Nodalities

THE MAGAZINE OF THE SEMANTIC WEB

talis®

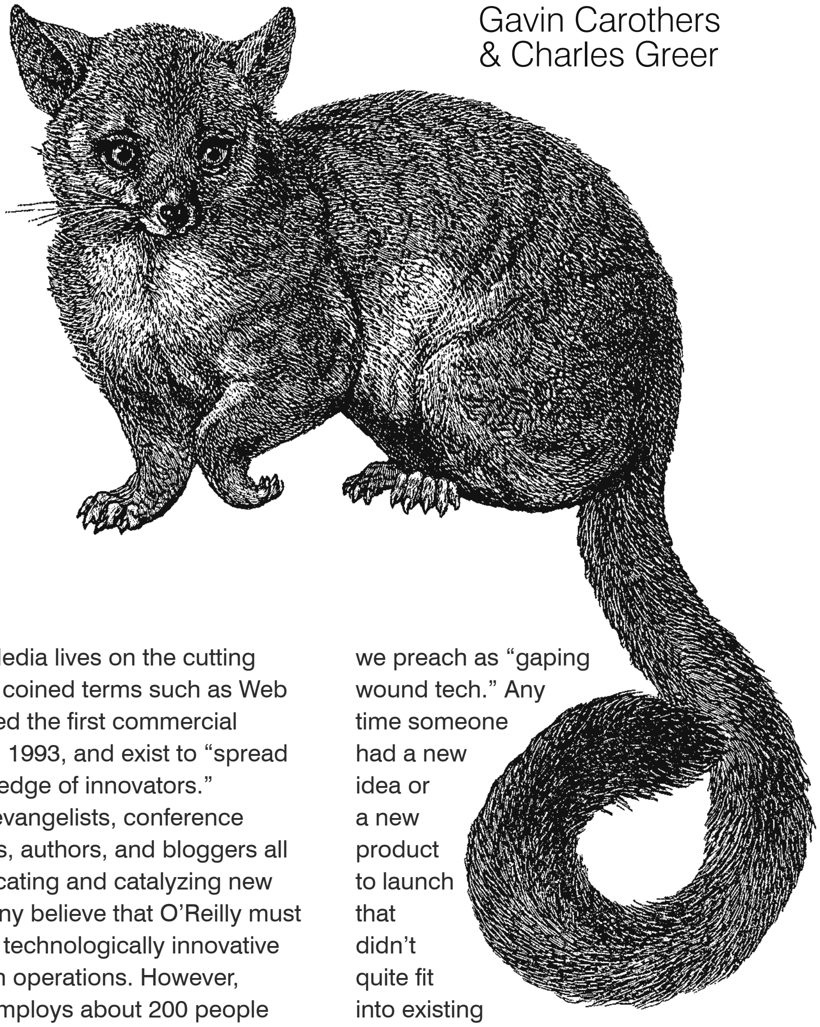
www.talis.com/nodalities

March / April 2009

INSIDE

Linking Data and Semantics at OReilly

Gavin Carothers
& Charles Greer



O'Reilly Media lives on the cutting edge. We coined terms such as Web 2.0, created the first commercial website in 1993, and exist to "spread the knowledge of innovators." With our evangelists, conference presenters, authors, and bloggers all communicating and catalyzing new ideas, many believe that O'Reilly must be just as technologically innovative in our own operations. However, O'Reilly employs about 200 people but only half a dozen developers, so naturally ideas are thrown at our developers faster than it is possible to implement them. We've been known to refer to this tension between our public position on the cutting edge and internal expectation to live up to what we preach as "gaping wound tech." Any time someone had a new idea or a new product to launch that didn't quite fit into existing systems, we found some way to shoehorn it in, with a quick Perl script or some clever custom SQL. As we did this, more and more of our work became preventing our systems from collapsing under the weight of those one-off ETLs and scripts. The cost

continued on page 3

SUBSCRIBE NOW

Nodalities Magazine is made available, free of charge, in print and online. If you wish to subscribe, please visit www.talis.com/nodalities or email nodalities-magazine@talis.com.

MEET US AT

Talis staff will be presenting and attending several events over the next few months, including;

WWW2009
Madrid, Spain | 20-24 April 2009

ESWC2009
Heraklion, Greece | 31 May - 4 June 2009

For further information visit: www.talis.com/platform/events

Nodalities Magazine is a Talis Publication

ISSN 1757-2592

Talis Group Limited
Knights Court, Solihull Parkway
Birmingham Business Park B37 7YB
United Kingdom
Telephone: +44 (0)870 400 5000

www.talis.com/platform
nodalities-magazine@talis.com

The views expressed in this magazine are those of the contributors for which Talis Group Limited accepts no responsibility. Readers should take appropriate professional advice before acting on any issue raised.

EDITOR'S NOTES

Thank you for picking up your copy of Nodalities Magazine. This is our 6th edition, and we're very pleased to continue sharing Semantic Web stories from around the world. For over a year, now, we've been publishing articles from innovators and explorers from an emergent industry. However, the tone has been subtly changing. Instead of hearing about what we could and can do, we're listening to people tell us what they've been building and are currently launching. This is a truly exciting time to be writing in this field! Through telling these stories, we're beginning to see the emergence of a much bigger narrative: that of the Semantic Web as a whole.

That's not to downplay innovation in any way. Indeed we are seeing innovation being taken very seriously. In this issue we discuss SemaPlorer, winner of the 2008 Billion Triples Challenge. SemaPlorer attempts to bring human context to the vast quantities of semantic data on the web. Benjamin Nowack also explores the concepts behind Paggr, his personal (linked data) aggregator. Elsewhere, Alex Tucker explains how his contract with the defence industry lead to innovation in the area of DNS and exposing SPARQL endpoints, serendipitously making use of Apple's Bonjour protocol.

Talis' Tom Heath also reviews a book with a title only a Semantic Web Geek could love (but assures me the book is excellent!) and Leigh Dodds puts forth his notion of data being like rivers, streams and pools—depending on how we make use of them.

Finally, web giant O'Reilly Media have opened their OPMI to the rest of the web, exposing large amounts of linked data to the web. They tell the story of moving from a hacked-together system for organising their books, to a working RDF model of semantic metadata.

If you would like to subscribe for a print copy of Nodalities Magazine (free of charge), please visit the site at www.talis.com/nodalities. I also welcome article ideas and comments. Please send any requests to nodalities@talis.com.

Zach Beauvais

Continued from front page.

of simply keeping track of which scripts were using what bit of transformed data and where that data came from had became so high as to become unsustainable. We'd accrued so much design debt that only the most radical of approaches could save us from being crushed by the weight of our inherited code.

Of course, we didn't really know that at the time. Today we have a Linked Data, Semantic, RESTful, URI-based, highly buzz-wordy solution mostly by accident and through ruthless pragmatism. Instead of embracing the ideas of the Semantic Web at the outset, we arrived at the Semantic Web because it was the only solution. We thought we were traveling down two completely unrelated roads. We started down the first while trying to replace a Java Bean Shell script that copied book content to a few different places. The other road began when we wanted to know what color to make the border of a PDF. The first would lead to an Atom Publishing Protocol server and clients, the second to our modeling all product metadata in RDF and opening that to the public.

As it turns out, the two roads weren't so unrelated after all. RDF is designed to handle modeling information in a distributed manner and provides the underpinnings for the actual metadata we store, aggregate, and use. AtomPub's RESTful interface is ideally designed for managing individual chunks of all this distributed data over time and provides programs and people a simple, standard interface for publishing, accessing, and updating it. As we progressed down each path, we were making (often unknowingly) major progress in generating linked data and semantics, the two pillars of the Semantic Web.

The RESTful Road

In 2005, soon after O'Reilly launched a custom book publishing platform, we discovered that we'd deferred a hard question. Although we had a good way to represent books elecronically, Docbook, we didn't know how to make sure that we could easily add new books as they came down the production pipeline, and how to incorporate errata so as to maintain the best possible information in the system at all times. In response to this problem we looked to our own books and found that REST provides the tools to model and maintain ideas like “a canonical document” and “synchronization over time.”

Before our adoption of REST as a pervasive strategy, our DocBook files were scattered across many filesystems. On an as-needed basis we transformed the DocBook with one-off scripts and moved them around with FTP from one filesystem to another. We simply didn't have a way of addressing fundamental questions like “Where is the latest, cleanest copy of a book's markup?” Tracking down the best representation of a book's content was a laborious, error-prone task.

Before our adoption of REST as a pervasive strategy, our DocBook files were scattered across many filesystems.

Around the same time we ran into this, we noticed Tim Bray's superb presentations about the then-draft Atom Publication Protocol (AtomPub). This protocol builds on REST and Atom to define a very straightforward method to create and maintain content over HTTP. It gaves us the means to store an atomic chunk of data, assign it a URI and access and update it through a standard interface. Each book in an AtomPub store (or our derivative Atom-based webapps) is modeled as an <atom:entry/>. This atom:entry is the “latest, cleanest copy of a book's markup” and its URI is the canonical location for this content. We decorate this entry with many views of a document, so that anybody can figure out how to get, for example,

- A book's ”source code“, the DocBook markup
- The print book, as an ISBN
- The table of contents (as HTML or as a set of links to other URIs)
- A HTML, PDF or other representation generated from the source
- Whatever Tim O'Reilly or the business folks asked for next

We took this idea of our corpus of content, and turned it into a fast, highly-available way to get N views of our content over HTTP. No more painful ETL and cleanup for any new systems we need to provide. However, now that systems could reliably find and update our content using URIs, another pain point reared its head — we still had no equivalent method to use product metadata across the business! That's where RDF came to the rescue.

No more painful ETL and cleanup for any new systems we need to provide.

rdf:isNeat

“Can our PDFs have the same branding and colors as the printed books?”
—Marketing Person

“Sure! How hard can it be?”
—Innocent Developer

At this point in our journey we have more than 900 titles in the AtomPub repository and addressable by URI. We've (unknowingly) hit a significant Linked Data milestone and everything is progressing well. Dynamically creating a PDF from these entries is as easy as running our DocBook-XSL customization for the correct series to produce XSL-FO and then rendering that XSL-FO into PDF. The only problem was discovering which series (In a Nutshell, Animal Guide, Missing Manual) the content fell under. At that point all progress stopped.

Our definitive source of book and product information is the Product Database (67,000+ lines of Perl, C++, SQL, and a dozen other languages). The database and web application has its own home-rolled "XML Format," as I'm sure many other companies have had. Based directly on the column names from the SQL database, our Book XML was a quick and very dirty way of getting our centralized relational data out into the world as XML. A host of new client applications grew around this new access to product data, but we quickly saw the problems of reusing an adhoc, undefined, schema-generated format. The XML service was also incredibly slow.

