. The only bit of that protocol you ever notice is '404', which is the HTTP error code meaning 'I don't have anything by that name'.

The bit of glue is the Uniform Resource Identifier (URI) [2]. That's a uniform *naming* scheme for things on the web and beyond. Everything on the web has a URI, and most URIs refer to things on the web.[2]

The markup is mostly Hypertext Markup Language (HTML), the angle-brackets which indicate how web pages should be formatted. But it's also RSS and Atom feeds (ie, blogs and podcasts), wiki syntax, PDFs, and a few other more obscure alternatives.

These components come together when you click on a link in a web page (the following explanation of how the web works is probably not new to you, but I'm spelling it out in order to make clear how the various components work together). The web browser program on your computer or tablet or phone is a *client*, which displays a page previously sent to it by a program sitting on a *server* somewhere in the internet (we will use these latter terms repeatedly below). The page (typically) arrives at your computer in the form of HTML which the browser knows how to format and display as headings, sidebars, images, and links. The link associates some text on the page with a URI, and clicking on it tells the browser 'go and look at this page instead'. The browser then examines the URI to discover which web server is providing that page, then immediately makes another HTTP request to that server to retrieve the page, display it to you, and start the cycle once more.

[2] That's a sort of 'mindspace' most: this statement probably isn't numerically true if you go by numbers of objects or volume of data.

Before we go on, I should (parenthetically) be careful to distinguish the web from the *internet*. The internet is a set of protocols, plural, for exchanging material between networked computers. When you send an email or retrieve a web page, the material travels over the internet, but under the control of different programs – an email client and a web browser – using a combination of lower- and higher-level protocols, or languages.[3] Internet telephony (such as Skype) and time services are two reasonably visible internet services which are distinct from the web. The key point is that the web – and this includes the Semantic Web – is a notably simple structure sitting atop the internet.

There are a few key dates in the history of the web.

- 1990: Tim Berners-Lee and Robert Caillau first proposed the system which became the World Wide Web [3].[4] The first server and client implementations appeared as CERN-internal services later that year.
- August 1991: First public web server. Initially, users interacted with the server by using Telnet (another non-Web internet protocol) to connect directly to a client program at CERN, and thence to the server.
- December 1991: First web server outside Europe, at the Stanford Linear Accelerator Center (SLAC, like CERN an experimental high-energy physics laboratory).
- April 1993: Mosaic, from the (US) National Center for Supercomputing Applications (NCSA), was the first graphical browser. Mosaic led to Mozilla, which led to today's Firefox; Mozilla also led to the Netscape browser. About the same time, NCSA released their 'httpd' web server, which eventually mutated into the now-ubiquitous Apache.[5]
- October 1994: The World Wide Web Consortium (W3C) was founded, with Berners-Lee as its director, to act as the standards body of the emerging system. Early standards included HTML 3.2 (1997),[6] the first version of XML (1998) [5], and the first model and syntax for RDF (1999) [6].
- 2006: Lolcats (and other 'user-generated content'). We return to this in Sect. 2

Although it may seem a slight tangent, it is worthwhile briefly discussing what it is that makes the web so special, and so very successful, since the semantic web is in protocol terms identical to the web we are familiar with, and so shares the same special features.

The web is not the first hypertext system (Vannevar Bush's hypothetical *Memex* system [7], and Ted Nelson's *Xanadu* system[7] can probably lay claim to that), and it's not the first distributed hypertext system (VAX Notes and Lotus Notes can probably claim that), but it is the first truly successful worldwide distributed hypertext system.

The web gets some things right:

[3] This point is rather muddied by the observation that many people now read email messages in a web browser; nonetheless the email is still transported between systems, in the background, using the decades-old email protocols.

[4] This 1990 document comes about 18 months after [4], which is a broad discussion of information management, in the context of "the management of general information about accelerators and experiments at CERN." This document – on which Berners-Lee's manager wrote "vague, but exciting" – is clearly ancestral to both the web and to [3], but it is particularly interesting from the point of view of this chapter, because in hindsight it more closely prefigures the structure and potential of RDF and the *Semantic* Web. One could almost call the textual web 'Semantic Web 0.1'.

[5] Apache originally consisted of a set of software 'patches' to NCSA `httpd`, hence the (unfortunately apocryphal, it seems) etymology of its name as 'a patchy server'.

[6] HTML 2.0 was an IETF standard; earlier versions were not formally standardised.

[7] `http://xanadu.com/`

- It is distributed, or decentralised, so that there is no *centre* to the web – there is no single point which can fail, or be co-opted, or which can grant or refuse permission.
- It is non-proprietary, or open: the protocols which define the web are free to obtain, and may be implemented without any inhibitions from licences. Also, the web's governing body (which the W3C is, in effect) does its work through an open process.
- The web is simple, in the sense that it is architecturally simple (the notion of the web, as servers plus browsers plus links, is easy to comprehend), and straightforward in protocol terms (the core protocol of the web, HTTP, is such that a crude server or client implementation can be developed in a relatively short time).
- It's easy to join in: this is to a large extent a consequence of the simplicity and openness.
- You can waste hours on Wikipedia.

One sense of 'easy to join in' is that the web has *always* been read-write: Berners-Lee conceived it as a read-write medium, the first browsers could write to the web as well as read it, and 'anyone' can put up a webpage. Now, 'anyone' here meant 'anyone with access to a unix box connected to the internet, who can build, configure and install a web server'; so not quite everyone's 'anyone', but the key point is that it was usability that stopped you from putting up a web page, not the need for permission. Thus the (genuinely) big innovation of what became known as 'Web 2.0', or 'the read/write web', was that it was 'Web 1.0' for everyone else.

The last point in the list is not frivolous, but is the point that, on the web, you can link to *anything else on the web*, and that this is both culturally acceptable and encouraged. You can wander through a lot of web servers by starting at one page and 'following your nose'. This is a consequence of the simplicity and openness, which together mean that there are very many web servers, from very heterogeneous sources, serving web pages written by experts and non-experts, and that linking between these is not inhibited by any requirement for pre-coordination.

The web's success is also partly due to getting some things *wrong*:

1. links can break, and pages disappear;
2. links are *all* 'see also' (as opposed to 'parent', 'next', 'author' or anything more informative);
3. everything is a string – the information on the web is, fundamentally, communicated through text.[8]

One might also say that 'no quality control' is a vice, but since one can simultaneously claim that 'no control at all' is a virtue, this is at least debatable.

Xanadu, for example, guaranteed link integrity, had typed links, and tried to develop a new intellectual property model, all at the same time. Berners-Lee and Caillau's insight, in [3], was that these are simply ignorable problems – it's acceptable for things to break, and occasional 404s are a price worth paying to avoid the need for pre-coordination or registration; and the web's type-less and one-directional link is such a powerful notion that its intrinsic vagueness is only

[8] The HTTP protocol can be, and is, used to transport digital media of all types – plain text, images, PDFs, audio, and other data; for our present purposes, I take the term 'the web' to refer to the global hypertext system of Berners-Lee's original conception; this does not affect the essential point.

a detail. The point of the semantic web is that it starts to address the second and third problems.

Why is 'everything is a string' a problem? If you search on the web for 'jordan', you get links about the country, the river, the glamour model, the breakfast cereal, the brand of shoes, various small businesses, the basketball player, the mathematician, and more. These are not all the same thing.

This doesn't matter, however, because we as humans know they're different things, and we're not likely to get confused (and if we are, briefly, confused we think it's our fault, rather than Google's). We can, in other words, add the semantics ourselves, in exactly the same way that we do so when reading text anywhere else. But this means that computers are flying blind when they try to perform actions on the web on our behalf. They have no idea what all these strings mean, nor (more importantly) how they relate to one another.

The web works very well for many things, and search engines of course work spookily well in many cases.[9] But it only really works when there's a human in the loop, or where there's a lot of statistics to build on, and you wouldn't want to let your computer unsupervised onto the web, with your credit card.

