REST and linked data
A match made for domain driven development?

Kevin Page
Oxford e-Research Centre, University of Oxford, UK
Electronics & Computer Science, University of Southampton, UK

WS-REST 2011, Hyderabad, 28/03/2011
Context

- Building systems and tools for e-Science and e-Research in several domains
- All could be considered data-centric (though that's not to forget method)
  - Computational musicology, Music Information Retrieval
  - Geographers, oceanographers
  - Scientific workflow (bioinformaticians etc.)
Context (continued)

- Common requirements
  - Structure information for the domain
  - Expose data for use (and re-use)

- Have had some success with
  - RESTful APIs
  - Linked Data
• Common requirements:
  • Structure information for the domain
  • Expose data for use (and re-use)

• Have had some success with
  • RESTful APIs
  • Linked Data
  • *But not necessarily at the same time (why?)*
Commonality
Commonalities

• The Primacy of Resources
  • *Identification of resources is the key abstraction in REST and RDF where it is also the means to express relationships*
Commonalities

• The Primacy of Resources
  • *Identification of resources is the key abstraction in REST and RDF where it is also the means to express relationships*

• Linking is not optional
  • *Links to other URIs to discover more things (Linked Data); and as the engine of application state (REST)*
Commonalities

• The Primacy of Resources
  • *Identification of resources is the key abstraction in REST and RDF where it is also the means to express relationships*

• Linking is not optional
  • *Links to other URIs to discover more things (Linked Data); and as the engine of application state (REST)*

• Segregation of Semantics
  • *Semantics have their place (and it's not in the resource addressing/URIs)*
Adaptability

• Both approaches can evolve over time
  • REST: state transitions can be changed by modifying the links returned by representations

• Linked Data: assertions about the same resource can be made at different times, in different places, using different ontologies
Adaptability

• Both approaches can evolve over time
  • REST: state transitions can be changed by modifying the links returned by representations (modifying the hyperstructure)
  • Linked Data: assertions about the same resource can be made at different times, in different places, using different ontologies (modifying the hyperstructure)
Differences
Differences or Complementarity?
Model or API

• What purpose are the commonalities put to?

• Resources and their relationships are used to:
  • REST: identify data and transition to other resources; the means to develop an application; an API
  • Semantic Web: encapsulate the underlying data model; move to more data related using the model
Domain Driven Design

• Both the information model and API design are driven by the domain requirements

• This focuses differentiation and complexity where it should be: around those issues specific to the domain
  • A common model can be shared between the data and the API
So...

- Are all Linked Data applications today RESTful?
- Are there lots of RESTful systems using Linked Data?
Tensions

- Are the remaining differences fundamental mismatches or artefacts of current use?

- SPARQL

- Content negotiation
  - Information and non-information resources
  - 303 overhead
In Summary

- REST and Linked Data are complementary in the domain
- An opportunity to build powerful domain centric systems with a common API and data model
- Questions?
Acknowledgements

Authors

Kevin Page, David De Roure
Oxford e-Research Centre, University of Oxford, and
Electronics & Computer Science, University of Southampton

Kirk Martinez
Electronics & Computer Science, University of Southampton

Thanks

Colleagues from the SensorGrid4Env, SALAMI, NEMA and
myExperiment projects

Funding

European Commission IST FP7-223913 (SensorGrid4Env)
JISC Digitisation and e-Content, Digging into Data (SALAMI)