Each client kept mappings of these magic values in order to make the data understandable. Those mappings broke, of course, whenever new product types and imprints were added. Even more dangerously, because the semantics of the XML were totally unspecified, element names were opaque and sometimes actively misleading. We might have redesigned the format to include more data and added more and more fields to it but this wasn't an explicitly designed schema, just something generated from the SQL. On the road to exposing this data more cleanly we tried everything. Remodeling the SQL to be more relational didn't offer much benefit and we still couldn't tell what the column names meant. Sitting down and trying to write up a data dictionary was a great exercise, but it became out of date almost immediately. We experimented with JSON-based CouchDB prototypes, but those had the same issue as the SQL with missing meaning. Our Subversion repository is littered with Relax-NG, XML Schema, and Schematron documents to create new XML-based format. Somehow they never got finished as we discovered we either had to define everything or try to design for extensibility. We knew we didn't have the time to create our own Book Metadata Standard. We wanted defined semantics.

```
<IPFamily>

  <Book>
    <product_id>5549</product_id>
    <parent_product_id>6380</parent_product_id>
    <imprint_id>1</imprint_id>
    <product_status_id>5</product_status_id>
    <product_type_id>10</product_type_id>
    <isbn>0596515618</isbn>
    <isbn13>9780596515614</isbn13>

    <final_date>2003-07-02</final_date> <!-- Actually the day the last QC
phase ended -->
    ...
```

As you can see from the snippet above, clients had to deal with knowing exactly what imprint 1 (O'Reilly Media, Inc.) and product type 10 (PDF) meant.

There is at least one obvious XML vocabulary for a publisher looking to capture book metadata: ONIX. Unfortunately, the ONIX standard is archaic, with obscure element names like b004 (ISBN) and g343 (PrizeJury, obviously) (Footnote: Yes, these are the short versions and a longer set of names is also allowed. However, many of the most important vendors only support the short versions.) We did consider ONIX for a time, but then we noticed that every vendor we sent ONIX to treated the fields a bit differently. Even with pages and pages of specification there wasn't any agreement on what elements were important or what they meant. Using ONIX as a format would not solve our semantic deficiency, we still wouldn't know what the "columns" meant.

> *We did consider ONIX for a time, but then we noticed that every vendor we sent ONIX to treated the fields a bit differently.*

In the process of trying to create an XML format we asked a number of people in the company how to find the Publication Date for a book. The answer was surprisingly complex. The value was computed independently by each of the ETL hydras, with subtly different implementations that had evolved with particular client needs. O'Reilly isn't a huge company with layer upon layer of bureaucracy; most questions can be quickly answered with a chat at a desk or an email to the other coast. Imagine our surprise, then, at the results of the Publication Date poll. Most people were confident that one of five dates was the right date, but disagreed on which of the five it was. Retail Availability Date, Actual In Stock Date, Estimated In Stock Date, etc each had its backers. What was really going on was that we discovered the subtle different needs that each business unit had. The strategy we could most easily support? Concensus on a public standard. As we've learned so many times, we needed to go outside the company to find the correct solution. Public standards, specifications, and ontologies could save us from ourselves.

Enter: Dublin Core. We couldn't define our own format or use the industry standard (ONIX), nor could we agree on what a publication date was. Our only choice was go borrow/steal some other group's ideas. It turns out that our problems had already been solved by the library community. The Dublin Core Metadata Initiative created standards, guidelines, and examples for storing and sharing basic, essential metadata. We had a way out, here was a group of people who'd already done a great deal of thinking for us.

Of course, they hadn't done all our thinking for us. Mapping all of our old data into well-designed and well-documented Dublin Core, MARC Relators, FOAF, or any other ontology was going to be hard. So we didn't do it. Instead we mapped the whole of our old, horrible, ugly mess into an undefined ontology called the "Product Database Legacy Ontology." We then moved some of the more obvious items like title and author into Dublin Core and waited. Only once we had a proven need for a new data point in real application would we go though the process of researching, defining, cleaning, and moving it into a modern, public ontology. For those following along closely: no, trim color isn't yet in the public or internal metadata. As it turns out, no one really wanted it. At least, not yet.

All Together Now

Since Gavin's first frenzied port of product metadata to an RDF model, we've been able to negotiate changing requirements, establish data validation and control rules, and bring on new applications with little time spent on data modeling. In other words, meeting our immediate need of a centralized, validated data store of high agility and performance has paid off several times over in deploying new software systems for the rapidly changing company.

One example of the intertwining of Linked Data and Semantics is our Electronic Media distribution system, which lets customers download ebooks, pdfs, videos and the like. Book descriptions, titles, authors names, cover images even the help text provided on the Electronic Media page is simply linked data, built from RDF relationships. When we want to change the help text or a category label, we change it in one document, and everything else in the RDF graph referencing it changes with in moments as well. Just following links pays off.

Previously, the buttons that let a customer add a book to our shoping cart were generated by a system that used nightly ETLs nicknamed "the sync". So new products would have to be prepped for release the night before. We gave special care to their timely appearance in the morning. Alas, they frequently did not appear as hoped, as the ETLs that made up "the sync" had to run in a very precise nightly schedule or we had to take manual corrective action. Now, a reasonably simple HTML template bound to the RDF for a book generates "Buy Buttons" in near realtime without an ETL in sight.

The greatest challenge of updating our legacy IT infrastructure hasn't been replacing the ETLs or synchronization. It's been achieving consensus on the meaning of data elements.

The greatest challenge of updating our legacy IT infrastructure hasn't been replacing the ETLs or synchronization. It's been achieving consensus on the meaning of data elements. In the past, data maintainers might adjust the title of a book to change how retailers present it. Then our website's title would change (the next day), and we would have to bring resources to bear on reconciling the meaning of "title." By using <dc:title/> for our title element, we've established what to expect from those who change the value. It's simpler to make sure people enter particular kinds of data, and then ask for help to extend or change requirements for downstream apps. The publicly available ontologies, we hope, will help everyone communicate more effectively about business needs and shared data points. So far the results are encouraging.

In the Public Eye

Having built several of our own applications using our new RDF metadata and our initial linked data APIs, we thought it might be a good idea to let someone else have a crack at it too and see what they made of it. It took us two weeks to develop the O'Reilly Product Metadata Interface, a simple layer on top of the Deli. A caching proxy preserves the reliability needed by our own applications, while a predicate filter prevents private information from leaking to the public. A bit more about how you can access it can be found at http://labs.oreilly.com/opmi.html or you can just dive right in by giving it an ISBN, IE: http://opmi.oreilly.com/product/9780596529260.

Sharing our work with the public forced us to be much more deliberate and rigorous about our data, but also exposed some simple blunders. On the day we launched the service we waited for the praise to come in and finally saw a tweet! Someone is using... Oh wait:

"OPMI's book identifiers aren't resolvable. Sigh." —Jeni Tennison

"Of course they're resolvable," we thought. "You just have to parse the URN and understand how to pass the URN to... oh, yeah good point." In the process of implementation, we'd forgotten Tim Berners-Lee's second rule of Linked Data:

2. Use HTTP URIs so that people can look up those names.

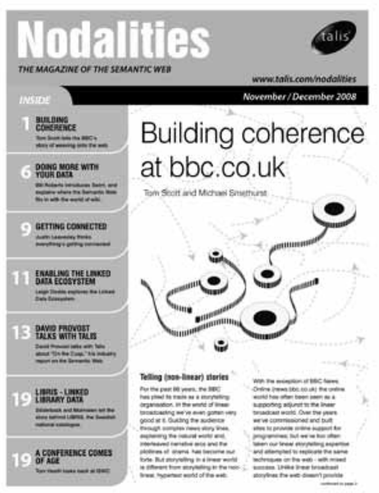
At the start of the process we'd talked about about using some sort of identifier for our products. But that conversation had taken place before we really had all the RDF and Linked Data applications working, so at the time there wasn't any point nor could anyone see the need for a resolvable identifier. Within a few hours of making the data public, the need became blindingly apparent. Part of embracing "anyone can say anything about anything" is that anyone needs to be able to find the anything they want to talk about. And when you've got a statement to make, it's remarkably handy to be able to quickly find out what else has been said. "I loved urn:x-domain:oreilly.com:product:9780596529260.BOOK" is a bit hard to figure out. "I hated http://purl.oreilly.com/product/9780596529260.BOOK" is a lot better.

Lessons so Far

The combination of our RDF modeling and RESTful web services created a simple, cost-effective platform for O'Reilly's product offerings and digital content. While we began our efforts around REST, and then the semantic web, as a systems integration strategy, it has the promise of significantly impacting how publishing approaches information transfer. Our initial ROI came when we started to deploy consolidated business logic for a distributed web environment, it has been our public offerings that have brought us constructive criticism and excitement about showing how O'Reilly has taken the lead in new publishing paradigmata (!)

Gavin Carothers is a Software Engineer and Publishing Technologist at O'Reilly Media.

Charles Greer dabbles in semantics and manages the software development team at O'Reilly Media.



Nodalities
THE MAGAZINE OF THE SEMANTIC WEB
talis
www.talis.com/nodalities
INSIDE
November / December 2008
1 BUILDING COHERENCE
6 DOING MORE WITH YOUR DATA
9 GETTING CONNECTED
11 ENABLING THE LINKED DATA ECOSYSTEM
13 DAVID PROVOST TALKS WITH TALIS
19 LIBRIS - LINKED LIBRARY DATA
19 A CONFERENCE COMES OF AGE
Building coherence at bbc.co.uk
Tom Scott and Michael Smethurst
Telling (non-linear) stories

Nodalities
THE MAGAZINE OF THE SEMANTIC WEB
talis
www.talis.com/nodalities
INSIDE
September / October 2008
1 ANATOMY OF A SEARCHMONKEY
8 TWINE
10 GENERATION ZERO
13 A CALL TO ARMS
16 DISCOVERY AND USAGE ON THE WEB OF DATA
22 FINDING ANSWERS ON THE SEMANTIC WEB
Anatomy of a SearchMonkey

Discovering SPARQL

Confessions of a Semantic Web Junkie: Alex Tucker on using Bonjour and SPARQL endpoints

By Alex Tucker

Confessions of a Semantic Web junkie

Let me start with a confession: I've been banging on about the semantic web to anyone who will listen to me for the past nine years. For some apparently deep seated reason, which some have even labelled perverse, I've kept at it despite endless meetings, misunderstandings, off-handed dismissals and blatant refusals to accept what seems to me to be obvious. It's been a long and sometimes despairing nine years, but looking around now at the state of all things semantic web, one can't fail to realize that it's finally taking off and that companies like Talis are fully committed to its success.

During much of this time I've been working for various defence organisations showing how using semantic web standards and tools can help solve issues of interoperability — getting different systems to talk to one another. That the semantic web, and more specifically the semantic web ontology language OWL, can address these kinds of issues shouldn't be too surprising, given that the US Defense Advanced Research Projects Agency (DARPA) helped fund the process which eventually led to OWL, precisely to solve issues of semantic interoperability.

More recently, I've had the pleasure of working with various teams and projects within NATO who have embraced the semantic web ideas and expanded on them in novel ways. A fundamental shift in attitude in defence, which has in part been driven by the 9/11 Commission's findings, is that intelligence agencies must move from a 'need to know' to a 'need to share' mentality. For me, this shift has echoes in the 'linked open data' ideals, and for someone who also bangs on about open source, open standards and all things open, it's a step in the right direction. Obviously, sharing information in a military context is a little different to the ideals of linked open data, but the fundamental issues around getting the information out of existing silos and usable by other systems are much the same.