But in fact you *do* want to let your computer onto the web with your credit card. In a famous and seminal paper, Berners-Lee, Hendler and Lassila [8] describe an extended scenario in which computers are able to interact with each other to organise travel and other appointments. This scenario has come about to some extent: price-comparison websites now routinely interact with other websites to extract price data; and sites like `tripit.com` work by parsing the confirmation emails from travel websites such as Expedia, to extract the details of flights and hotel stays. The problem is that these services work by painfully scanning the content of webpages or emails directed at humans (this is known as 'screenscraping'), heuristically parsing them, and acting on what may or may not be the intended meaning.

This is hard work for limited results, but (at least until computers somehow manage to understand text) it is a fundamental limitation of the *web of strings*.

1.3 ...designed...

I have stressed that Berners-Lee and Caillau's original conception is architecturally still very close to the web we see now, 25 years later. Berners-Lee went on to form the W3C, and it is this body which by general consent – for no compulsion is possible here – still shepherds the development of the collection of standards which underlies the web (other bodies, most notably the IETF, are responsible for the technical governance and development of the internet, by a similar process of general consent). Although the core standards are easily identified – HTTP, URIs and HTML – these are accompanied by a blizzard of other agreements, major and minor, ranging from the syntax of XML to consensus on the formatting of dates. These agreements take the form of 'W3C Recommendations',[10] collaboratively authored by W3C Working Groups formally drawn from academic and commercial W3C member organisations, but with wide and occasionally noisy participation from interested individuals world-wide.

Much of the development work for the semantic web, in particular, has come from universities and a small number of research-active commercial organisations, often with quite close links to academia. There is a strong inheritance

[9] If you actually search for 'jordan' in, for example, Google, Bing, and DuckDuckGo, you will find that *several* of these distinct senses appear on the first page, so that there is more variety than would result from simply listing the most popular pages with that string in them. This is because search engines can *statistically* identify that there are multiple clusters of related pages and thus, for example, display one hit from each cluster in a situation like this. The search engines are understood to improve their performance here with some lightweight semantics, but the core work of a search engine is at present still probabilistic rather than logical.

[10] `http://www.w3.org/standards/`

from preceding decades of research on Artificial Intelligence (AI); this inheritance included crucial experience of the logical underpinnings of what became RDF (see Sect. 1.5 below), and experience with related technologies, which meant that the first RDF standards were remarkably self-consistent and well-developed. However this same process meant that these first standards were rather hermetic, which will have contributed to their early reputation for incomprehensibility.

However comprehensible these standards are – and we will touch on the core ideas below – it is important to stress that it does not require some foundational understanding of the mathematico-logical foundations of RDF in order to describe, for example, a book's bibliographical details. In recent years, the 'linked data' paradigm has emerged, again with both academic and industry backing, with a focus on the practical steps to bring about the immediate-term payoffs of the semantic web. We discuss this in a little more detail below, in Sect. 3.

1.4 …to transmit machine processable meaning…

The Semantic Web has some family relationships with the world of AI, and indeed has suffered, in marketing terms, from its association with that discipline's repeated postponements of its great promises. The Semantic Web is not concerned with machine understanding, or with machines' creation of meaning, but instead with the more modest goal of transmitting meaning across the web, in a more reliable way than is possible with the web of strings.

Our confusion, above, about the different things called 'jordan' arises because these various very different things all share a single label – 'jordan' – and it is only our ability to understand the context, and our knowledge of the structure of the world, that allows us in practical fact to distinguish the various denotations (we do not get confused when the term 'jordan' appears on a news website's politics and celebrity pages, denoting different things). Faced with the same text, computers can do little more than (be programmed to) rely on statistics and heuristics. Search engines show that this approach is more successful than one might expect, but until computers do finally manage to 'understand' things (if they ever do), there is little more we can do, if we stick with just the text.

Semantic Web technologies allow us to take a step beyond these limitations, by

- providing a foundation with which to define more specific labels for the concepts and categories which make up our world; and
- allowing us to manipulate these labels (and by analogy manipulate the concepts and categories) in a more or less simple calculus of relationships.

An example is useful here.

One name for the River Jordan is `http://dbpedia.org/resource/Jordan_River`. This derives from DBPedia [9], which is a collection of Semantic Web names derived directly from Wikipedia. The name `http://sws.geonames.org/7874114/` is another one, which derives from the Geonames database.[11] The resemblance of these 'names' to ordinary web URLs is not a coincidence – all the names within the Semantic Web are syntactically of this form. One of the reasons for this is that it preserves the 'decentred' property of the (text) web: the owners of the `geonames.org` domain can create what names they like within that domain, since 'creating a name' in this context consists solely of deciding that a particular URI should act as such a persistent name, and providing RDF (see Sect. 1.5) to describe it. This has the immediate attractive property that (usually) one can find more about a particular name by typing the name into an ordinary

[11] `http://www.geonames.org`

web browser; although it's not a requirement, the usual good practice is for such a URI to produce a human-readable description of some type.

Anyone can create such a name: I own the domain `nxg.me.uk` and decided, by personal fiat, that `http://nxg.me.uk/norman/` should be my 'name' in this context. After a short bit of website configuration, it was so.

Once we have names, we can start to describe the things so named.[12]

Since anyone can – and many do – create online names for things, one of the first things we might want to do is to address the apparent problem of duplication. So, for example, we might want to say that `http://dbpedia.org/resource/Jordan_River` is the same thing as `http://sws.geonames.org/7874114/`. This will be true in some contexts, but false in others. In this case, it turns out that the geonames.org designers have decided that `http://sws.geonames.org/7874114/` refers to the River Jordan *in Jordan*, and that the 'same' river in Israel is named `http://sws.geonames.org/294624/`. Therefore, in some contexts, it would be simply false (and importantly false) to state that `http://dbpedia.org/resource/Jordan_River` is 'the same as' one or the other, and we might find it better to say that `http://sws.geonames.org/294624/` is 'part of' `http://dbpedia.org/resource/Jordan_River`. This sort of (often barely consistent) subtlety is something which we are comfortable with in human conversation, but which we have to spell out in pedantic detail as soon as computers are involved.[13]

Other things we might want to say are that `http://dbpedia.org/resource/Jordan_River` has a certain length, that it is a member of the category of 'rivers', that 'rivers' are a type of geographical feature, that the river is *located within* (as opposed to being a component part of) both Israel and Jordan; we might want a computer to 'know' that the categories of 'rivers' and (for example) 'cats' are disjoint (computers are *very* ignorant).

The term 'know', here, does not of course refer to any cognition on the part of a machine. Instead, it refers to a calculus of manipulations which would allow it to detect that a statement that '`http://dbpedia.org/resource/Jordan_River` is a cat' is inconsistent with what it has already stored, or that a search for 'things labelled "jordan" which are geographical features' should not return any basketball players.

This last point indicates one fragment of where the applications of the Semantic Web might lie. We know how to store and manipulate *facts* on computer – we can match an employee number with a name and department, or a star at particular coordinates with brightness and colour – but often these facts lack the real-world structure and interrelationships that are so important to how we as humans want to handle them. Even when such knowledge is structured in some contexts, so that the internal structure of a library catalogue might reflect the real-world structure of the organisation that maintains it, this structuring is not shareable without close prior cooperation. If I emailed you a copy of such a catalogue as a spreadsheet, you as a human would swiftly work out what this document was and what some of the columns meant, though you might (for example) have difficulty telling apart the columns with publication year, acquisition year and, say, the year the book was first borrowed. I would have to talk to you to tell you which column was which; neither of us would expect the spreadsheet

[12] I note parenthetically – for there is a *very very long* distraction possible here – that the URIs `http://dbpedia.org/resource/Jordan_River` and `http://nxg.me.uk/norman/` are names for the physical things themselves, and not for any online resource describing the things; in the same sense, a DOI is a digital identifier for an object such as an article, and not for the record describing the article. This is true even though both of them have the form of a URI which is retrievable (or '*dereferenceable*') in the sense that a web browser can retrieve something from that address. If you retrieve either of those URLs by typing it into a browser, you are redirected to another web page which describes the resource (that is, a resource with a different name, which *is* a webpage rather than a person or concept); in neither case do you get me, or a river-ful of water, delivered to your computer. If you are interested, and have a day or so to burn, search for 'httprange-14'.

