Overview

- Graphs
- The Wide Open World
- Ontologies and Identities
- Serializations and Technologies
- Linked Data
- Temporal Issues

- Summary

<blink> WARNING: Controversy Within </blink>
RDF, Linked Data, Semantic Web… oh my!

The Semantic Web was a great idea in 2003
The Semantic Web is still a great idea in 2013

Lots of progress made, but lots of challenges remain

This talk is not:

• An introduction to RDF
• An advertisement for Linked Data
• An apology for the Semantic Web
Graphs

✔ Graphs are very powerful for modeling reality
  ✔ Tree is just a simple Graph  
    (directed, acyclic, with known root node)
  ✔ Novel information can be automatically inferred
  ✔ More interesting questions can be asked
  ✔ Don’t end up in XML semantic/syntactic hell
Graphs: Structure and Data

➢ Querying is much more complicated

Graph: Structure and Data important, but data currently treated as second class citizen
Other: Only Data important, so easier to work with

Graph Query: Find books written by “Sanderson”

?who a f:Person ; f:name “Sanderson”

Or when you don’t need to worry about structure:
au:Sanderson
Graphs: Structure and Data

- Serialization and Storage are complicated

Serialization has a start and end, unlike a graph
- No obvious place to start or stop writing
- Need to link within the serialization, and without
- Will come back to this … it gets worse!

Storage can’t look at “documents”, deals with structure
- All assertions in graph (triples) are stored in one big lump
- Serialization is a very temporary artifact to transfer some subset of the triples
- Needs to worry about structure and data, not just data
Graphs: Structure and Data

- Visualization is difficult to get right
  - ... and hard to know when it is right

- Documents “easy” to visualize
  (eg browser, pdf viewer, image viewer etc)

- Data visualization understood
  (eg ManyEyes, charts, time series, etc)

- Graph visualization almost universally terrible
  ... because structure is crucial
Visualization Done Right

Not So Right
Graphs

✓ Graphs are very powerful
➢ Graphs become complicated to work with

Mitigating Factors:

• Software libraries can manage complexity
• NoSQL solutions improving rapidly
• Treat part of the graph as a document (having cake)
  And also ingest into TripleStore (eating it)

• Other structures don’t get complicated
  because they lack the expressiveness of a graph
The Open World

✓ A Single Global Graph that everyone contributes to
  ✓ Great for data re-use
  ✓ Richness of data from multiple sources
  ✓ Anyone can make assertions about anything
  ✓ Global identities
  ✓ Distributed: Can incrementally add to others descriptions
  ✓ Fits with the WWW: The Data Web

Technically: If a statement is not asserted, then its truth-value is unknown, rather than false.

*Data:* Grass is Green.

*Question:* Is Grass Red?

*Closed World:* No

*Open World:* I Don’t Know
The Open World: Positive Use Case

Jon publishes an Annotation about part of a web page.
The Open World: Positive Use Case

Brewster archives the page … and says where it is.

✓ Without modifying the annotation at all!
The Open World: Global Complexity

- New assertions can drastically change meaning

Let’s turn this around a little…
Fred asserts:

Without modifying the annotation at all?!
The Open World: Local Complexity

- Every assertion is considered true in all contexts

Q: How do we say that Canvas 2 comes after Canvas 1, and Canvas 3 comes after Canvas 2?
The Open World: Lists

Did you think this?
Remember anyone can say anything, and it’s global…
Now there are two next links from Canvas 1, and our list is ... a graph.
Use Case: Manuscript has different page order at different times
ORE introduces proxy nodes, as not just order is local.
Eg may wish to cite a resource in the context of a set of resources.
The Open World: Lists

Shared Canvas uses multiple classes and the rdf:List construction. Serializations hide the list’s anonymous nodes.
The Open World

✓ A Single Global Graph that everyone contributes to
  ➢ Local constructions are complex
  ➢ Remote assertions can change understanding

Mitigating Factors:
  • Local identity for local context is good practice
  • Harder to take short cuts, forces understanding
  • Some grass is red. Could be both red and green
  • Trust is required in all data, not just linked data
Ontologies and Identities

✓ Shared ontologies increases semantic interoperability
  ✓ dc:title is ‘name’ or ‘label’, not property title or Dr.
  ✓ Re-use of semantics makes it easier to build applications
  ✓ Communities can develop own ontologies independently (as opposed to microdata/schema.org)

✓ Shared Identity makes it possible for graph to merge serendipitously
  ✓ Everyone can mint own IDs using http URIs
  ✓ By reusing ids, graphs will merge, creating new knowledge
Ontologies and Identities

➤ “The nice thing about … Ontologies … is that there’s so many to choose from”
➤ Far far too many to choose from, hard to find the right one
➤ If almost right, do you reuse and hope for the best, or specialize and create yet another ontology?

http://xkcd.com/927/
Ontologies and Identities

HOW STANDARDS PROLIFERATE:
SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.