A recent success in one such NATO project has been the decision to move from using a bespoke query protocol for RDF, a sort of 'query by example', to using the World Wide Web Consortium's new semantic web query language, SPARQL. This might seem obvious, but a few years ago when the protocol was being developed, SPARQL didn't exist — therein lies another discussion on how organisations need to be more agile to be able to cope with the length of a 'Web Year'. As far as the project group are concerned however, this is great news, as they don't have to come up with a 'Standardization Agreement' or STANAG to define their own query protocol and can instead just point to and use an existing standard, SPARQL.

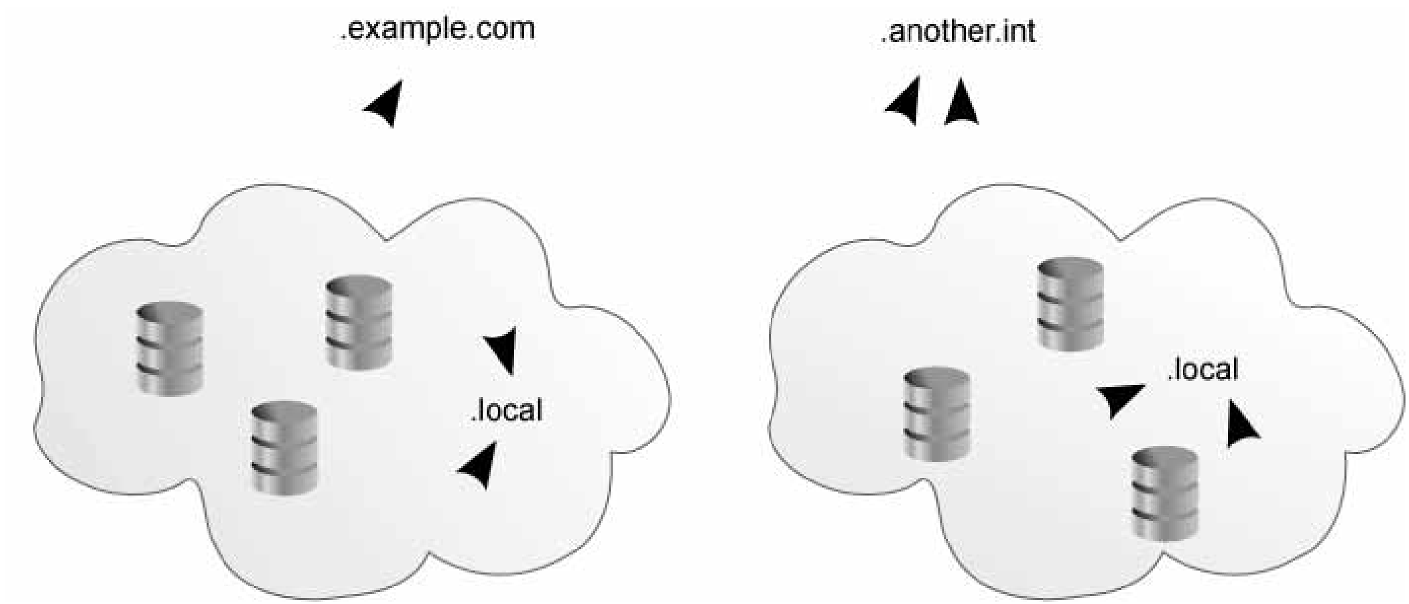
Bonjour

Another standard used by this project is Bonjour (formerly Zercoconfig or Rendezvous), Apple's open standard for no-nonsense network configuration and service discovery — the technology behind iTunes' ability to discover and play music from other iTunes applications on different computers. The great thing about using Bonjour for this project is that it's decentralized: there's no need to set up the registry of information providers beforehand. Instead, each provider is free to publish their existence and their capabilities, both on a local and wide area network, without setting up any new infrastructure.

Each service in a local network can publish its details in any number of domains. The clouds above represent local networks, the arrows represent the act of publishing, or registering service details to a domain, the cylinders represent services, and the '.local' is the domain for the local network. iTunes, for instance, currently publishes itself only to the local area network (which is different to the initial iTunes release where people realised they could share their music with friends over the internet by advertising their iTunes database to their own domain name, how terrible). The domains themselves correspond to the normal domain name system (DNS) names we're used to, since DNS is really at the heart of the Bonjour protocol.

> *Each service in a local network can publish its details in any number of domains.*

Services can then be discovered simply by looking for a particular service type in a particular domain; a typical Bonjour service discovery query would ask for all printer services in the local domain. These service types are given slightly cryptic names (iTunes, for example, uses _daap._tcp), but are all listed on-line at dns-sd.org/ServiceTypes.html for everyone to see and re-use.



Hello SPARQL

What we've done then is create a service type for SPARQL, allowing information providers to publish their SPARQL 'endpoints' to be discovered by information consumers. Technically, the SPARQL service type is _sparql._tcp and is listed at dns-sd.org/ServiceTypes.html along with a short description of the properties which should be published.

In keeping with the RDF mantra that "anyone can say anything about anything", we've tried to ensure that anyone can publish a description of not only their own SPARQL services, but also those of others. A published SPARQL service record has two properties, one called 'path' and one called 'metadata', from which a client derives two URLs, the former pointing to the SPARQL service endpoint, and the latter pointing to some arbitrary (RDF/XML encoded) metadata it can fetch. The normal approach would be to just use simple paths (e.g. /sparql and /service-metadata.rdf) for the values of these properties, which would be interpreted as pointing to the discovered host (e.g. http://dbpedia.org/sparql and http://dbpedia.org/service-metadata.rdf). However, by using a full URL for the value of either property, a service record can be published which points to a service or some metadata on a different host entirely, allowing us to form what are essentially proxy records for existing SPARQL services which don't (yet) use Bonjour service discovery.

The Bonjour specifications are firmly geared towards making the user's life simpler, so for instance while the name of a service is really just a normal DNS name, the specifications insist that it should be a human readable name with proper capitalization, spaces etc., although no dots are allowed. The specifications also give guidance as to the sorts of properties and values a service record should contain, and are keen that service records be as concise as possible and leave much of the nitty-gritty details of discovering service capability to the main protocol, in our case SPARQL.

Into the voID

However, the SPARQL protocol doesn't have much to say about what to expect of an endpoint other than, *here it is.* Our approach to this has been to allow a little bit of extra information in the service record, for instance using the 'vocabs' property to declare the URIs of any vocabularies the service uses. To find out more information about a service, a consumer can use the value of the 'metadata' property to fetch a fuller service description. The only restriction is that this service description should be marked up in RDF/XML — what better way to encode metadata?

As to what this service description metadata should contain, well that's currently left to the provider. In an ideal world, using RDF and OWL to describe a service would mean that it is completely self-describing and that a consumer could fetch the document, fetch any referred ontologies, and figure out what it all means completely automatically. In reality, we at least need some existing, vocabulary to refer to, even if the client can infer and interpret meaning using ontologies. For us, that vocabulary is a bespoke ontology which is currently being worked on, but which is good enough for our needs right now.

So I was heartened to be shown a glimpse of voID last year, which upon further reading looks to offer exactly the sort of service description vocabulary needed to complement our Bonjour SPARQL service type. Another great thing about RDF (see, I'm such a zealot), is that an RDF document can easily use multiple vocabularies, and the instance data doesn't even necessarily need to be linked together, so the service description document can happily accommodate the use of more than one service description vocabulary or ontology without breaking anything. We'll see what happens, but my vote would currently be for voID to be the suggested minimum requirement.

Why?

There's currently a list of SPARQL endpoints on the World Wide Web Consortium's ESW wiki, along with a comment along the lines that the list will probably not get much bigger in the long run. The comment itself is well reasoned and not necessarily meant to be a negative one, but along with similar comments from peers certainly makes us stand back and wonder at the usefulness of any kind of registry, or even a decentralized set of registries, of SPARQL services.

There's no doubt that the project I'm working with needs this right now, so it's certainly useful, but what about when, and if, things get much bigger? A well known web technologist has published Bonjour service types for HTTP and HTTPS at dns-sd.org, but lately there doesn't seem to be much take-up for this kind of facility, even if Apple's Safari web browser has built in support with its Bonjour bookmarks feature.

> *There's no doubt that the project I'm working with needs this right now, so it's certainly useful, but what about when, and if, things get much bigger?.*

On the other hand, Google does such a wonderful job, giving us a single endpoint to query the whole of the web, that it's tempting to think that the semantic web will ultimately be sucked up in a similar fashion into a great big semantic data warehouse in the clouds, and there'll be no need for anyone to offer up their own SPARQL endpoints.

My own expectation is that we'll end up with something in between. On the one side there will be one or a few massive Google like entry points which will, by spidering across the web, perhaps following the linked data trail, suck up much of the semantic web and present it to users in a simple, easy to consume and re-use entry point. But just as when you're searching for something specific, for instance buying a book, or trawling through your bank statement, you don't start at Google, I think there are plenty of cases where you'd want to be able to automatically discover SPARQL endpoints which you can access directly and which offer you precisely the details you need, there and then.

The semantic web is different to the (syntactic?) web, in that it makes it much easier to take information from multiple sources and join it together into something new and more useful. Imagine if everyone's desktop were part of the semantic web, with all the information about from emails, photos, music etc. offered up through SPARQL services (take a look at the Nepomuk project for an existing implementation of this). Imagining the sorts of applications we could build on top of the Semantic Desktop is an exciting prospect, but it's not something I'd expect most people would want to make available through a single Google style data warehouse, even if there were privacy safeguards in place, in the same way that most people don't want to necessarily share all their photos on Flikr or Picasa.

> *The semantic web is different to the (syntactic?) web, in that it makes it much easier to take information from multiple sources and join it together into something new and more useful.*

Using Bonjour to discover SPARQL services means that a user can easily create or select a list of domains where their client applications will look for published enpoints, be they public or private, and given the right credentials they will be allowed to use those services to build the sorts of applications they want, no matter how esoteric: show me a list of all the photos of my son, sorted by the number of teeth he had, using my local photos as well as the photos taken by his grandparents; give me a list of washing machines, along with prices from these suppliers, where the washing machine is no wider than 60cm and has a Which Best Buy rating over 70%.

Of course, if we start making all this sort of information available through our own SPARQL services, then there are all sorts of issues around trust, privacy, provenance and accuracy, but at the very least we are in control of our own data.

Try It!

Using Bonjour to publish and discover SPARQL services is simplicity itself, and I invite you to take a look at www.floop.org.uk/eagle/discovering-sparql for some command-line examples.

While at the moment none of the SPARQL servers out there publish their details using Bonjour, I've created a simple application for Tomcat which can publish any of the services it runs using Bonjour, and have used it to publish Joseki based SPARQL services. I'm also working on both a SPARQL endpoint and Bonjour publishing for Plone and Zope. See www.floop.org.uk/projects for more details of these nascent projects.

What would be great is if the existing SPARQL servers and triples stores out there could add optional support for registering the details of any endpoints they make available, using Bonjour.
It would certainly help the uptake of SPARQL in the projects I'm involved with.