[13] We make some extra remarks about the logic of RDF, and identity, in Sect. 1.5.

to be usable on a computer without this human-to-human interaction.

The goal of the Semantic Web is to make this sort of structured knowledge – about geography, stars, people, or retail opportunities – generally shareable and manipulable by computers, while preserving as many as possible of the (text) web's properties of decentralisation (there is no permission or coordination required), openness (what we can communicate is not limited *a priori*), and simplicity (there is still only one protocol, HTTP).

That is, the Semantic Web is not about machine *understanding*, but about the technology required to let computers manipulate these URI-based names in ways which are, as far as possible, not inconsistent with the properties of the real-world objects for which they are analogues. This analogy – the functional consistency between the behaviour of real-world objects and the declared rules for manipulating their names in the machine – is the 'semantics' in the Semantic Web.

1.5 …through a logical framework named RDF…

So far, so abstract. How do we actually *write down* the statement that '`http://dbpedia.org/resource/Jordan_River` is a river'? For this, we must examine the Resource Description Framework (RDF).

RDF is a *framework* for describing things, and their mutual relations. It's a rather abstract framework for thinking about the mechanics here, and is not, itself, a specific data *format* or language (though there are formats and languages intimately associated with it).

In the logical language in which it was first described, RDF is rather simple (or at least compact, which is not necessarily the same thing). It seems best, therefore, to follow the overall plan of this section and give the compact explanation first, and subsequently gloss it at a little more length. So (omitting a few details which are unimportant at this point):

- The world is described by a mixture of *resources* and *literals*.
- *Literals* are simply strings, such as "River Jordan" or "Río Jordán".
- *Resources* are things in the world such as individuals, categories of things (such as people or rivers), abstract concepts ('world peace'), things on the web or off it, or indeed *anything that can be given a name*.
- Resources are given *names* which are all syntactically URIs.
- One can make *statements* about resources, all of which are of the form of a *triple* of subject (the resource in question), predicate, and object. Subjects and predicates are always resources; objects may be resources or literals.

RDF version 1.1 is formally described in [10].[14]

To say more, we will have to write down RDF. There is no single RDF syntax; we will pick the syntax Turtle [11] from the available options. In this syntax, we can write

```
<http://nxg.me.uk/norman/>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
    <http://xmlns.com/foaf/0.1/Person>.

<http://nxg.me.uk/norman/>
```

[14] This replaces the original version 1.0; there are further documents, including links to primers at different technical levels, at `http://www.w3.org/standards/techs/rdf`, and a broader collection of resources at `http://www.w3.org/RDF/`. The RDF suite of documents was comprehensively refreshed in February 2014, with the release of new versions of many of the standards which had accumulated over the previous 15 years.

```
  <http://xmlns.com/foaf/0.1/name>
    "Norman Gray".
```

There are two statements (ie, triples) here, one stating that `http://nxg.me.uk/norman/` is a Person, and the other stating that resource's name. In each case, we can see the structure of the subject-predicate-object triple, with the resource `http://nxg.me.uk/norman/` being the *subject* in both cases, and the resource `http://xmlns.com/foaf/0.1/Person` and literal `"Norman Gray"` being two *objects*. The first predicate – stating the type of the resource – is one of the predicates defined in the W3C's RDF 'Schema' standard (RDFS) [12]; the type in question is one defined by the independently defined Friend of a Friend (FOAF) schema (which we will return to below). The FOAF schema also defines a 'name' predicate, which gives the literal, conventional, name of the resource that is its subject.

This notation is obviously rather cumbersome. The Turtle syntax states that we can abbreviate the 'type' predicate by just '`a`', that we can collapse repeated subjects, and that we can provide abbreviations for cumbersomely long URIs; as a result, the more idiomatic way of representing the statements above is just

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.

<http://nxg.me.uk/norman/>
  a foaf:Person;
  foaf:name "Norman Gray".
```

There are a few more subtleties to this notation, which can be found in [11], but they need not detain us, since our goal here is simply to concretize the generalities of earlier sections, and to allow us to use this notation in examples below.

You may object that this seems a terribly long-winded way to indicate someone's name, and you would be right. There is another notation called RDF/XML [13] which is even more long-winded, which only its mother could love, and which we will pass over in silence. And there is yet another RDF notation called RDFa[15] which is concerned with embedding RDF statements within other documents, and most particularly within human-readable HTML pages; the intention is that the same document that a human reads describing, for example, a library book or a 'retail opportunity', is at the same time reliably interpretable by machine. But we must not allow ourselves be distracted by syntax.

The central goal of RDF is not to provide a data-transport format (and here is a good point to emphasise that the 'F' in RDF stands for 'framework', and not 'format'), but to provide a set of primitive notions with which we can represent a wide breadth of knowledge. Those notions are simple enough that it is feasible to translate a wide variety of other, possibly more natural, formats into RDF – for example the rows in a database table, or the elements in an XML file; and they are well-enough defined that we can use logical tools to process the results and draw out their implications, while standing a good chance of preserving their real-world meanings (cf. Sect. 1.6).

Saying that we need not store or transport knowledge in this form is not saying that we should not do so, and it is perfectly reasonable to store and transport RDF if that is what is convenient. The databases in which one stores RDF is called a 'triple-store', after the subject-predicate-object triples they contain; they are architecturally different from the relational databases of tables, rows and columns, with which you might be more familiar, and although they are not quite as technically mature as relational databases, they are steadily improving.

As a final point, the statements of RDF are not required to be true, consistent, or even meaningful (you can say 'Truth smells of muesli' if you want to, or 'the present King of France is bald') – anxieties about such statements are for the higher levels of inference and ontologies that we will come to shortly.

[15] `https://www.w3.org/standards/techs/rdfa`

A brief logical excursion, for the enthusiast: RDF by itself has an exceedingly simple core semantics. The link between a name (in the form of a URI) and its reference (in RDF terms, a 'resource') is made in natural language, and is not expected to be meaningful to a machine. RDF does not distinguish between, for example, sense and denotation (in Frege's terminology), and is uncommitted about the identity or otherwise of resources; thus it is possible to name both Mark Twain and Samuel Clemens with URIs, and it is not possible, within RDF alone, to make statements about their mutual identity or otherwise. OWL ontologies (see below) are able to make statements about equivalence, but even here the framework is uncommitted about what equivalence means (it does not, for example, distinguish sense and denotation), and in one context it may, and in another may not, be useful for the author of a statement/triple to state that Mark Twain and Samuel Clemens, are 'equivalent resources', or the model Jordan and the écrivaine Katie Price, or the Geonames and DBPedia names for the River Jordan. Once an association has been made between a name and a referent, the goal of RDF is to allow this association to be preserved across machines and networks, and to allow machines provided with both RDF statements and ontologies (which may come from different sources) to discover the statements entailed by the ontologies, which have the same truth values as the initial RDF. For further detailed logical discussion, see [14].

1.6 …enhanced by machine inference…

In the Turtle example above, I 'described' the resource `http://nxg.me.uk/norman/` by saying that the thing with that name is a `foaf:Person`, whose `foaf:name` is "Norman Gray". These are two terms from a simple *ontology* (see Sect. 1.7) called FOAF.[16] The FOAF ontology has a *namespace* `http://xmlns.com/foaf/0.1/`, meaning that all of its types and predicates start with that URI. It includes a number of types such as 'Person' and 'Project', and a number of relations such as 'name', 'mbox' (email address), 'publications' and so on. Another ontology, well-known in the library community, is Dublin Core (DC). This is described authoritatively in `http://dublincore.org/documents/dcmi-terms/`, and the RDF expression of this vocabulary describes the namespace `http://purl.org/dc/terms/`, so that the DC 'title' predicate is, in full, `http://purl.org/dc/terms/title`, which is most often seen abbreviated to just `dc:title`. The DC ontology is focused, at least initially, on metadata for bibliographic and archival resources, and includes relations such as 'title', 'creator', and so on.

Importantly, these various terms have no intrinsic meaning, and RDF places no restrictions on the predicates attached to a resource, nor who attaches them. Thus if you wish to say

