Yahoo! Semantic Web in Production

David Beckett

Yahoo!

David Beckett - who am I?

- **Free software / Open Source developer**:
  Redland, Raptor, Rasqal, Flickcurl, Debian, …
- **W3C Standards editor and participant**:
  RDF, RDF/XML, Turtle, SPARQL, GRDDL, RDFa, …
- **Metadata / RDF / Semantic Web supporter**:
  Planet RDF, SWIG, …
- **Software Architect at Yahoo! Inc**:
  HTTP Web Services, REST Web APIs, Metadata formats, Design, Code, …

http://www.dajobe.org/

Overview

- Yahoo! media sites content and metadata problems
- Semantic Web as solution
- Semantic Web in production

This presentation is not about…

SearchMonkey

http://developer.yahoo.com/searchmonkey/

Peter Mika, *Making The Web Searchable*

Wednesday 2:45pm, Semantic Search track

This presentation is about…

- Looking at some Yahoo! media websites problems with content and metadata.
- Wanting to building these sites (properties) both better and faster.
- Showing how using semantic web technology has helped this.

(Some) Yahoo! media websites Content and Metadata problems

Terminology

- Content:
  - The text, images and data items that are used to build web pages
    (this is still a subset of all content)
- Metadata:
  - Annotations, descriptions and relationships between content items.
- Media Sites (MS)

Types of content: providers, partners and Yahoo!

- Documents: news articles, recipes, reviews
- Streaming content: video, music
- Images
- Data: Finance stocks, Sports statistics
- Descriptions and relationships: Movies, TV
- Lists: feeds, “top 10s”, playlists
- Most are/contain semi or structured data.
- Large amounts of content and underutilized

Types of content: User generated content

- Web pages, blogs, images, videos, …
- RSS/Atom feed content
- Reviews and comments
- Ratings
- Tags

Mostly markup or lightweight semantics.

Some problems with legacy Implementations

- Content was too scattered and hard to connect/relate.
- SQL used to encode specialized domain knowledge
- “Private” application-specific concepts in databases
- Silos of content that were not discoverable
- Custom stores and custom metadata
- Remixing content to create special websites was expensive, slow and not scalable.
- Many of them had weak conceptual models

Applying XML to these problems

Web services existed that generated XML, however:

- Each one had a different format or data model.
- This is not scalable or connectable: "mashups" require lots of pre-knowledge.
- XML trees cannot be easily composed.
- XML Schema (XSD) does not aid this problem.

Conclusion: XML alone does not help

(However, see Yahoo! Open Strategy (YOS) for another approach to this)

Applying Search to these problems

Sites (in the large) approximately take stored content and render to markup such as HTML:

- This step is hard for machines to reverse.
- Search technology attempts to do this over web page content
  (Search Monkey + RDFa will help here!)
- Search systems are focused on item/entity retrieval, not relationships, not metadata

Applying Markup to these problems

- Such as semantic markup, Microformats or *insert-your-favourite-markup-here*
- At least the rendering-to-HTML has some indication of where the data went.
- However each format has it's own data model (except RDFa provides a meta-microformat :)
- Which again can be connected up only with pre-knowledge.
- Plus you have to discover microformats and that means hard-coded rules (but GRDDL fixes this)

W5: Things people actually care about

- Who: people, celebrities, actors, politicians, CEOs
- What: current events, products, albums, movies, TV shows, companies
- Where: places, areas, countries
- When: this minute, hour, day, …
- Why: topics, subjects, categories

Semantic Web:
It's all about:
Enabling people
to connect their stuff

People need concepts for their stuff

- Web pages encode the things (concepts) that people care about
- Conceptual modeling needs to be *separate* from web presentation
- This is **not** news to anyone working with semantic markup, microformats or more formal representations.

Yahoo! media sites content platform

Solution:

- A new content platform for media sites
- Flexible:
  to wrap legacy systems
- Modular and extensible:
  to build new systems part-by-part

Content and metadata vision

To enable:

- *Sharing* of content domain knowledge
- *Enriching* content with metadata
- *Discovering* content people care about
- *Changing* content and metadata continuously

Content objectives

- Provide value-added content and metadata services
- Make it easy for webdevs to deploy and for editors to use
- Improve content publishing to aid discovery, sharing and (re-)use
- Eliminate duplicate efforts

Content architecture

- Distributed component architecture using web technologies
- Shared content stores (repositories) for major content types
- Property owned content stores for editorial content
- All content identified by URIs
- Web APIs for access, storage, query…