Linked Data In(ter)action

Benjamin Nowack discusses Paggr Personal Aggregator

By Benjamin Nowack

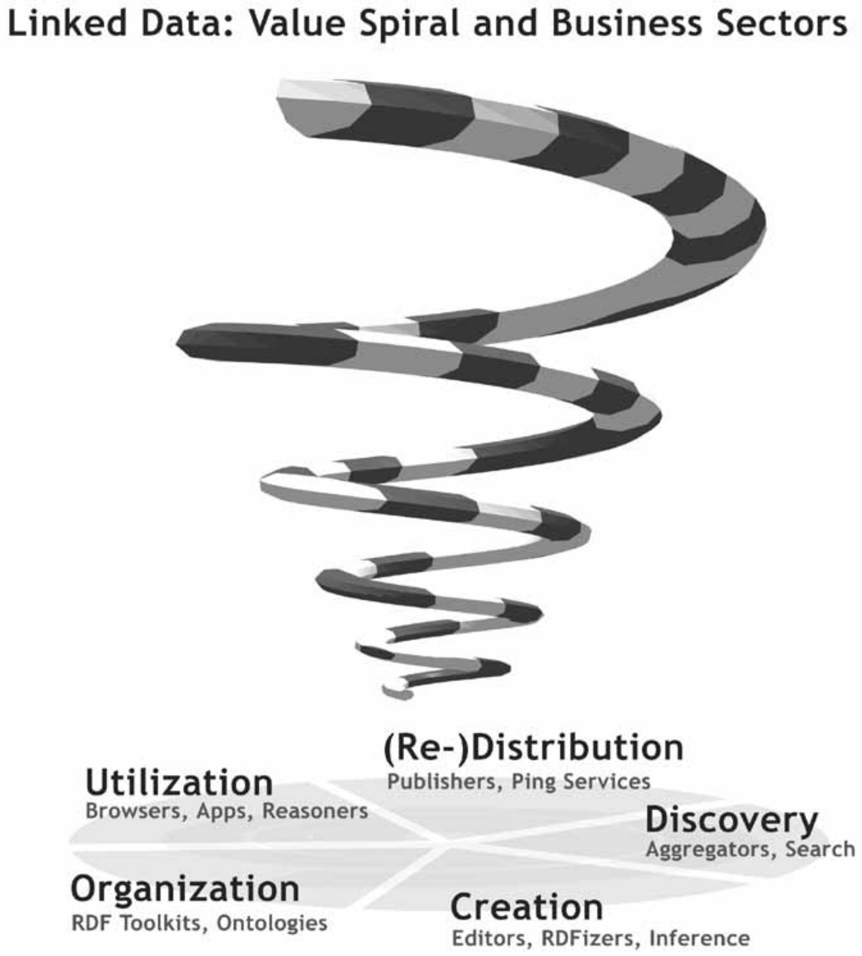
During the recent months, the Semantic Web community is accelerating its progress around web-enhanced information and knowledge management. Specifications such as RDF and SPARQL are increasingly applied by developers and organizations, RDF software is maturing. Even the initial chicken and egg problem around data and applications has now been solved by the Linking Open Data (LOD) project, which is bringing dataset after dataset online, each following recommended practices for simplified information access and repurposing. The time has finally come to move on and create the distributed data applications we have been dreaming of for so long.

Just like the Web's true innovation was not hypertext as such, but freeing it from isolated CD ROMs, the Semantic Web's value proposition is not information integration per se, but doing it on a global scale. Network effects will play an important role and have to be considered by application developers. Mashups on a semantic web are not one-off combinations of existing sources and APIs. They will feed their added value back into a self-enforcing Linked Data Ecosystem, thus enabling chains of applications, with each reaping the benefits of the previous one. RDF developers these days often use terms like "Meshup" or "Hyperdata" to describe the direction they are headed.

Linked Data is all about portability and off-site use: The more a respective application attracts users, the more will it let them take their data with them and also integrate external sources. With a bit of luck, we will see not one, but a wealth of killer applications, where the "unique selling proposition" is personal and defined by each user individually.

Despite the ongoing advances, some pieces of the puzzle are still missing. This becomes clearer when we correlate the current state of the Linked Data market to a typical information life cycle classification. While we can name solutions for each value-increasing process (Creation, Organization, Utilization, Distribution, Discovery), the Utilization and Application stage represents a bottleneck. Products start to benefit from Linked Data, but few are also re-distributing their internally enriched information. Additionally, the Creation phase today is mostly driven by dedicated efforts such as the LOD project, although data manipulation and enhancing should also be possible right while people are interacting with semantic web content.

A few months ago, Talis researcher Tom Heath wrote an inspiring IEEE Internet Computing essay titled "How Will We Interact with the Web of Data?" where he described the upcoming challenges and opportunities in the context of human-computer interaction. He suggested that on a web where the granularity is


Linked Data: Value Spiral and Business Sectors
(Re-)Distribution
Publishers, Ping Services
Utilization
Browsers, Apps, Reasoners
Discovery
Aggregators, Search
Organization
RDF Toolkits, Ontologies
Creation
Editors, RDFizers, Inference

increased from documents to arbitrary things, user interfaces should treat individual objects as first-class citizens, ideally providing context-specific functionality, direct manipulation, and coherence across personal usage scenarios. Application models that go beyond browsing and which are both universal and user-friendly are an ongoing challenge.

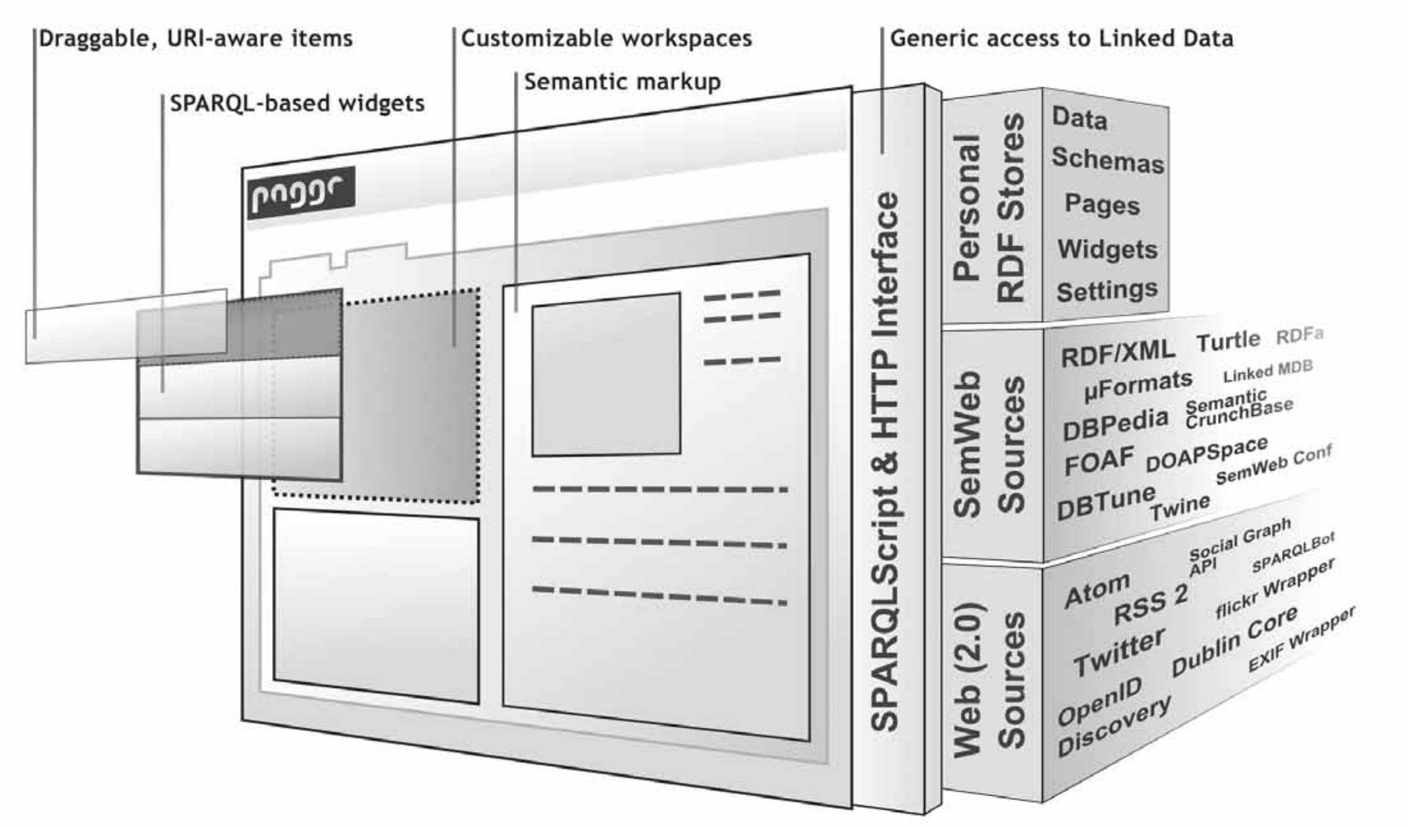
A system that aims at finding a sweet spot between simplicity and standardized interaction is Paggr (paggr.com). The basic idea is to combine successful Web 2.0 solutions and trends with Tim Berners-Lee's concept of an "RDF Clipboard" for polymorphic data exchange between desktop applications. The required technical trick for copy-by-reference across desktop and web applications was introduced by Ray Ozzie three years ago through his "Live Clipboard". Around the same time, AJAX and converging browser capabilities mass-enabled interactive HTML elements, and personal portal builders such as Netvibes brought widgets and drag and drop to end-users. The amount of open datasets and technical possibilities finally led to a first prototype for building Linked Data Dashboards a few months ago.

A system that aims at finding a sweet spot between simplicity and standardized interaction is Paggr (paggr.com).

The system used Netvibes-like pages with three resizable colums that could be populated with so-called Sparqlets. A Sparqlet is a SPARQL-powered widget, defined by a set of queries and result templates. The output consists of machine-readable HTML which addresses three essential requirements:

- Widgets can easily be copied to other dashboards, their complete definition is retrievable via HTTP (by de-referencing the widget identifier).
- Individual items in a widget can be interactively linked to other items, as each element is associated with a URI. This makes semantic drag and drop possible, such as dragging a person representation on a map or an address book widget.
- Being able to instantly feed augmented data back into the personal or public data cloud.