```
<http://nxg.me.uk/norman/>
    foaf:name "Capitania General Bernardo O'Higgins";
    myprops:shirtType "class C".
```

then you are at perfect liberty to do so, even although the first statement (stating that my name is Bernardo O'Higgins) is false, and the second (apparently saying something about my shirt) is meaningless to me, though doubtless somehow useful to you.

The primary goal of these various relations is to describe a relationship sufficiently unambiguously that even a machine can process it. Thus a 'title' in the DC sense – that is, the object of the `dc:title` predicate – is always a bibliographic title, such as `dc:title "Transition to the Digital"`. The FOAF ontology also has a 'title' predicate, but in this case it is always and only a person's honorific: `<http://nxg.me.uk/norman/> foaf:title "Dr"`.

[16] The 'Friend of a Friend' ontology was originally conceived of as a way of creating a peer-to-peer social network.

```
<http://nxg.me.uk/norman/>
    foaf:name "Norman Gray";
    foaf:mbox <mailto:norman@astro.gla.ac.uk>.

<http://example.org/a.n.other>
    a foaf:Person;
    foaf:name "Aloysius Naismythe Other".

<http://someone.org/ping>
    foaf:mbox <mailto:norman@astro.gla.ac.uk>.

<isbn:????> # XXX INSERT BOOK'S ISBN HERE
    dc:title "Transition to the Digital;
    dc:contributor <http://someone.org/ping>.
```

Figure 1: Some statements about people.

But we can do more than this.

Consider Fig. 1: this provides a FOAF name and email address for `http://nxg.me.uk/norman/`, states that `http://example.org/a.n.other` is a (FOAF) Person, and provides an email address for an otherwise unidentified something named `http://someone.org/ping`. You might already have a picture of the entities being described here, in terms of their number and type.

If you were to put this descriptions in Fig. 1 into a triple-store, and then query it to retrieve all of the Persons described, you would obtain only a single Person, namely `http://example.org/a.n.other`, because this is the only resource which has been explicitly stated to be of type `foaf:Person`. It is obvious to us as humans, however, that if something has a name and an email address, then it's almost certainly human, and the FOAF specification indeed makes this stipulation, that a `foaf:mbox` property can be attached only to a `foaf:Person`. In formal language, we can say that `foaf:Person` is the ‘*domain*’ of the `foaf:mbox` and `foaf:name` predicates, meaning that only things of type `foaf:Person` are permitted to have those predicates (this is distinct from the use of ‘domain’ to name a collection of machines on the internet, as in for example `www.w3.org`); similarly, we can say that `dc:Agent` is the ‘range’ of the `dc:contributor` predicate, meaning that any object of this predicate can be deduced to be a `dc:Agent`. In consequence, an RDF triple-store with this extra information (more precisely, a triple-store possessed of basic inferencing capabilities) can *deduce* that `/norman/` and `/ping` are Persons, and so would give three answers, when asked to list the Persons it knows about.

A further thing that humans know is that, while individuals may have multiple email addresses, an address is usually owned by a single person. FOAF agrees: only a single `foaf:Person` can be the subject of a `foaf:mbox` property. But both `/norman/` and `/ping` have this property with the same value. The resolution is straightforward: these two URIs can be deduced to be merely different names for the same Person. Primed with this extra information about the domain of `foaf:mbox`, a triple-store, asked how many Persons were represented in Fig. 1, would answer ‘two’. Such a triple-store can answer more complicated queries without getting confused. Asked for the titles of things to which the person named `"Norman Gray"` has contributed, a triple-store provided with Fig. 1 could answer `"Transition to the Digital"`, by following the chain of allowed *inferences* – it synthesises statements, such as “`http://nxg.me.uk/norman/` is a `foaf:Person`”, and “`/norman/` sameAs `/ping`”, which are only implicit in the original set of statements.

The end result, as may now be clear, is that a triple-store can aggregate information from multiple sources: the information in Fig. 1 might have been gathered from a publisher's website, a membership database, and an individual's home page, and may have started off in any or all of RDFa, XML, or a relational database. Once it is gathered, the triple-store can draw the conclusions which are not apparent from any data source by itself.

Before going on to describe briefly just how these various relations are articulated, we have a few final remarks to make. Firstly, when we say, above, that "a `foaf:mbox` property can be attached only to a `foaf:Person`", we are saying something different from the apparently similar statement we might find in an XML schema. In XML, such a statement is a syntactical one, saying that an 'mbox' element (in that example) is *permitted* to be attached only to a Person, so that if it is attached to something which isn't a Person, or at least isn't known to be a Person, then this is an error. In RDF, in contrast, the statement means that anything to which the `foaf:mbox` property is attached must be *deduced* to be a Person. It is not even illegitimate (although it is in fact untrue) to say that