SITUATION:
THERE ARE 14 COMPETING STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES. YEAH!

SITUATION:
THERE ARE 15 COMPETING STANDARDS.

BIBO
Dublin Core Elements
Dublin Core Terms
RDA
MARC21RDF
PRISM
NSDL
...

BibFrame
Ontologies and Identities

“The nice thing about … Identities … is that there’s so many to choose from”

- Far far too many to choose from, hard to find the right one
- As anyone can create identity for anything, they do
- Identity can have a contextual component – does LANL’s identifier for Oppenheimer differ from DBPedia’s?
Ontologies and Identities

✓ Shared Ontologies increase Interoperability
✓ Shared Identifiers make the graph merge
➢ Multiplicity of Ontologies
➢ Multiplicity of Identities

Mitigating Factors:
• Assertions of equivalence are just assertions. Can apply same parsers, trust mechanisms etc.
• Well-known ontologies and identifier schemes
• As the global graph continues to increase, winners will become obvious
The new JSON-LD format is actually pretty good?

```json
{
    "@context": "http://www.w3.org/ns/oa.json",
    "@id": "http://example.org/ann01",
    "@type": "oa:Annotation",
    "annotatedAt": "2012-11-10T09:08:07",
    "annotatedBy": {
        "@id": "http://public.lanl.gov/rsanderson#me",
        "@type": "foaf:Person",
        "name": "Rob Sanderson"
    },
    "hasBody": {
        "chars": "This... is CNN."
    },
    "hasTarget": "http://www.cnn.com/"
}
```
Serializations

- WAY Too many serialization formats

- The recommended RDF/XML is absolutely terrible
  - “RDF/XML was the Semantic Web’s 3 Mile Island incident”

- Multiple formats means multiple identifiers for descriptions (one per format)
  - Content Negotiation is a pain
  - Not everyone implements every format = interop hell

- Leaves room for competing models/syntaxes
  - Microdata, Schema.Org etc.
Serializations

✓ JSON-LD  (but is it just the 15th serialization?)

➢ Everything else

Mitigating Factors:
  • Software libraries help, but are inconsistent
  • … and that’s all I’ve got!
RDF and Semantic Technologies rapidly improving

- Open Source and Commercial
- Much better scalability
- Some cross-platform APIs
- In use in Fortune 500 companies
  - Internally for business intelligence
  - Externally, eg Google Freebase
- QuadStores rather than TripleStores
- Inferencing becoming real
- People working on User Interfaces
Linked Data

✓ Provides practical implementation advice
  ✓ … which people have implemented
  ✓ Used to be too much rope to know where to start tying the noose, now it’s easy!

Linked Data:

✓ Give everything a URI
✓ Make them HTTP URIs
✓ Return a description of the thing from the HTTP URI
✓ Link to other things
✓ Make the description RDF
  (cough RDF/XML mumble moving right along…)
HTTP Status 303 Hijacking

- Just took it and asserted a bunch of extra semantics
- Arguably (and seemingly indefinitely so) …
- … unnecessary with other, easier patterns
Linked Data: Description Headaches

- Extent of Description?
  - Only information about resource with URI, and “follow your nose” to information about other URIs
  - But what about URIs you don’t control but want to describe?
  - And URIs that are information resources, like data or images that have a representation already?

- Increases number of HTTP requests
  - Each of which has a very small response

- HTTP Headers (e.g. for Content Negotiation) are a pain
Temporal Issues

- Resources change over time
  - Reality, Data and Ontologies
  - Neither document nor data web has a solution
  - Need to remain in sync in distributed environment
  - New URIs for every version doesn’t work
  - Coherency: does assertion still apply when other dataset changes?

Mitigating Factors:
- Memento
- ResourceSync
Summary

• **Graphs** are powerful structures, and the complexity can be managed by tools
• **Open World** introduces complexity, but enforces best practices. Without it, would be just data on the web
• **Trust** is required for reuse of any data, not just RDF
• **Winners will emerge** for competing ontologies and identities and better than the alternative.
• **JSON-LD** is very strong, and parsers exist in all common languages for all serializations
• Good people are working on the Temporal issues ;)

Thank You!

Slides:  
http://www.slideshare.net/azaroth42/  
rdf-resource-description-failures

Open Annotation:  
http://www.openannotation.org/  
http://www.w3.org/community/openannotation/

Shared Canvas:  
http://www.shared-canvas.org/

Rob Sanderson:  
rsanderson@lanl.gov  //  azaroth42@gmail.com  
azaroth42  
http://public.lanl.gov/rsanderson/