The prototype received encouraging and very helpful feedback at the International Semantic Web Conference (and even won a prize). We are clearly not ready for the mainstream user yet, but building on established interaction models seems to be a promising acceptance strategy. The next iteration of Paggr is now almost finished and we are looking forward to putting it online. The first public applications will be limited to focused use cases (such as an organizer for conference attendees) as we are still working on certain interface behaviors,

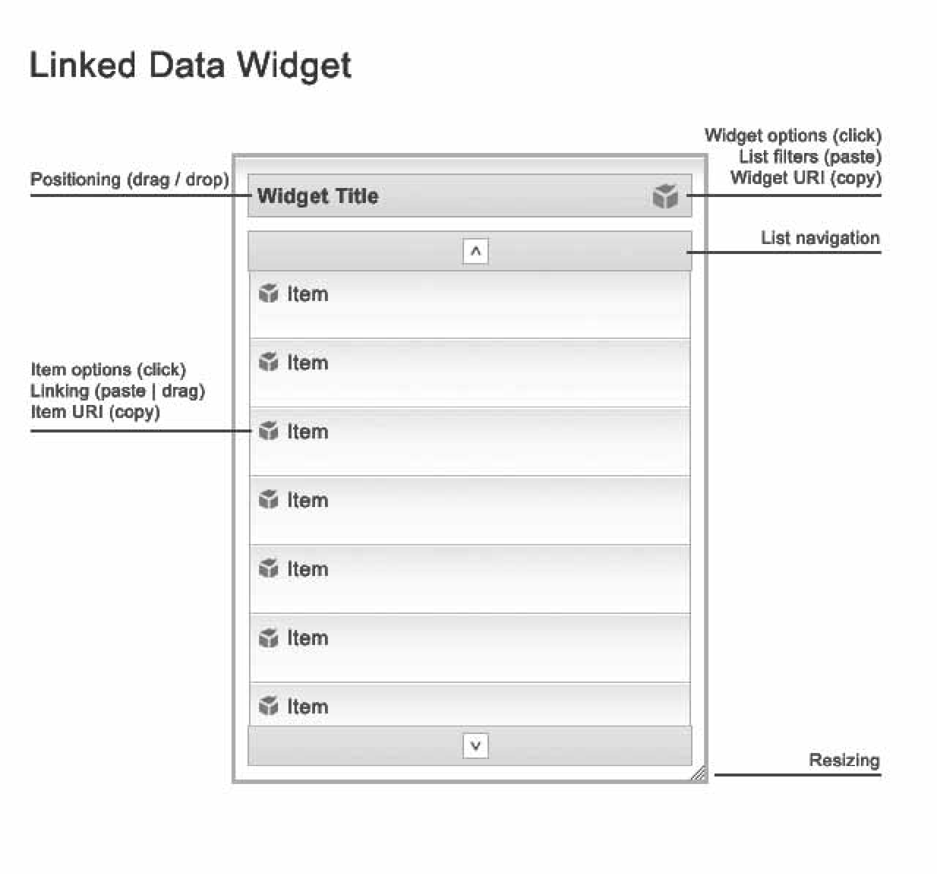


but a private alpha phase with less restrictions is planned, too.

Linked Data Dashboards face a number of usability challenges. The big question is how to tie the wealth of possibilities to a generic user interface without sacrificing work efficiency. Application convenience often boils down to feature reduction and contextual options, possibly combined with shortcuts for common tasks. To reduce complexity, Paggr lets the user (or app creator) break the theoretically infinite possibilities down into separate dashboards, where options and relations can be furhter spread across widgets.

The more complicated part starts at the widget level. Semantic drag and drop is often multi-modal. Dragging an event on a calendar does not necessarily mean "Add", there are many ways to link two persons to each other, etc. Also, working with Linked Data is sometimes like having a backstage pass for a concert: very exciting, but also a bit rough, easily overwhelming, and if you open the wrong door, you can quickly find yourself getting kicked out. Raw data (or equally ugly RDF/HTML dumps) are always just a link away, application designers will try to carefully shield non-developers from being exposed to things like DBPedia pages. For developers, on the other hand, this equivalent to the early Web's "view source" feature can be very valuable.

> *The big question is how to tie the wealth of possibilities to a generic user interface without sacrificing work efficiency.*



Now, what exactly are the requirements and nice-to-haves, and (how) can they be implemented through widgets without leading to cluttered screen estate? As mentioned above, in order to support drag and drop as well as copy and paste between different browser tabs or even at the operating system level, we can use a technical trick introduced by Live Clipboard: transparent form fields that natively provide "right-click / paste" and similar functionality. For a consistent user experience, this means that we need distinguishable (but unobtrusive) fields for each interactive element. In Paggr, small Semantic Web icons next to widget items and title bars signal the availability of advanced options. They enable

- widget filtering
- copying widget or item identifiers
- removing items from and adding items to widgets
- interlinking individual items
- custom contextual menus

The approach of using dedicated interaction zones has desirable side-effects. Non-expert users are less likely to get confused, as the general markup keeps its expected behavior. It also becomes possible to disable the semantic extensions simply by deactivating and hiding the icons. A public dashboard or shared meshup may look and feel just like a normal website.

There are still several unresolved issues left and future iterations could well require a complete re-design, but Paggr is just one of a growing number of consumer-oriented Linked Data systems. After years of hard infrastructure work, the Semantic Web community is finally starting to benefit from the investments. Data-wise, we have probably reached the tipping point already. Even former critics start to make their information available in RDF, efforts like microformats, once regarded as competitors, have become accessible from SPARQL, and services like OpenCalais, Yahoo!'s SearchMonkey, or the Zemanta API are constantly reinforcing the network effects of structured open data. It should only be a matter of months until we are going to see the first fully-fledged Linked Data applications for end-users.

Book Review

Semantic Web for the Working Ontologist

By Tom Heath

One shouldn't judge a book by its cover, or so the saying goes. In the case of *Semantic Web for the Working Ontologist* neither should one judge a book by its title. Behind this cryptic, and frankly unappealing, title lies a real gem of a book that should be of interest to anyone whose work revolves around managing, sharing or providing access to information.

Use of the term "Semantic Web" is on the increase, often accommpanied by confusion about what it actually means. In its simplest form the Semantic Web is an evolution of the World Wide Web in which not only are Web pages linked together, but data itself. For example, imagine two Web pages - one about an author and the other about a book she has written. If a hyperlink exists between these two Web pages we, as human beings, can read the text on the pages and infer something about the relationship between the author and the book. Unfortunately computers aren't so smart, so we have to give them some help. Semantic Web technologies allow us to state explicitly the connection between author and book, and share this information on the Web. We may choose to complement this with additional data about the publisher of the book, a biography of the author, or holdings information about that title.

By publishing and connecting data in this way we enable a new generation of smarter computing applications. Simultaneously we reduce the duplication of data (and duplication of effort in managing this data) by making it easier to share and reuse. Some of these trends have already been seen in so-called "Web 2.0" applications and services. The Semantic Web enables these trends to reach their full potential, and presents an opportunity for a new generation of information management, sharing and discovery services on a scale not seen since the arrival of the Web.

Semantic Web for the Working Ontologist gives a deep insight into the principles and technologies that underpin the Semantic Web idea. The authors, Dean Allemang and Jim Hendler, are leading figures in the Semantic Web field, and consequently bring a wealth of knowledge, insight and experience to the text. This isn't a book that paints a picture of the future or the full potential of the Semantic Web. In fact it will probably raise at least as many questions as it answers. Where it excels however is in acting as a reference manual for many of the underlying ideas and technologies.

The book starts with a general introduction, relaying ideas similar to those raised already in this review, before introducing the concept of Semantic Modelling in Chapter 2. Chapter 3 is probably the key section for readers who don't consider themselves "working ontologists", as it introduces the Resource Description Framework (RDF), the standard form for publishing data on the Semantic Web and linking it to data elsewhere on the Web. This chapter does a very effective job of explaining the nuts and bolts of RDF using examples from a fictional database scenario, providing a useful conceptual hook for readers with experience of that field, but hopefully without excluding other readers through overly technical language or examples.

For those readers itching to understand how applications can be developed using these new technologies and principles, Chapter 4 introduces the key architectural building blocks of Semantic Web applications. Not only does this help orient the more technical reader, it provides answers in advance to questions likely to arise in the remaining chapters. One has the feeling that Chapters 5 and onwards constitute the substance of the book from the authors' perspective. These introduce the role of inference in the Semantic Web and detail the schema/ontology languages RDFS and OWL that support this process, along with a series of design patterns (good practice) and anti-patterns (practices to avoid) to watch out for when creating ontologies.

These later chapters are aimed squarely at the "working ontologist" of the book's title, and as such are unlikely to provide much value or interest for the casual reader. However, for anyone interested in approaches to managing and structuring data they provide a sufficiently deep insight into ontology-oriented methods. Perhaps more importantly, as a general introduction to the technologies and principles behind a trend set to change the face of the Web over the next few years, this book is excellent. A little imagination may be required to apply the examples to one's own areas of interest, and depending on one's eye for detail a number of typographical and diagrammatic errors may need to be overlooked. However, for anyone interested in the technology behind the future of information management, sharing and discovery in the age of the Web, this book comes highly recommended.

Introducing: STI International

By Hench and Wahler

In the 1930's, Moritz Schlick brought together an impressive group of critical thinkers to thoroughly investigate the meaningfulness in the logical syntax of language: the Vienna Circle. Today, almost a century later, and only a couple of blocks down the street, many of the leading minds in our field meet to discuss the same fundamental problem. Except this time the discourse has taken a pragmatic shift: the members of STI International have a shared obsession in the meaningfulness in machine languages – or more importantly, what can actually be accomplished with such executable formalisms.

In the spring of 2007, the Semantic Technology Institute (STI) International was formed in order to bring these expert opinions under one scope. The foundation of STI International was a long time coming; it's predecessors, particularly organizations such as the Digital Enterprise Research Institute (DERI) research network, more or less brought the semantic spotlight to Europe long before semantic technologies earned their enabler status. Though our main office is located in Vienna, Austria, STI International extends the same collaborative sentiment on the global scale; to put it simply, STI International is a worldwide association of interested scientific, industrial, civil and political organizations, which share a common objective: to support the progressive development of semantic technologies and to ensure their role as a foundational core of modern computer science.

As we progress through our second year, we've grown to over 40 member organizations yet we are already facing the inevitable challenges paired with upholding the same open principles of the World Wide Web and its semantic successor. And though the Semantic Web will perhaps always remain our favorite showcase, our cooperation with the W3C and other notable standardization consortiums will not suffice.

Our intention is to continue to provide an encompassing vision and collective roadmap for the future development and application of semantic technologies beyond the Web, thus shaping the landscape for research in semantic technologies. As a means to this end, we support several working and interest groups, and organize multiple conferences, workshops, and events throughout the year. Three of our biggest conferences coming up next year will be the European Semantic Technology Conference (www.estc2009.com), the Future Internet Symposium (www.fis2009.org) and the European Semantic Web Conference (www.eswc2009.org).

Finally, in order to break free from the vicious cycle of successful research followed merely by more successful research, we invest significant resources in the establishment of education programs, raising awareness of new and exciting semantic technologies, and supporting the outreach to businesses in order to eventually bring such technologies to market. While this is all easier said than done, the services provided by STI International in order to achieve these ambitious goals can be loosely categorized into three main concentration areas: research, technology, and realization. These services span beyond the boundaries of what individual institutions or project consortiums are able to accomplish.