```
<http://nxg.me.uk/norman/> foaf:name "Norman Gray";
  dc:bibliographicCitation "Gray (2014)".
```

even though the domain of `dc:bibliographicCitation` allows a triple-store to deduce that `/norman/` is of type `dc:BibliographicResource` (that is, something like a book, that can have 'Gray (2014)' as its citation). Only if the store is subsequently told that `foaf:Person` and `dc:BibliographicResource` are *disjoint* will it raise any objection, and that objection will not be a syntactic one, but the announcement of a logical inconsistency.[17]

Secondly, we might reasonably object that, in the real world, a library (which is not a `foaf:Person`) might have an email address for enquiries, and that a role-based email address might in fact be shared by multiple people. This is accurate, but the restrictions we placed on the `foaf:mbox` property above are part of the approximation to the real world that we must make when we describe that world in terms simple enough for a computer. The library's email address is therefore not the object of any `foaf:mbox` property, for precisely the reason that a library is not a `foaf:Person`; and similarly for the role-based address, for precisely the reason that it is incompatible with the definition of `foaf:mbox` for an email address to have multiple owners. A different 'people' ontology might have different restrictions, and so permit different implications.

Another logical excursion: Above, we mentioned the idea that (some of) the River Jordan is located within Jordan, as opposed to being part of it. The truth of this statements depends in part on what, precisely, 'is part of' means, and so potentially drags in a set of philosophical consequences, and indeed political and cultural ones; but the only ones that matters to the computer are the logical consequences: given the information that 'A is part of B', what other statements is it permitted to synthesise? That is a largely technical problem concerning what `isPartOf` or `sameAs` is intended to mean in a particular system, and it corresponds to the human problem of articulating what one knows within a formal system, in a way which allows the machine to draw the conclusions one expects. It is possible to spend a good deal of work-day effort arguing about just what 'is' is. These questions are what permit an ontology to have sufficient structure to make inferences possible and useful, and working out the necessary balance between approximation and expressiveness is what makes the design of ontologies challenging. All that said, such exotic questions are largely irrelevant to the *users* of

[17] Saying that the two classes are *disjoint* is to say that there are no objects which are in both classes; this is incompatible with the deduction, above, that `/norman/` *is* in both classes, and it is this inconsistency that machine will object to.

an ontology, as long as they are broadly aware of the potential dislocation between statements in the formal language and timeless statements about the real world.

Thus we can see that RDF statements, and the ontologies which structure them, are not really, or not only, about *truth*, other than in the abstract logical sense that valid arguments must preserve the truth values of their inputs. Similarly, we are not concerned with developing a single true ontology of the world, or even with the claim that an ontology which is useful in one context, will be useful or even meaningful in another, or that two ontologies describing one object will necessarily be commensurable. Developing ontologies is a type of programming activity, and the types and predicates, and the relationships between them, must be chosen to match the things being modelled, to accept the precision of what can be said about them, and to support the range of conclusions one may hope to draw.

FOAF and DC are relatively simple ontologies, with rather lightweight constraints; they are therefore very widely applicable for describing people and artefacts, respectively, and, in consequence, are very useful for integrating otherwise rather disconnected input datasets. In contrast, a large ontology such as the Gene Ontology [15, 16] is concerned with a logical structure which is sufficiently intricate, and sufficiently closely mapped to nature, that the ontology can draw scientifically valid conclusions, at the cost of being considerably harder to use.

In the examples above, we have talked of various ways of adding structure to the list of types and predicates in an ontology; now is the time to discuss briefly how this actually happens.

1.7 …based on OWL and other ontologies.

The internal structure of ontologies is the most technically intricate part of our story here. Fortunately, it is not necessary to discuss that structure in deep detail: our goal in this section is simply to concretize the rather general statements about ontology structure above, and to give general pointers towards more detailed advice.

There's a well-known description of an ontology by Thomas Gruber [17],[18] in which he declares that an ontology is

> "a formal specification of a shared conceptualization"

That appears opaque at first glance, but in fact it *very* concisely pulls together all of the key concepts. That is to say, an ontology is

conceptualization a set of concepts

shared …which at least two people agree about

specification …and which has been written down

formal …in a machine-readable way.

Like 'semantics', the term 'ontology' has a forbidding aspect. In this context, however, it simply labels one end of a spectrum of ways of structuring information. In Deborah McGuinness's 'semantic spectrum' (in [18], and redrawn in Fig. 2) she illustrates a range of 'shared conceptualizations', and suggests that the term 'ontology' is most naturally restricted to those more formal structures to the right of the central line, dividing informal from formal 'is-a' relations; Gruber's definition applies to most of this spectrum.

[18] This is the by now apparently traditional slight misquotation of Gruber's 'explicit specification' as 'formal specification'.

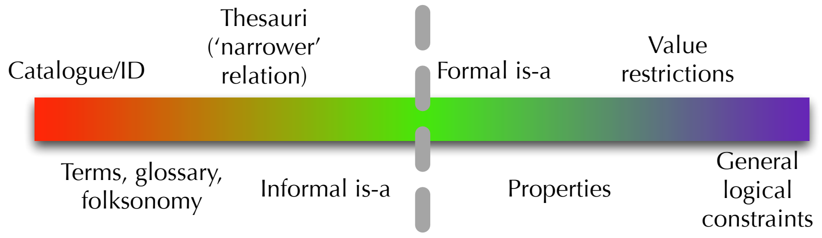
Catalogue/ID
Thesauri ('narrower' relation)
Formal is-a
Value restrictions
Terms, glossary, folksonomy
Informal is-a
Properties
General logical constraints

Figure 2: The semantic spectrum, simplified from [18]

First, at the extreme left, simply listing identifiers for a set of objects is a sort of primitive shared conceptualization.

Next, and marginally more formally, a *controlled vocabulary* is simply a deliberate restriction of the terms we use to describe the world in some context. In the same general category, a *folksonomy* might be defined as a 'collaboratively' or 'loosely' controlled vocabulary, in which, ideally, some consensus emerges on the terms to use to describe some universe of resources. In both cases, though, there is no real structure to (the relationship between) the terms.

A *thesaurus* is a controlled vocabulary representing concepts, plus some declared relationship between the concepts. Most typically, a thesaurus is used for, or at least associated with, information retrieval: you might go into a library and ask to be directed to books about cats, or what we might call the concept of 'Cats', and a librarian might be able to take you to a shelf of useful books. If you're interested in specifically 'domestic cats', they're on a smaller part of the shelf; if you're actually interested in 'mammals' in general, they might be spread over a few shelves. The relationships between these *broader* and *narrower* terms – for this is the most common structural relation in thesauri – is a practical one: any information retrieved by use of a given concept will also be retrieved by use of the 'broader' concept. These are 'is-a' relations, in the sense that every domestic cat *is a* cat, and each cat *is a* mammal; but this is an 'informal is-a', in the sense of Fig. 2, since in the layout of a pet-shop, for example, or the layout of its supplier's catalogue, 'cat collars' may reasonably be a 'narrower' concept with respect to 'cat' without any suggestion that a cat collar is a type of cat, and a library book about cats may be labelled 'cat' (or for example with the Dewey notation 636.8) without anyone or anything being entitled to conclude that the book itself would be improved by the insertion of fish.

The W3C standard for thesauri is the Simple Knowledge Organization System (SKOS),[19] which formalises these broader/narrower relations and a very small set of further ones, in a deliberately simple framework which is designed to be broadly deployable.

Next, and stepping over the dividing line of Fig. 2, the simplest *ontologies* are those where this hierarchy of 'is-a' relations is indeed expected to hold in a logically useful sense, and where labelling something with an ontology term is indeed an assertion that that thing is an instance of the labelled type, and therefore of any supertype. For example, the Linnaean name for cats is *Felis catus*; this is a species of the genus *Felis* in the family *Felidae*, in the order *Carnivora*, all the way up to the kingdom *Animalia*; and this hierarchy is explictly intended to allow me to conclude, on being presented with something labelled *Felis catus*, that that thing is indeed a cat rather than a book about cats, and that the (by now indignantly wriggling) subject is a carnivorous animal, with claws.

This 'formal is-a' is the intended interpretation – that is, the intended 'seman-

[19] `http://www.w3.org/2004/02/skos/intro`

tics' – of the statement in Fig. 1 that the thing named by `http://example.org/a.n.other` is of 'type' `foaf:Person`. The usual word for the type in this context is *class*, and the (FOAF) URI `http://xmlns.com/foaf/0.1/Person` (usually abbreviated to just `foaf:Person`) is the name of the abstract class of persons; individual people are *instances* of this class.

The next step along the semantic spectrum is to be able to declare that instances of some classes may possess *properties* (the term *predicates* is generally interchangeable) such as `foaf:name`. An ontology of this type may declare the domain and range of a predicate, restricting the type of subject or object that the predicate may have. A triple-store can then use this information to make certain deductions, as we illustrated in Sect. 1.6 on p.11. However, there is no way, in this type of simple ontology, to *require* that an instance of a class must have a particular property (I can be a `foaf:Person` without having an email address, for example, bizarrely victorian though that notion may seem), and a reasoner which does not know my email address, because it hasn't been provided with an `foaf:mbox` property for me, is not entitled to conclude that I therefore don't have one.[20]

The simple ontological framework represented by RDF Schemas (RDFS) is capable of expressing no more than this: that classes exist and that some classes are subclasses of others, that resources can be instances (or members) of classes, that properties exist and have classes as domain and range. The payoff is that reasoners[21] which implement this logic can be simpler and faster than a reasoner capable of more intricacy. Referring back to the discussion around Fig. 1, such a reasoner would be capable of concluding that `http://nxg.me.uk/norman/` is a `foaf:Person` (because that resource is asserted to have a `foaf:name`, implying that `/norman/` is in that property's domain), and it could conclude that `http://someone.org/ping` is a `dc:Agent`; but it could not use the identity of the `foaf:mbox` properties to conclude that these are the same person, and it could not detect any inconsistency in a subsequent assertion or discovery that `/norman/` is a `dc:BibliographicResource`.

To get such extra reasoning power, it is necessary to go beyond RDFS to the right-hand side of the ontology spectrum, towards more elaborate frameworks such as OWL. The Web Ontology Language (OWL[22]) comes in several standard varieties – OWL Lite, OWL DL, and OWL Full – and further varieties associated with particular implementations (see [19] and references at `http://www.w3.org/standards/techs/owl`). The difference between these varieties is that some are more *expressive*, in the sense that it is possible to articulate more complicated relationships between resources; the practical tradeoff is that the more expressive variants are harder or more expensive to implement. It is in OWL, and not RDFS, that it is possible to say

