Questions Submitted By ERRT Members for Linked Data Session

1. From Richard Urban:

There are a lot of benefits to the atomic metadata statements that are making up Linked Data resources. At the tippy top of the Semantic Web stack is a block labeled "Trust." I've most frequently seen it discussed in terms of trusted e-commerce transactions. But what does "trust" look like for Linked Data. There are many sources, some data comes from trusted institutions, but frequently it is copied, translated and made available through other venues (e.g. data.gov information represented in DBPedia). Your work on OPM suggests that provenance assertions can quickly surpass the quantity of the original data being described. How much do we need to build trust (and how do we trust the provenance data?)

In general surveys have suggested that people trust information institutions like libraries, archives and museums. How can these institutions build trust in their Linked Data that distinguishes it from other sources?

Trust is overrated. Information-poor networks rely on trust to mitigate the paralysis that results from not having enough information confirm the factual basis upon which a decision needs to be based. The solution is more information, not more trust. Provenance is useful not because it leads to trust, but because it leads to confirmation (confirmation can lead to trust, but so what?). The metaphor I like to use is situational awareness. Not all semantic web statements are analogous to utterances of conscious agents; in my line of work they can be, for instance, data points produced by sensors. We don't need to “trust” our eyes and ears; we just need to make sure they e.g., agree with each other; external sensors are no different, as long as we take basic steps to prevent tampering and detect failures.

I joke that the level above “trust” in the Semantic Web stack would be “love”, but maybe a more accurate extrapolation would be “faith”. I say this because I think the amount of substance decreases as you go up the layers, and the implication that trust is the ultimate goal is misguided. In science, our goal is reproducibility.

I don't think my work on OPM suggests anything like what you're saying it does about data volume. For two reasons:

1. In HPC we measure data in petabytes, and there is no class of metadata that comes anywhere close.

2. OPM is designed to not impose any level of granularity, so it can be used to record “coarse” provenance information. Some provenance systems that do impose granularity requirements, such as PASS, produce orders of magnitude more provenance information than we generally produce in OPM from a workflow system, but again, high-resolution data itself is orders of magnitude more voluminous than even PASS provenance.

Database provenance is one area where the amount of provenance information approaches the amount of information it describes, but for data-intensive applications, bulk data in files is generally larger than row data in databases.

2. From Cathy Blake:
Yes I have a question is there a quick study guide I could read to ask better questions about this model?

I really want to know if the original frog still exists after it has been vaporized ;) [Note: this comment refers to an earlier, ontological discussion of identity across change]

Because linked data is a simplified and restricted subset of RDF, it helps to understand RDF and the Semantic Web. Tim Berners-Lee's 2001 Scientific American cover story on the Semantic Web gives one vision of what kinds of issues the semantic web is designed to solve, focusing on the autonomous agent use case. The RDF Primer (at w3.org) and the O'Reilley book “Practical RDF” also fill in bits of the puzzle. Some recent writing by Berners-Lee is relevant as well; all of this is easily found on the web.

As for the frog, I think the burden of proof is on anyone who wants to reject the naïve realist notion that vaporizing a living thing destroys it.

3. From Tim Cole:

Would be interested in Joe's opinion and possibly a bit of discussion about the Media Fragments Working Group 1.0 Recommendation from a Linked Data perspective. The broader issue is how linked data apps should or might be able to deal with statements about URIs that are fresh minted on the fly to talk about parts of resources, e.g., as in the media fragments case regions of still images or segments of video. Such URIs are unlikely to repeat, but there may be times when recognizing a relationship between 2 media fragment URIs that share a common base before the #. Is there any sense of hasPart or resource hierarchy accommodated in Linked Data models?

One of the stated design criteria for the Media Fragments work is providing a way to generate URL’s that refer to parts of media items, so that RDF statements can be made about those parts. That’s relatively unproblematic if you just want to make a statement. More interesting is when you want to find relevant statements.

Here I think resource hierarchy is actually not problematic; I can assert, on the fly, that a fragment belongs to a media resource it’s a fragment of, and vice versa. If I want to manage a set of assertions about a media resource and associate them with the resource via a predicate like dc:hasPart, I don’t need a new representation or new features of RDF or linked data to do that.

Where it gets interesting is where we get into spatial (including temporal) relationships. These are geometric, and RDF has no provision for geometry at all.

Consider overlapping media fragments. How do we write a SPARQL query that returns any fragment that overlaps with a given fragment? In short, we don’t. We could probably hack something together with range queries, but again we’d be kicking a larger can down the road, which is that to do spatial queries in any complex domain, we need spatial indexes and coordinate transformations.

I am not an advocate of attempting to modify the RDF or linked data model to “natively” support these kind of operations. It is much more practical to simply represent spatiotemporal extent
using ordinary semantic web representations, and then build hybrid systems that can index and query that kind of information. Time may be exceptional in that we generally think identity persists across time, whereas we don’t generally think it persists across space (for sparse entities we simply have spatial extent models with holes in them). Alejandro Rodriguez’s “time-annotated RDF” addresses this issue, but its semantics have yet to be nailed down (we’re using it anyway).


4. From Kevin Trainor:

I had a conversation with Joe recently regarding the building of a CIRSS portal using Liferay. I recall Joe having said that he had lost interest in custom portal interfaces because of his belief in the Linked Data approach. So, my questions are: To what extent are Web portals and Linked data alternative solutions? To what extent are they complementary solutions?

Portals (and some content management systems) are very good at collecting data and keeping it in one place. Their interfaces are useful, but they are designed to be extended by adding features to the portal, not by having the portal interoperate with cooperating, independent tools used as part of a complex work process. For example much of the data backing a Liferay instance (last time I checked) is represented in Hibernate. This means that the persistent store contains collections of serialized programming language objects, and to interpret them one must use the (nonstandard) Liferay API. There are various reasons portals have been architected this way, but the primary reason is the persistent habit programmers have of thinking that every problem can be solved by developing yet another three-tier database-backed web application. Ultimately, these applications do not scale. We need distributed data management, and as soon as we get there, we need representations that achieve economies of scale—so: global identifiers, links, and indexing on globally-relevant spatial relationships like time and geolocation. Local structures are opaque at scale.