Aside from promoting and disseminating the impressive results of the STI member organizations, we also maintain the open STI International Community where all are invited to participate, regardless of affiliation or membership. Currently we provide over 3,500 profiles of those somehow involved in the development or application of semantic technologies, as well as a blog, a mailing list and an extensive collection of recommended resources for academia and industry alike. Community members drive STI International from the bottom up. Although the STI International board members represent some of the most prestigious organizations with expertise in semantic technologies, the opinions and contributions of all stakeholders in future semantic technologies are also of great value. Additionally, joining the broad STI International community provides potential members with the opportunity to find out what we're all about prior to applying for actual membership. If you're interested please take the time to visit our website (www.sti2.org).

Given that the development of semantic technologies has now matured to a stable point, i.e. semantics is now recognized as a field of computer science, it would be naïve of us to not look ahead. For those of you new to the community, or for those of you who have been around for a while, yet are still a little puzzled whenever our logo or acronym shows up, perhaps the simplest explanation is via analogy: those of us with a vested interest in the continued development and maturity of the field of semantic technologies make up the parts of the machine; STI International is the oil.

Changing attitudes to data?

Paul Miller's Cloud Piece

By Paul Miller

The sound bite is a dangerous thing. Whilst clearly valuable in providing a memorable and accessible hook upon which to hang a richer and more nuanced argument, there is the ever-present danger that we forget the nuance and focus upon the superficial simplicity of the reproduced words.

However, I heard two sound bites recently that resonated powerfully, crystallising some of my own views and serving as extremely effective aides memoir to the underlying argument.

First, speaking at the Powered By Cloud event in London's Millbank, JP Rangaswami called on his audience to revise their presumptions as to enterprise data management, exhorting them to;

"move from data centre to data centric."

Just two days later, Sir Tim Berners-Lee addressed the latest TED conference in Long Beach, California. As part of a wide-ranging piece of advocacy for the notion of Linked Data, he boiled an earlier meme of Rufus Pollock's down to its bare essentials, demanding that the assembled glitterati;

"stop hugging your data."

When I repeated Tim's call in a presentation of my own two weeks later, a member of the audience was quick to assert that data which are hugged will be 'good' data; data about which we care and in which we invest attention are more useful within the organisation and (probably) more valuable outside it. This is almost certainly true, but misses Tim's point. He is not (I would contend) suggesting that we stop lavishing attention on our data and cast it, unwanted and unloved, out onto a harsh and unwelcoming Web. Rather, he's calling on data's curators to be an awful lot less possessive. The hug of which Tim speaks is not, I believe, one motivated by care or love; but rather by possessiveness, jealousy and an unhealthy short sightedness.

Both Tim Berners-Lee and JP Rangaswami clearly recognise that the value proposition with respect to data is shifting, and that far greater opportunities await those with the foresight to set free much of that which they previously guarded jealously.

As another Tim was quick to spot, Web 2.0 companies prepared to provide access to large quantities of behavioural data in aggregate have hit upon something interesting (but, as yet, largely un-monetisable);

"By contrast, I've argued that one of the core attributes of 'web 2.0□ (another ambiguous and widely misused term) is 'collective intelligence.' That is, the application is able to draw meaning and utility from data provided by the activity of its users, usually large numbers of users performing a very similar activity. So, for example, collaborative filtering applications like Amazon's 'people who bought item this also bought' or last.fm's music recommendations, use specialized algorithms to match users with each other on the basis of their purchases or listening habits. There are many other examples: digg users voting up stories, or wikipedia's crowdsourced encyclopedia and news stories." (http://radar.oreilly.com/2007/09/economist-confused-about-the-s.html)

Data of this sort have often been perceived differently to those collected, managed and hoarded in corporate data centres. They have been the 'digital exhaust' belched out onto the Web as the by-product of more mission-critical activities, and few have taken this 'waste' sufficiently seriously to consider the ways in which it might be used.

The success of various attempts to harness this flow of essentially public information has led to a rethinking of its utility, although viable business models that rely solely upon monetising access to or analysis of the exhaust itself remain hard to find.

Within more traditional enterprises, too, we are beginning to see examples that demonstrate the benefits to be gained in being a little more forthcoming with respect to much of the data stored in corporate data centres.

For too long, procedures and policies have been applied to corporate data in toto; the same protection is expensively applied to a street address, for example, as to details of the customer who resides there.

By thinking more about the data rather than the data centre, and by adopting a default attitude in favour of - rather than against - sharing and openness, we may end up saving a lot of money whilst at the same time producing an incredibly powerful resource.

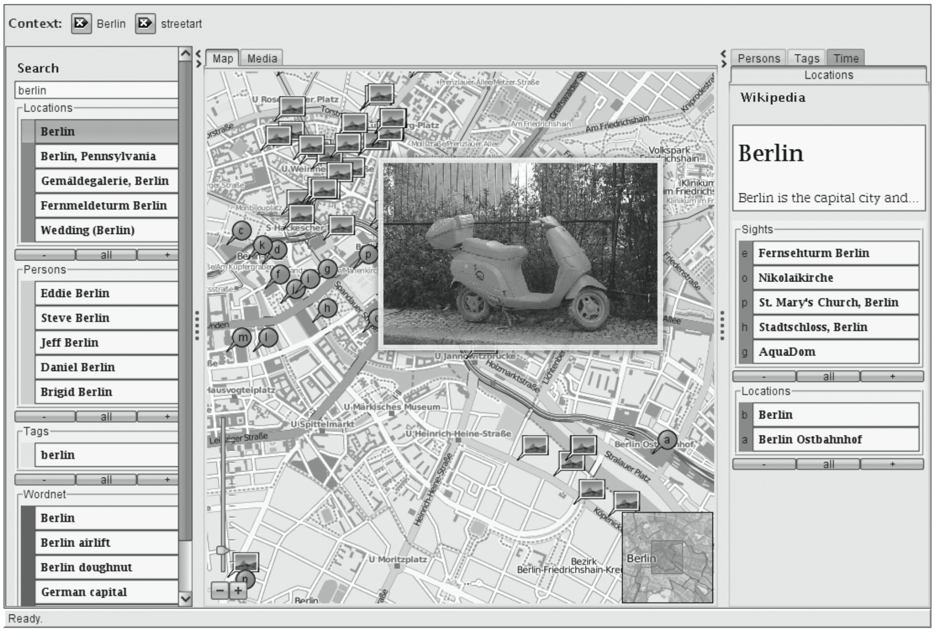
It's certainly worth thinking about.

Social Semantic Web scales in the Cloud

Simon Schenk introduces SemaPlorer, 2008 winner of the Billion Triples Challenge

By Simon Schenk

A research team from the University of Koblenz, Germany has won the Billion Triples Challenge 2008 for demonstrating the scalability of the Social Semantic Web with their SemaPlorer system. SemaPlorer allows for searching and browsing of multimedia data using semantic background knowledge. This background knowledge is obtained from distributed sources of linked open data using distributed views and querying based on the Resource Description Framework (RDF) and its query language SPARQL. Cloud computing services are used to make these large amounts of data available on demand. SemaPlorer proves that semantic technologies can be used for the significant improvement of user experience with social media while at the same time allowing for great flexibility and short development cycles. The underlying technology developed at the Information Systems and Semantic Web (ISWeb) group is being used in multiple European research projects and in commercial environments.



> The underlying technology developed at the Information Systems and Semantic Web (ISWeb) group is being used in multiple European research projects and in commercial environments.

Everyone knows the Web 2.0: Wikipedia, Google Maps, Flickr, location names like Geonames, directory structures such as WordNet, and much more are visible via a few clicks. However, it remains tedious and difficult to connect pieces of such information using traditional information systems. There, the Semantic Web comes into play. It interlinks information and allows for intriguing questions, e.g.: “Where do I find streetart in Berlin and how does it look like on photos?”

However, semantic technologies did not have the very best reputation – until now. Prejudices involved complaints about complexity of the technology and a lack of speed. In order to reveal that new developments of Semantic Web technologies have made such rumors become outdated, Peter Mika (Yahoo!) and Jim Hendler (Rensselaer Polytechnic Institute) have initiated the Billion Triples Challenge at the 7th Int. Semantic Web Conference.

A challenge to scale the Semantic Web

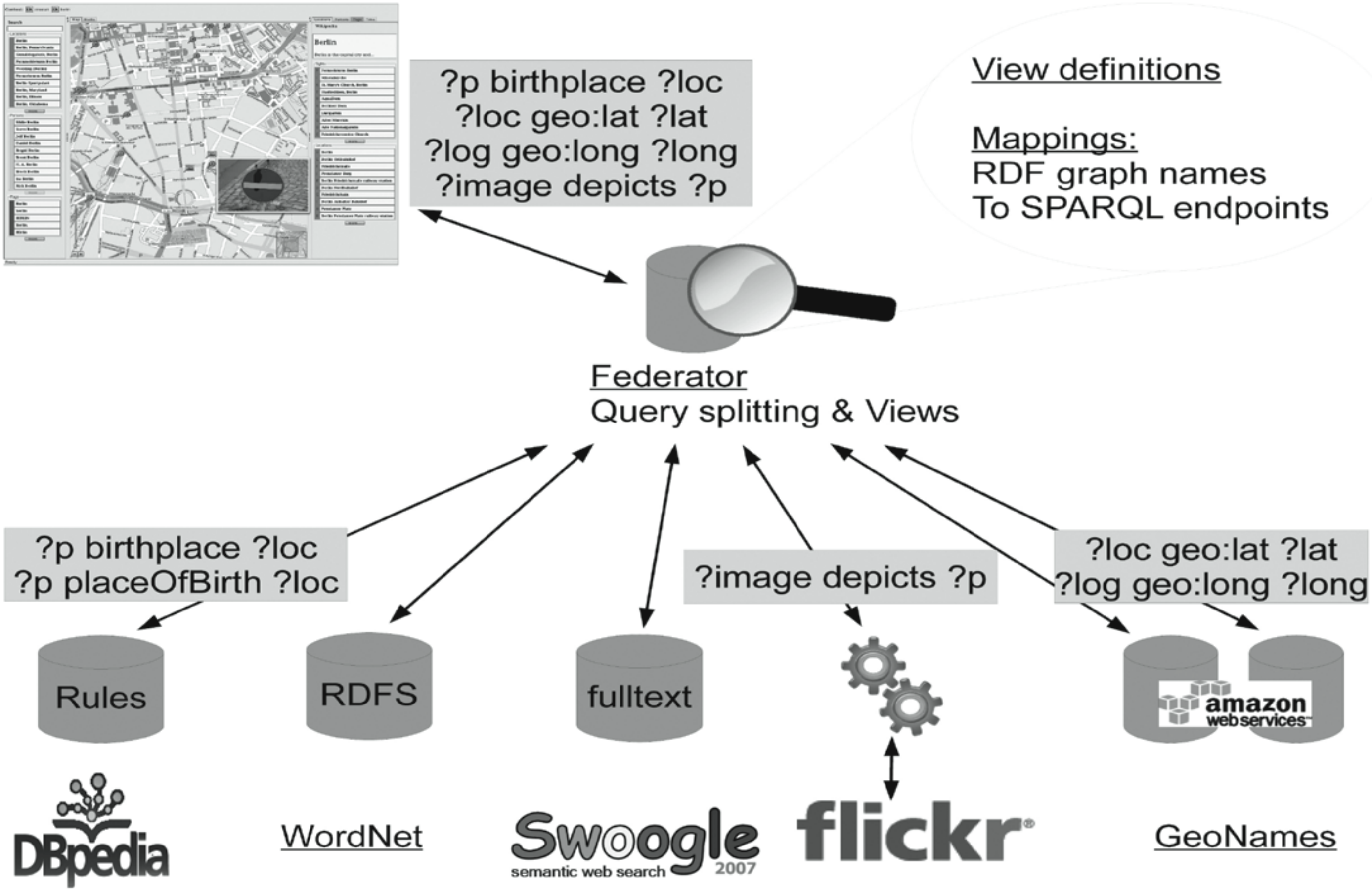
The aim of the Billion Triples Challenge is to “add value to [a] very large triple store. [...] The key goal [was] to demonstrate an interaction with the large data-set driven by a user or an application. […] It was desired that all applications assume an open world, i.e. that the information is never complete, [that the application is] scalable (in terms of the amount of data used and in terms of distributed

[1] http://dbpedia.org

[2] http://geonames.org.