```
dc:Agent owl:disjointWith dc:BibliographicResource.
```

A reasoner, once programmed to calculate the logical implications of the `owl:disjointWith` predicate, can thereafter conclude that if a resource such as `http://nxg.me.uk/norman/` has been asserted or discovered to be in both classes, then it has detected a logical inconsistency. This might form part of a longer chain of reasoning, or it may be practically useful for discovering latent errors in a

[20] This 'permission for ignorance' is known as the 'open world assumption'. It is terribly important for the mathematical logic which a reasoner implements, but most non-specialist users of RDF can be forgiven for feeling it to be a rather abstract detail. It is in contrast to the 'closed world assumption' built in to more traditional relational databases: if there is no salary beside my name in the personnel department's database, then the system will conclude that I am a person without a salary, as opposed to a person whose salary is unknown (which is the only conclusion possible in an RDF analogue of this database); in this case, the personnel office is certain they're not going to pay me.

[21] A 'reasoner' in this context is a piece of software which can implement the logical calculus indicated by an ontology.

[22] Yes, the standard is indeed cracking a Winnie-the-Pooh joke here.

database: if a mapping database discovers a feature that is marked as both a river deep and mountain high, then its curators can thereby discover they have some repairing to do.

1.8 So what, briefly, is the Semantic Web?

What we have ended up with is this: in RDF we have a flexible model for representing relatively simple statements about the world, in a primitive subject-predicate-object pattern. This framework has a number of syntaxes which are more or less suitable in particular contexts; these are most prominently RDF/XML, Turtle and RDFa, but there are also some emerging alternatives suitable for JSON. Despite this abundance of syntax, it is the simplicity and very broad applicability of the primitive triples that is key, since almost any reasonably structured data source can be wrangled into RDF form, one way or another, and this makes RDF an excellent mechanism for combining heterogeneous data sources.

An important step in this combining mechanism is the carefully-struck balance between the logical precision of the calculus of resources which ontologies express (given a set of RDF triples, what further triples are logically entailed?), and both the flexible *im*precision of the natural-language link between URIs and their denotations, and the variable precision of ontological designs, which may range from the broad-brush utility of FOAF ('only `Person`s have email addresses') to the intricacy of the Gene Ontology.

Furthermore, the very close technical analogy between the Semantic Web and the Web of Strings reassures us that the properties which made the web successful (its openness, fault-tolerance, and so on, as discussed in Sect. 1.2) will support the Semantic Web, too.

Once information is in RDF form, and stored in a 'triple-store', it can be queried[23] either directly, or with the assistance of a 'reasoner', which synthesises extra triples based on the inferences obtained from the originally asserted triples, in the presence of the extra structure provided by a simple or a complex *ontology*.

We could say more about the 'Semantic Web'. We could talk about some of the details of triple-stores and query engines, but that would very quickly immerse us in a level of technical detail which can verge on the impenetrable, and which is in any case still in active research-level development. We could introduce some of the features of ontology design: though this is now largely stable, it would take several more chapters to cover, without immediate intellectual profit. We could talk about deployment, but many of the main deployments of Semantic Web technology are in back-end systems supporting complicated data-integration projects, and do not make for vivid illustration.

Instead, the goal of this chapter has been to provide an introduction to the core concepts of the 'Semantic Web', at a level and to an extent that will allow the reader to understand the goals and starting-point of a project using these technologies, and to understand the way in which the 'Semantic' Web relates to the Web of Strings of our last two decades' familiar experience. For more details, the W3C's 'Data Activity'[24] provides useful links to further reading; and for some practical details, including a rather more sceptical account of the Semantic Web than I have produced here, see [21].

1.9 Where is the Semantic Web?

It is harder to point to the Semantic Web than to the textual web. Although we encounter the text web whenever we look at Wikipedia, book a flight, or search for cat videos, the semantic version remains in the background.

[23] The standard W3C query language for RDF is called SPARQL [20], which is to a triple-store what SQL is to a relational database.

[24] `http://www.w3.org/2013/data/`

The French National Library (BnF) has a large catalogue, split into multiple heterogeneous parts, using various standards (such as EAD and MARC), and making links to a dozen or so external sites. A Semantic Web approach has made this information available through a homogeneous interface which provides intelligibility and syntactical consistency, without sacrificing the open-endedness which is necessary if such a complicated and long-accumulated information source is to be made fully available [22]. This catalogue is now in use by various applications and smaller libraries, or enhanced and expanded by more specialised catalogues elsewhere. Going from the catalogue to the instance, [23] describes a system which makes available, in machine-readable form, elements of the semantic content of papers in PubMed Central. Reference [22], is in the 'in-use' track of that year's ESCW conference proceedings, where you might find other illustrative examples; there are further illustrations in the collection of open datasets maintained by the Semantic Web Journal.[25]

At the time of writing (2014), it is not yet clear, at least to me, just when the Semantic Web will be manifestly 'working', nor how it will get there; it is not even transparently clear what 'working' would mean. The highly-integrated scenario of [8] is still in the future, not because it is infeasible, but most likely because it is predicated on a good deal of large-scale coordination which is difficult to force on a decentred web. Linked Data (see Sect. 3, below) shows a path into the future, but is not an end-point. But perhaps we are asking too much for a grand design. The evolution from web pages, to blogs, to Twitter is obvious only in retrospect; the fact that (for example) Instagram, Facebook, Twitter, Flickr, and up-to-date conference organisers all use hashtags is explicable, but could never have been predicted; conversely, it is rather surprising that we are still using SMTP-based email 32 years after it was first standardised, and in the face of numerous claims for killer alternatives. Perhaps the most likely future for the Semantic Web matches that of Artificial Intelligence research, even though the former community has always been nervous of the comparison: AI never obviously delivered its grand promises – it never 'worked' – but in a world of self-driving cars, face-recognition, and pocket-sized machine-translation devices, it's clear that it's been quietly engineering its way into successful deployment for decades.

The two final remarks to be made concern, in the first place, Web 2.0, which is sometimes linked to the Semantic Web, but which in fact has almost nothing to do with it; and in the second place the Linked Data paradigm, which is arguably the first appearance of the Semantic Web in more nearly everyday experience.

2 Web 2.0

The term 'Web 2.0' has disorientingly many meanings. To some, it refers to the cluster of mid-level technologies which let services such as Google Maps or Facebook act as an 'application inside a webpage'; to others it has meanings which can be grouped under the slogan 'the read/write web'; another camp sees it as a marketing expression and is divided about whether this is an obviously good or an obviously bad thing; and for yet others it is a proxy for a broad change in society towards a more decentred, or engaged, or empowered, or open-licensed, or otherwise utopian future. These terms are not remotely equivalent: in some cases they may not even intersect, so something as obviously next-generation as Wikipedia is Web 2.0 under some definitions but not others.