Metadata architecture

- Metadata for content identified by URIs
- Relationships between content items
- Descriptions of categories about content
- Descriptions of concepts in content
- Lightweight content metadata: e.g. tags
- Flexibility to extend metadata schema

Technical architecture: acronyms

- Unambiguous names for *resources*: **URIs**
- Common data model to access and describe resources: **RDF**
- Access to that data: **HTTP**
- Data formats: **RDF, XML, RSS, Atom, …**
- Vocabularies and constraints: **RDFS, OWL, PRISM, DC, …**
- Business rules and logic: **OWL**, ***Rules***

(There are many other supporting systems and technologies not mentioned here)

Technical architecture: HTTP serving

To make easy to use web systems:

- **REST** architectural-style: "*HTTP Web Services*"
- Resource-based content identifier URIs
- CRUD with GET / PUT / POST / DELETE
- Test and debug in the browser, with curl(1), …
- Web tech is available in all systems
- Web tech interoperates

Technical architecture: Semantic Web

- RDF: describing open relationships of distributed resources
  - Open vocabulary: flexible and extensible
  - Open-world data model: assumes change
  - Open standard: W3C royalty-free
- SPARQL query language
- OWL for constraints (and inference later)
- (Built in Redland + PHP + MySQL)

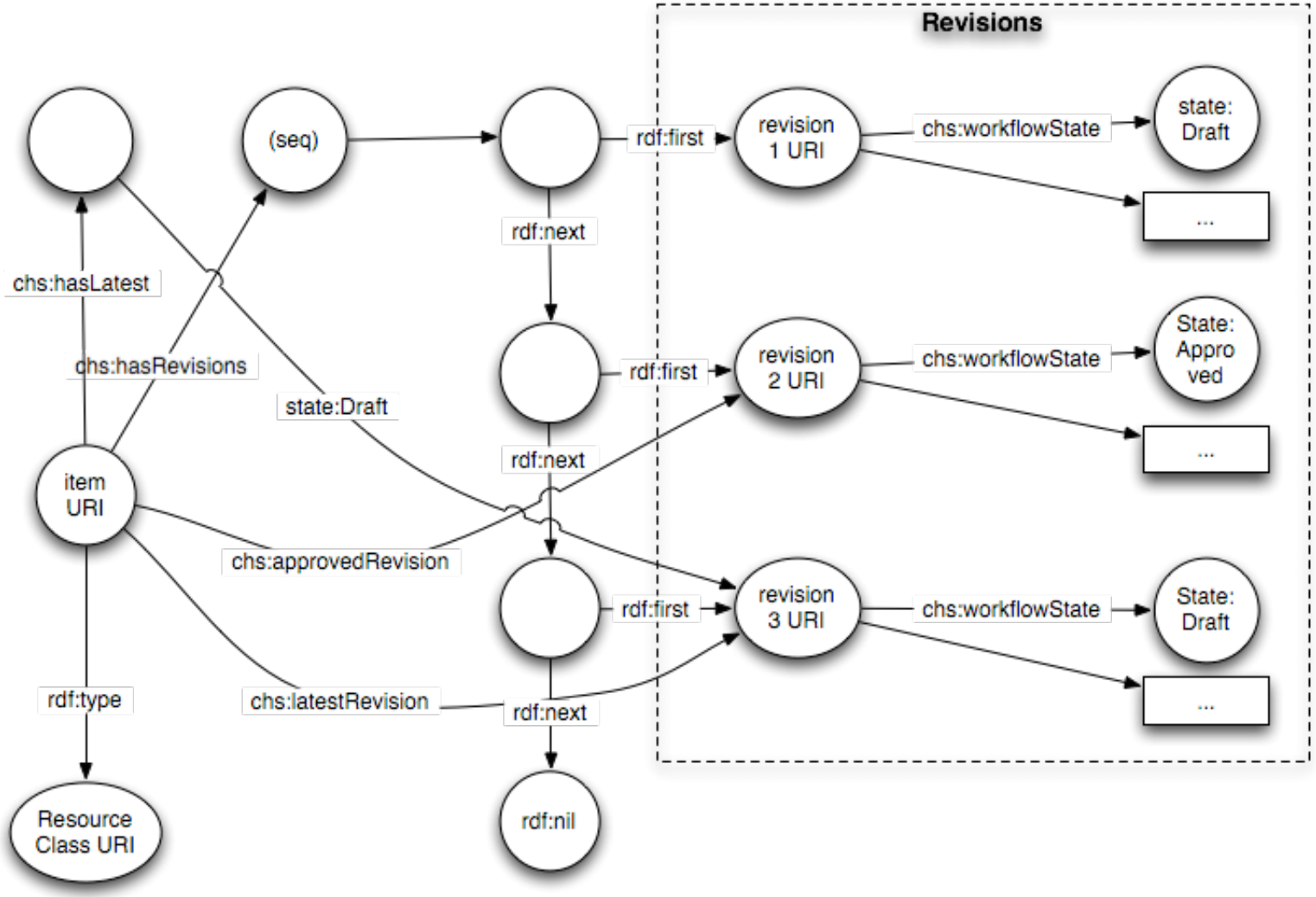
Why Semantic Web technology?