[3] each 3http://wordnet.princeton.edu

[4] http://swoogle.umbc.edu



components working together) [and] the application should [...] function in real-time."

A predefined dataset was provided from the organizers of the Billion Triples Challenge. The dataset included DBPedia[1], a mapping of the online encyclopedia Wikipedia to the knowledge representation language RDF, geographic information from Geonames[2], statistical information from the CIA World Factbook, Eurostat, US Census, a formalization of the English language called WordNet[3], as well as crawls of semantic information from personal home pages such as Swoogle[4]. This data greatly varies with respect to quality – from official statistical data such as Eurostat, via Web 2.0 content such as DBPedia, to personal FOAF files in Swoogle. For the SemaPlorer system, DBPedia, Geonames, WordNet, and the Swoogle crawl have been used. In addition, a crawl from Flickr was taken and converted to RDF. This crawl contains the photos uploaded to Flickr over a one year period and includes the photo's tags, users, groups and geographic and temporal references. The SemaPlorer dataset added up to a size of 1.2 billion RDF statements.

The SemaPlorer dataset added up to a size of 1.2 billion RDF statements.

SemaPlorer

The SemaPlorer application developed by the ISWeb research group led by Prof. Steffen Staab outperformed the eight international competing teams. SemaPlorer offers traditional keyword search as well as content-based navigation along factual relationships. Searching, e.g., for "Berlin" lists proposals for disambiguating the term such as the several locations called Berlin but also persons with "Berlin" as their last name. Selecting the German capital displays a map and related information such as celebrities, sights, and points of interest. One may refine the search by typing in further keywords or by interacting with the semantic data provided by SemaPlorer. As one can see in Figure [insert reference to screenshot here], contextual information for the currently selected area of interest is shown in the facets on the right hand side of SemaPlorer. For example the Reichstag (house of the German parliament) is linked to, e.g., photos from Flickr (including information about their creators), information about the Reichstag from Wikipedia, nearby locations, people living or working in the area, and others.

When a user adds a search term or selects an item linked to the current content or location, the map view and context panels of SemaPlorer are updated. For example, adding a keyword like "streetart" results in narrowing down the displayed pictures in the are. Clicking on a location or sight changes the entire area of interest. Hence, navigation and search corresponds to building a semantic context of the content. Every time the user clicks on some item, the context changes and is translated into a set of up to eight queries against the underlying data storage, which are concurrently evaluated. The query results are displayed as the new context.

Federated Querying and Views for the Semantic Web

Some of these context queries span multiple datasets. For example, sights are geographic entities from GeoNames, categorized as visitor attractions in DBPedia. Hence, facilities for integrated querying of multiple datasets are needed. Such facilities can be large centralized triple stores holding all relevant data, or lose coupling and distributed querying.

Nowadays semantic repositories are available that could handle the aggregated size of the datasets used for SemaPlorer. However, the Semaplorer Team believed collecting all relevant data into a single repository is not desirable on the Semantic Web:

- The Semantic Web – just like the document centric Web today – is inherently distributed. Its very quality is the ability to integrate data from heterogeneous sources and its extensibility and openness.
- Scaling a single repository will solve the problem for the first billion of triples, but in the mid term – or even today for large social networking or media sharing sites – we are facing amounts of data orders of magnitude larger than that.
- Finally, data on the web is always incomplete. Thus, an approach that allows for inclusion of additional data sources at almost zero cost is needed. Hence, SemaPlorer is based on a dynamic federation of distributed RDF repositories. Each of the repositories holds (a fragment of) a single dataset. A federation component takes care of forwarding queries to the appropriate repositories. If a query spans multiple repositories, it is split and evaluated in a distributed manner. All this is completely transparent for the actual application (although, of cause, the federator could be built into the application). For querying only standard SPARQL is used. Hence, arbitrary SPARQL conform repositories can be hooked into SemaPlorer.

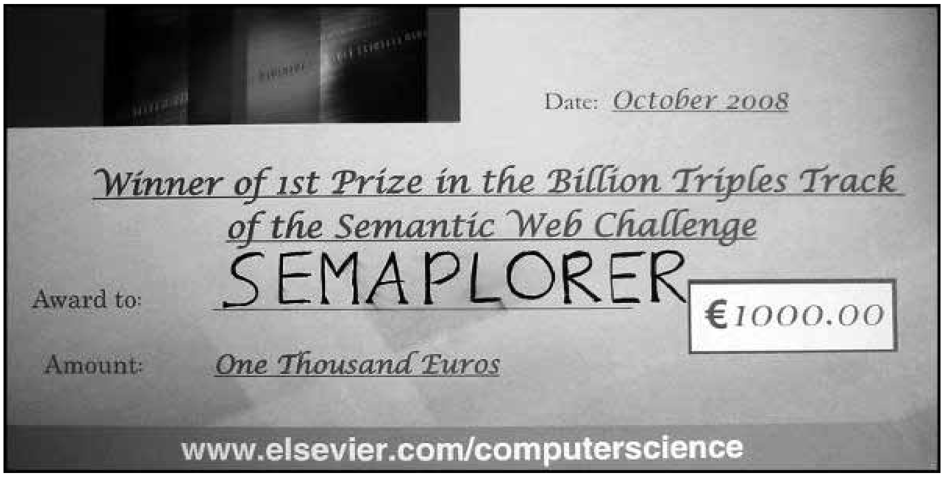
Nowadays semantic repositories are available that could handle the aggregated size of the datasets used for SemaPlorer.

SemaPlorer is based on a dynamic federation of distributed RDF repositories. Each of the repositories holds (a fragment of) a single dataset.

The datasets on the Semantic Web employ different schemata. Moreover, some of them are of medium or low quality. For example, the DBPedia dataset provided contained 13(!) ways to express the place of birth of a person. In order to cope with this ambiguity, ISWeb member Simon Schenk developed a view mechanism called Networked Graphs[5]. It allows using SPARQL CONSTRUCT queries to define views for RDF, just as SQL for views in relational databases.

Using Distributed SPARQL and Networked Graphs, the SemaPlorer dataset has been distributed over 25 repositories. 10 of them were hosted by University of Koblenz in Germany, the rest was run on the EC2 cloud computing service by AmazonTM in the US.

All configuration of the system is done in RDF and stored in the repository of the federator. This makes reconfiguration very light weight, as it boils down to changing a couple of RDF statements



Date: October 2008

Winner of 1st Prize in the Billion Triples Track of the Semantic Web Challenge

Award to: SEMAPLORER

€1000.00

Amount: One Thousand Euros

www.elsevier.com/computerscience

[5] http://isweb.uni-konblenz.de/Research/NetworkedGraphs

[6] As the Flickr API is not as expressive as full SPARQL queries over the crawled data, some queries possible in the demonstrator used for the challenge are no longer possible on top of the current infrastructure.

in the federator. Subsequently to the challenge, the team developed a mapper from SPARQL queries to Flickr API calls, which now works as a virtual RDF repository. To switch from the crawled Flickr data stored on EC2 to the live Flickr wrapper has been done within one minute in the live system and without any changes to the actual application. Thus, SemaPlorer now works on live Flickr data[6].

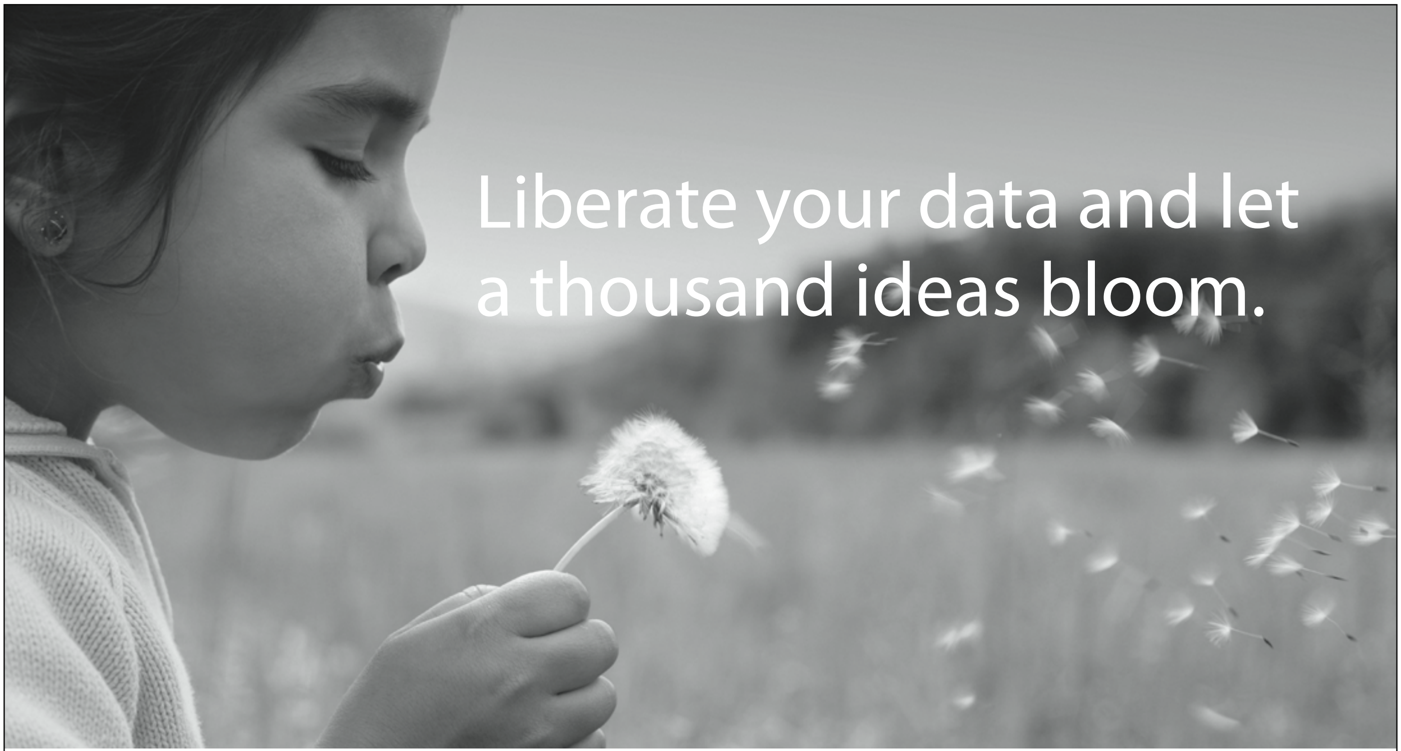
Conclusion

SemaPlorer demonstrates that the scalability necessary for building the Social Semantic Web is available today. Using data expressed in the very flexible and interoperable RDF format together with extensions for the standard query language SPARQL and cloud computing services has enabled the rapid building of an extensible semantic application. Prof. Staab comments that „[…] we could demonstrate for the first time that the vision of Web 3.0 – an advancement of Web 2.0 that includes semantics – actually works. So far, this had been done only with toy examples." Parts of the SemaPlorer infrastructure are used in various European research projects and industry.