The term is really only useful as an adjective, to denote something – anything – that is more than a collection of static web pages. However, even saying that

[25] http://www.semantic-web-journal.net/accepted-datasets

much begs the question: what is it that's so *wrong* with static web pages (which we're now obliged to call Web 1.0)? Indeed, depending on your conception of what 'Web 2.0' means, podcasts are firmly Web 1.0, and if you turn off visitor editing on your wiki pages, or commenting on your blog, then those become Web 1.0, as well.

The term 'Web 2.0' appeared as the title of a conference series run by O'Reilly Media in 2004, and the term spread from there to wider and still vaguer usage. There is no technical difference between 'Web 2.0' and the original version of the web; instead, the term refers to a loose collection of techniques and even attitudes, which became increasingly popular in the first half of that decade, which were heavily oriented towards supporting, encouraging, and exploiting user *interactivity* on the web.

Above, I characterised the web as a space where anyone can join in, but admitted that initially this was more true in principle than in fact. The web of the 90s was, for most, a read-only medium: the web was full of text and images, but the only way to talk back was through email, or a few now-arcane spaces such as Usenet. Though both were effective, and though Usenet is still, in 2014, Not Quite Dead Yet, they are hardly a free-flowing world-wide conversation. In the broadest sense of the term, a blog is a Web 2.0 thing: it's easy to set up for oneself, and if even that is too inconvenient, it's easy to use a blog set up by a service such as Wordpress. With that step taken, it's merely a matter of creative diligence to broadcast to the world, and to dispute, declaim, decry and generally bicker in comment threads. Broadly blog-like things such as Twitter and Facebook become much easier to conceive, as both provider and consumer, once the idea of a blog service plus comments has become part of the culture.

With that step taken, other technologies were able to collide; 'AJAX' was one such. The 'J' stands for Javascript, which originally appeared as a language supporting simple scripting within a web browser – providing an element of dynamism in web pages, so that they could adjust themselves to the browser's environment rather than appearing as a simple block of text sent from a server. This language, and the set of practices which AJAX represents, have grown to the point where a web browser may now be seen as a separate programming ecology. Some web pages, such as a Google Mail page, arrive as a minimal amount of HTML, enclosing a Javascript program which, running entirely within the browser, interrogates a server, and synthesises from scratch the HTML which the browser finally displays. From a protocol point of view, there are minimal differences between the earliest web pages at `http://info.cern.ch/hypertext/WWW/TheProject.html` (with a 'last modified' date of 3 December 1992), and the `twitter.com` homepage which creates the page programmatically and on the fly in the browser: both use HTTP to transport materials from a server, and both offer HTML for the browser to render.[26]

One of the other long-latent possibilities which was explored in the early 2000s was the idea of uploaded and shared content. Flickr (`https://www.flickr.com`) was one of the first broadly-known services to support users' uploading and sharing images. At a technical level (again), this is not much different from blogging, but Flickr and the social-bookmarking site `delicious.com` are interesting for this chapter's purposes because they were among the first widespread services to support *tagging*. Flickr allowed users to add simple labels to pictures, and Delicious allowed users to similarly label webpages; there is presumably some indirect link here with the rapid adoption of Twitter hashtags. These tags – which were intended to support both categorisation and search – were taken from an

[26] Indeed there is something of a paradox here: the web of 2010 and the web of 2000 look hugely different, to the extent that they seem to come from different worlds. But these differences are to do with new social possibilities, and with the browser's new and almost entirely unpredicted programming ecology, rather than arising from any change in the underlying nuts and bolts; and the closer we look at Web 2.0, the more out of focus it seems to become.

unrestricted vocabulary but with the expectation that users would spontaneously cooperate to choose broadly compatible tags, in a lightweight shared semantic environment which was soon labelled a 'folksonomy' (as we briefly mentioned above). However this is as far towards the Semantic Web as Web 2.0 has ever moved, and so, despite the suggestion of the 'Web 2.0' name and some of the accompanying rhetoric, Web 2.0 has rather little to do with the Semantic Web. There's more to explore in the programming ecology, but in web terms, Web 2.0 is rather a dead end.

3 Linked Data

Even acknowledging that the Semantic Web will probably remain somewhat obscure, one can ask how broadly it is deployed, or to what extent its deployments are reachable. As we discussed above, the Semantic Web is still largely used in 'server-side' applications, rather than becoming part of the commerce of the open web, as was originally anticipated. Part of the reason for this is that the Semantic Web technology stack is challenging to use: the tools are of varying maturity, and use a broad range of underlying technologies, so that deployments are still to a large extent bespoke. Even when simply making data available, rather than attempting to process it, making data web-ready may require a quite profound engagement with the Semantic and textual webs' conceptual foundations; many such deployments are still, even now, worthy of an academic paper describing the experience. Few non-specialist projects can make such investments.

An alternative approach is to use the *Linked Data* paradigm.[27] At heart, the linked data paradigm *is* the Semantic Web, but with a practical focus on the irreducible minimum required to get data onto the web in a way which is compatible with the goals and practices described in Sect. 1.

Quoting [24], the linked data principles are:

1. "Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).
4. Include links to other URIs. so that they can discover more things."

Notably, these principles have more to say about HTTP and URIs – that is, about the *mechanics* of retrieving this information, and by extension the mechanics of making it available – than they say about the semantics of the information being distributed.

Together, these say (in effect) that the 'linked data web' is just like the web of strings, familiar to us all, except that the raw materials are not HTML files, but files in one or other RDF syntax (so RDF/XML, Turtle, or RDFa). All four points are essentially adaptations of the design principles of Sect. 1.2 and best practices such as [25], which were, as it turned out, so very effective in making the web (of strings) so very successful. The third point states that looking up a URI should provide useful (RDF) information *to a machine*, but is also taken to suggest that putting a resource's URI into a web browser should provide human-readable (HTML) information as well – this provides one bridge between the two webs.

[27] The word 'paradigm' is probably the best term in general, to the extent that it represents the result of accepting the insight of a set of rather high-level principles. However making these principles concrete is intricate enough that 'Linked Data Best Practices' sometimes seems more descriptive; and when defending the principles against other approaches, the word 'dogma' springs quickly to mind.

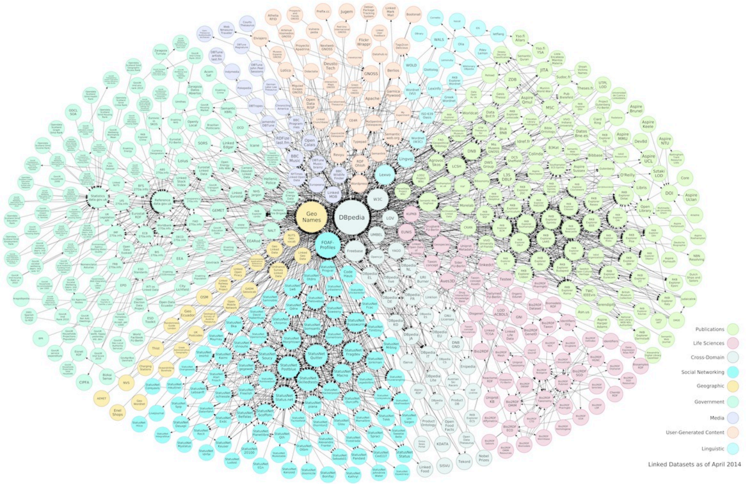

Figure 3: The Linking Open Data cloud diagram, as of April 2014. Although the text and most of the interconnections are too small to make out at this scale, the two largest central blobs – which act as rich sources of consensus names for objects in the world – represent DBPedia and GeoNames. The green-coloured clusters represent government (left) and bibliographic data (right); other clusters represent providers of life sciences, geographic, social networking, media, user-generated, and linguistic data. For details and credits, see `http://lod-cloud.net`.