- Appropriate for describing relationships of distributed resources with web technology
- Open schema/world encourages and expects change
- Easy to add value via new metadata annotations and additions
- Resource-based web metadata aligns with REST-based web services
- Prevents encoding domain knowledge into a mostly fixed and brittle database schema

Why Semantic Web technology? Opportunities

- Semantic Web technologies are:
  - uniquely flexible in representing metadata and relationships.
- Yahoo! has:
  - lots of content with potent untapped relationships representing valuable business possibilities.
- Therefore can use these to both:
  - increase monetization and decrease costs
  - enable novel new features

Obligatory RDF Graph



Just in case you think this is all done with RDF: No

Many other technologies are Involved

- Technologies include: Text search, Content enrichment, NLP
- Protocols and formats including: HTTP, HTML, XForms, XML, RSS, Atom, JSON, HTTP Caching/Proxy
- Software and systems: primarily PHP, Java and Perl with C/C++; Squid

Content enrichment tech + Semantic Web Tech

- Enrichment technologies enable identifying of concepts in text.
- Semantic web technologies enable connecting, describing and annotating the concepts
- Together they enable a rich descriptive graph to be formed mixing structured and semi-structured content.

Search tech + Semantic Web Tech

- Semantic web technology gives search technology some good hooks to index and query on

but I'm not talking about that here…

Semantic Web In Production

Some current Yahoo! users of this technology

- US Finance
- US Food
- US Green
- US Health
- US Kids
- US Movies
- US News *3
- US Pets
- US Shine
- US TV
- CA Travel
- CA Finance
- CA Lifestyles
- US Finance
- Lat. Am. Sports

Issues: #1 tradeoff

The choice:

1. Flexibility and functionality
2. Performance

Pick **one**.

> We started with functionality then worked on improving performance

Issues: Serving at Yahoo! scale

- Application business logic: PHP5
- Storage: Redland PHP/C to MySQL
- MySQL scaling is well understood
- HTTP GET **read** with HTTP acceleration via squid
- HTTP PUT/POST writes harder to scale
- Separate read and write HTTP traffic to scale reads at a faster rate

Issues: Scaling triple stores

- Write performance:
  must change multiple triples at one go
  named graphs
- Read performance:
  may need to index important triples separately
- Size:
  today, we are comfortable with current MySQL schema

Issues: Ensuring integrity

- Changes to graphs must be atomic, consistent to ensure integrity
- Write integrity: must change multiple triples at one go
- *named graphs* are needed for graph update management

Issues: Education

- Learning RDF and OWL can be a challenge
- The lower-level tools, APIs and libraries are solid
- Domain expert user / developer tools are still young and/or weak
- Explaining data models is the hardest part
- It can take a long time for people to go "*aha*"

Benefits: multiple forms of content context (metadata)

- Topic / subject / category
- Entity / concept
- Automatic analysis: clustering, interestingness, …
- Editorial and user annotations (tags)
- Hierarchies and taxonomies
- Geo-location: http://developer.yahoo.com/geo/
- Temporal information
- “aboutness”

Benefits: improve web sites

- Lowers time to produce specialized web sites
- Enables new search possibilities
- Builds on and add value to the existing content: **Semantic Mashups**
- Improves packaging of content to enable better user web sites experiences

Benefits: revenue

- Provides better hooks for personalization and relationships.
- When you know more about your data
  => better targeting
    => better monetization

Benefits: when data = metadata

- Sometimes data and metadata are not distinguishable
- The RDF model of representing them in a linked fashion allows this situation to be handled and queried as one.

Partially similar things that came along later

- Thomson Reuters' Open Calais
  - Enrichment & RDF
- Metaweb's Freebase
  - Flexibility in schema data in an "RDF like" manner

Conclusion

Semantic Web Technology at Yahoo!

- Works at scale
- Brings tangible benefits

Thanks!

Questions?

Dave Beckett

http://www.dajobe.org/

dave@dajobe.org

dajobe@yahoo-inc.com