The project team of SemaPlorer includes the leader of the research group ISWeb, Prof. Dr. Steffen Staab, the project coordinator and multimedia expert Dr. Ansgar Scherp, the semantics experts Simon Schenk and Carsten Saathoff as well as the computer science students Anton Baumesberger, Frederik Jochum, and Alexander Kleinen. The prize was awarded by an international jury during the 7th International Semantic Web conference – with more than 620 attendees the most successful in the series.

The work on SemaPlorer has been co-funded by the EU in FP6 in the NoE K-Space and NeOn project and FP7 in the WeKnowIt project.

Further information including a video and the SemaPlorer application is available at: http://btc.isweb.uni-koblenz.de



Streams, Pools and Reservoirs

By Leigh Dodds

As we start to move past the current boot-strapping phase of the semantic web in which we are constructing the web of linked data, its useful to begin discussing what other feature and infrastructure we need in order to support sustainable usage of this huge and growing data set: what services can be offered over linked data? Do we need to consider how to provide quality of service, stability and longevity to the data, or does the sheer scale of the web make these moot points?

In order to answer this question it's useful to compare the ongoing development of the linked data web with that of the web itself.

A Brief History Lesson

There have been several phases of activity in the development of the web. While in truth, these phases were of different duration, overlapped with one another, and have happened at different rates within different communities, essentially we have gone with the following basic steps.

Firstly we concentrated on just getting stuff on line. The early web was a new medium for document and data exchange and so was at its core a simple publishing device used as a collaborative space between small communities. But as the amount of content and the size and breadth of those communities grew, the emphasis shifted towards linking: tieing content together to create, - initially hand-crafted – indexes of the web and knit the available content into a greater whole.

The second, manual linking phase was quickly supplemented by a third phase of automated linking between content: search engines. A search engine is simply a way to quickly create a link-base based on some search criteria. The crawling and indexing of the document web by web crawlers allows users to quickly construct links to content of potential interest.

The third phase of the web's development has been triggered by the commodisation of search and the need for search engines to differentiate themselves and offer additional value-added services. Search engine features are now tailored towards particular uses or types of content (Google Image Search; Google Scholar); offer value-added features that capitalise on the ability for search engines to analyse the structure and traffic flows across the web (PageRank and similar indexing improvements; Google Trends); expanding the audience for content (Google Translate); and enabling community-driven customisation of the search experience (Google Custom Search; Yahoo Search Monkey, etc).

No doubt there will be subsequent phases of development, and the perspective of history will let us tease out common strands of development some of which will already be happening. But if we look at the recent, rapid development of the linked data cloud, we can already see that the same pattern is being repeated.

History Recapitulated

There has been RDF data available on t he web for many years, used by a limited community of researchers. This slow accumulation of content – echoing the first phase of content publishing on the document web – has been replaced by a rapid increase in data publishing encouraged through the Linking Open Data (LOD) project. By providing clear pragmatic guidance and instructions on how to publish data for the semantic web, that project has enabled us to accelerate our transition through that first content publishing phase. But it has also, crucially, encouraged the linking together of data sets (Phase 2).

This linking has to a great extent been manual. Not in the sense that members of the LOD community are manually entering data to link datasets together, but rather at the level of looking for opportunities to link together datasets, encouraging data publishers to co-ordinate and inter-relate their data, and by attempting to organically grow the link data web by targetting datasets that would usefully annotate or extend the current Linked Data Cloud.

The rapid growth of the Linked Data Cloud means that this "manual" phase will soon be over: there will be sufficient momentum behind the semantic web that increasing amounts of data will become available and no single community will be able (or need) to shepherd its development. The focus will shift towards the subject specific communities who will instead co-ordinate at a more local level. Semantic web search engines will also become a reality.

Semantic Web search engines need to be distinguished from semantically enabled search engines. The latter use techniques like natural language parsing and improved understanding of document semantics in order to provide an improved search experience for humans. A Semantic Web search engine should offer infrastructure for machines. This Third Phase is also beginning to take place. Simple semantic web search engines like Swoogle and Sindice provide a way to for machines to construct link bases, based on some simple expressions of what data is of relevance, in order to find data that is of interest to a particular user, community, or within the context of a particular application. And crucially this can be done without having to always crawl or navigate over the entire linked data web. This process can be commoditized just as it has with the web of documents.

> *This can be done without having to always crawl or navigate over the entire linked data web. This process can be commoditized just as it has with the web of documents*

Co-Evolution of the Web Infrastructure

Given the strong concordance between the phases of development of the document and linked data web, it is reasonable to make some predictions on how semantic web search engines, and additional supporting infrastructure, is likely to evolve by comparing them with the development of human search engines. For each of the specializations and value-added features listed earlier its possible to see an equivalent for the machine-readable web:

Document Web	Semantic Web Infrastructure	Description
Google Image Search	Type Searching	Ability to discover resources of a particular type: e.g. Person, Review, Book
Google Translate	Vocabulary Normalization	Application of simple inferencing to expose data in more vocabularies that made available by the publisher
Google Custom Search	Community Constructed Data Sets and Indexes	Ability to create and manipulate custom subsets of the linked data cloud
Google Trends	Linked Data Analysis & Publishing Trends	Identifying new data sources; new vocabularies; clusters of data; data analysis

These last two are particularly interesting as they suggest the need to be able to easily aggregate, combine and analyse aspects of the linked data cloud. This infrastructure will need to be able to support the community in working with data in a variety of ways, allowing data to flow and be collected where it is needed. Introducing a metaphor for this process might help highlight some of the processes and its consequences.

Flowing Data

Data is like water and flows of data are like streams. These streams of data can arise from any number of different sources: from a person entering data into a system; from a click stream generated as a side-effect of web browsing; application events; or generated from real-world sensor measurements. There are already many ways that we can tap into these data streams, using web-based query APIs, messaging systems like XMPP, or syndication protocols like Atom and RSS.

While these streams of data are already supporting a huge range of different applications and use cases, they are inherently limited: a stream has no memory. If historical context is required, e.g. to support more complex querying and reporting, then each consuming application must collect and store the data. We can think of these collections of data as pools; each stream of data on the web may feed any number of different application-specific pools.

A pool of data provides extra flexibility, but comes at the cost of requiring each consuming application to maintain its own infrastructure to hold copies of that data. Even if each source of data provides direct access to its own pool, e.g. by exposing a web-based query interface onto its database, or by exposing linked data, there are still unnecessary overheads. Each data provider must provide their own scalable infrastructure and support a rich set of data access options.

If we start building large pools of data, within a community supported infrastructure, then we have a reservoir. A reservoir is a pool of data that is maintained by and services a specific community. Reservoirs allow issues such as quality of service (reliable supply of water) and infrastructure costs (building of pipelines) to be solved at a community level.

Its possible to argue that the web already consists of streams, pools, and reservoirs, but there is a distinct difference between a web based on semantic web technology and a Web constructed of a mixture of XML documents or similar formats: like water, at the molecular level, all RDF is the same; its all triples. Unlike alternatives, RDF data is more easily pooled and collected and so is much more amenable to explorations of shared infrastructure. Like a relational database, an RDF triple-store can contain an huge variety of different kinds of data,.But unlike a relational database, an RDF triple-store, has the potential for the aggregate to be much more than the some of its parts. The seeds of convergence are built in, through reliance ah the most fundamental level on a global naming system (URIs) and standardised ways to state equivalence and relationships between resources.

In the real world, reservoirs do more than supply a community with water. The aggregate has its own uses: water skiing or hydro-electric power generation for example. And the same will be true of semantic web data reservoirs: large collections of data can be analysed and re-purposed in ways that are not possible – or at least not achievable without a great deal of repeated, redundant integration effort – using other techniques. The reservoir itself can be the source of new facts and new streams of data derived from analysis of its contents.

Flowing Data through the Talis Platform

The goal of the Talis Platform is to support the growth of the Linked Data ecosystem by providing the infrastructure to support the creation of pools of data. For additional background, see my article “Enabling the Linked Data Ecosystem” from Nodalities issue 5.

> *The goal of the Talis Platform is to support the growth of the Linked Data ecosystem by providing the infrastructure to support the creation of pools of data.*

At present the Platform provides a range of services that allow data to be easily streamed into and out of Platform stores, allowing data to be easily pooled in order to benefit from greater context. Data can be pushed directly into the Platform and we are exploring methods of supporting other forms of data ingestion to make it easier and more natural to begin to accumulate data sets within the Platform.

The core search service, which produces its results in RSS, allows the creation of simple data streams, while the SPARQL interface supports more complex data extraction methods. The Augmentation service provides an interesting twist on these conventional approaches, providing a means for any RSS 1.0 feed to be automatically enriched with extra metadata by feeding it through a Platform data store. This means of interaction is like fishing for data: it is possible to serendipitously find and extract data, capturing it as extra context to items in an RSS feed, without having to deal with writing SPARQL queries or constructing a keyword search. There are many more methods and modes of data extraction that will be added to the Platform to add to these existing services; this is just the beginning.

But the Talis Platform is intended to provide much more than just the ability to work with pools of data. The bigger vision is to support the creation of true data reservoirs, and enable many different ways of manipulating and analysing their contents in order to discover new facts and bring new context to that data. Creation of these larger pools of content will need to be made sustainable for the communities that are creating them, and deriving value from them. Sustainability covers a wide range of issues that go beyond just commercial issues: quality and range of services are additional factors, as are forms of governance, trust and quality that relate to the data sets themselves. The Platform is intended to address all of these issues.

To take a small example, the experimental “store groups” feature that was released at the end of last year, provides a simple method for combining datasets, without requiring that data to be completely loaded or copied into a single database. The store groups feature will ultimately support a range of services over the constituent data sets, allowing each pool of data to remain intact whilst still contributing to the whole; this will be important to support the new forms of governance that are beginning to emerge around datasets on the Linked Data web.

Leigh Dodds is the Programme Manager for Talis Platform

Nodalities Magazine SUBSCRIBE NOW

Nodalities Magazine is made available, free of charge, in print and online. If you wish to subscribe, please visit www.talis.com/nodalities or email nodalities-magazine@talis.com.

Nodalities Blog

From Semantic Web to Web of Data

blogs.talis.com/nodalities/