Although the traditional web is generally thought of as delivering human-readable pages, it has from the very beginning been used for delivering machine-readable content, such as data files for analysis or PDF files for printing. This is not the machine-readability we mean in this context. Just as a human might read a Wikipedia article, internalise the results, and click on a link for more information, a linked data client – typically a component of a larger application – is expected to process the semantic content of the data it finds at one URI, and then follow further links for further information. Given the RDF of Fig. 1, for example, a linked data client might store the information listed, and then follow the link `http://someone.org/ping` to see if there is any further information available there. It is this final step, which lets the machine 'follow its nose' that represents the key insight of the linked data paradigm, and which is the machine analogue of 'wasting hours on Wikipedia'.

The network of links between 'linked-data'-style data sources is illustrated in Fig. 3. A large fraction of the data in this cloud is in the form of SKOS vocabularies, or uses the simple FOAF or DC ontologies; but while these are useful for coordination between data sources, there is no restriction on the ontologies which are used in fact.

The archetypal Linked Data (client-side) application is a simple application – perhaps even running entirely within a browser – which reads the contents of a URI, interprets and perhaps displays what predicates it recognises (which will probably include at least FOAF), and lets a user move on to further related information. The archetypal Semantic Web application (if such a thing exists) might use SPARQL to query a large and semantically rich database, possibly requiring a fair amount of inferencing on the part of the service and within itself, and requiring significant effort on the part of its author, to understand the ontology or ontologies in which the information is encoded. Though these will be visibly different applications, and one is a lot easier to implement than the other, it is hard to articulate a fundamental difference between them.

There are further compact details in [26], a book-length discussion with a practical focus in [27], and pointers to current information at `http://www.w3.org/standards/semanticweb/data`.

4 Conclusion

One could derive the impression, from the long description of Sect. 1, that the Semantic Web is complicated and arcane. It can be, but the rude health of the linked data cloud shows that that complication is in the background of a simple and robust means of distributing machine-intelligible information. While we may not be quite at the stage of letting our computers run amok with our credit card, or organise our lives as the vision of [8] imagines, we have now confidently, if still rather discreetly, stepped into a web of data.

Acknowledgements

I am most grateful to Susan Stuart for exacting and detailed comments on the drafts of this chapter, to members of the `semantic-web@w3.org` list for 'in-use' references, and to Chris Bizer for making available an early copy of the 2014 Linked Data cloud.

Glossary

AI Artificial Intelligence.

DC Dublin Core: a set of core metadata terms developed by and for the library community; see `http://dublincore.org/`.

FOAF Friend of a Friend: a vocabulary for attributes of, and relationships between, people; see `http://www.foaf-project.org/`.

HTML Hypertext Markup Language; see `http://www.w3.org/html/wg/`.

HTTP Hypertext Transport Protocol: the underlying language of the web; see [1].

RDFS RDF Schemas: lightweight ontologies for RDF; see [12].

RDF Resource Description Framework: see [10].

URI Uniform Resource Identifier: the form of a resource name (almost the same as a URL); see [2].

W3C World Wide Web Consortium: the organisation which develops and standardises protocols for the web; see `http://www.w3.org`.

triple The triple of subject, predicate and object which is at the heart of the RDF model.

triple-store A database designed to store RDF triples, rather than the tabular data of a more common 'relational' database.

References

[1] Roy T Fielding, J Gettys, J Mogul, H Frystyk, Larry Masinter, P Leach, and Tim Berners-Lee. Hypertext transfer protocol – HTTP/1.1. RFC 2616, June 1999. URL: `http://www.ietf.org/rfc/rfc2616.txt`.

[2] Tim Berners-Lee, Roy Thomas Fielding, and Larry Masinter. Uniform resource identifier (URI): Generic syntax. RFC3986, January 2005. URL: `http://www.ietf.org/rfc/rfc3986.txt`.

[3] Tim Berners-Lee and Robert Cailliau. WorldWideWeb: Proposal for a hypertext project. Webpage, CERN, November 1990. URL: `http://www.w3.org/Proposal.html`.

[4] Tim Berners-Lee. Information management: A proposal. Technical report, CERN, March 1989. URL: `http://info.cern.ch/Proposal.html`.

[5] Tim Bray, Jean Paoli, and C M Sperberg-McQueen. Extensible markup language (XML) 1.0. W3C Recommendation, February 1998. First edition of `http://www.w3.org/TR/REC-xml/`. URL: `http://www.w3.org/TR/1998/REC-xml-19980210`.

[6] Ora Lassila and Ralph R Swick. Resource description framework (RDF) model and syntax specification. W3C Recommendation, February 1999. URL: `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/`.

[7] Vannevar Bush. As we may think. *The Atlantic Monthly*, July 1945. URL: `http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/`.

[8] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001. `doi:10.1038/scientificamerican0501-34`.

[9] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia – a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 2014. To appear. `doi:10.3233/SW-140134`.

[10] RDF 1.1 concepts and abstract syntax. W3C Recommendation, February 2014. URL: `http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/`.

[11] RDF 1.1 Turtle. W3C Recommendation, February 2014. URL: `http://www.w3.org/TR/2014/REC-turtle-20140225/`.

[12] RDF Schema 1.1. W3C Recommendation, February 2014. URL: `http://www.w3.org/TR/2014/REC-rdf-schema-20140225/`.

[13] RDF 1.1 XML syntax. W3C Recommendation, February 2014. URL: `http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/`.

[14] RDF 1.1 semantics. W3C Recommendation, 2014. URL: `http://www.w3.org/TR/rdf11-mt/`.

[15] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000. `doi:10.1038/75556`.

[16] Michael Bada, Robert Stevens, Carole Goble, Yolanda Gil, Michael Ashburner, Judith A. Blake, J. Michael Cherry, Midori Harris, and Suzanna Lewis. A short study on the success of the gene ontology. *Journal of Web Semantics*, 1(2), 2004. URL: `http://www.websemanticsjournal.org/ps/pub/2004-9`.

[17] Thomas R Gruber. A translation approach to portable ontology specification. *Knowledge Acquisition*, 5(2):199–220, 1993. `doi:10.1006/knac.1993.1008`.

[18] Deborah L. McGuinness. Ontologies come of age. In Dieter Fensel, Jim Hendler, Henry Lieberman, and Wolfgang Wahlster, editors, *Spinning the*

Semantic Web: Bringing the World Wide Web to Its Full Potential. MIT Press, 2003. URL: http://www-ksl.stanford.edu/people/dlm/papers/ontologies-come-of-age-mit-press-(with-citation).htm.

[19] W3C OWL Working Group. OWL 2 web ontology language document overview (second edition). W3C Recommendation, December 2012. URL: http://www.w3.org/TR/owl2-overview/.

[20] The W3C SPARQL Working Group. SPARQL 1.1 overview. W3C Recommendation, March 2013. URL: http://www.w3.org/TR/sparql11-overview/.

[21] Aaron Swartz. *A Programmable Web: An Unfinished Work*. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool, 2013. doi:10.2200/S00481ED1V01Y201302WBE005.

[22] Agnès Simon, Romain Wenz, Vincent Michel, and Adrien Di Mascio. Publishing bibliographic records on the web of data: Opportunities for the BnF (French National Library). In Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *The Semantic Web: Semantics and Big Data*, volume 7882 of *Lecture Notes in Computer Science*, pages 563–577. Springer Berlin Heidelberg, 2013. 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. URL: http://2013.eswc-conferences.org/program/accepted-papers, doi:10.1007/978-3-642-38288-8_38.

[23] L Garcia Castro, C McLaughlin, and A Garcia. Biotea: Rdfizing pubmed central in support for the paper as an interface to the web of data. *Journal of Biomedical Semantics*, 4(Suppl 1):S5, 2013. URL: http://www.jbiomedsem.com/content/4/S1/S5, doi:10.1186/2041-1480-4-S1-S5.

[24] Tim Berners-Lee. Linked data. Webpage, 2006. URL: http://www.w3.org/DesignIssues/LinkedData.html.

[25] Leo Sauermann, Richard Cyganiak, Danny Ayers, and Max Völkel. Cool URIs for the semantic web. W3C Interest Group Note, December 2008. URL: http://www.w3.org/TR/2008/NOTE-cooluris-20081203/.

[26] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data – the story so far. *International Journal On Semantic Web and Information Systems*, 5(3):1–22, 2009. URL: http://tomheath.com/papers/bizer-heath-berners-lee-ijswis-linked-data.pdf, doi:10.4018/jswis.2009081901.

[27] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool, 2011. URL: http://linkeddatabook.com, doi:10.2200/S00334ED1V01Y201102WBE001.