14 May 2013

# OSLC ResourceShape: A language for defining constraints on Linked Data

Arthur Ryman
Arnaud Le Hors
Steve Speicher

LDOW2013, Rio de Janeiro

# Outline

- Motivation and use cases
- OSLC ResourceShape specification
- Going forward
  - Proposed RDF Validation workshop

# Motivation

- Linked Data is a great way to integrate collaborating, distributed applications
  - software development tools are a prime example
  - e.g. requirements -> test cases -> defects
- Linked Data fuses REST and RDF
  - application developers need to understand the REST API contract
  - any API specification should describe both protocol and data format
- There are many examples of data format descriptions
  - OO class definitions
  - XML schemas
  - relational DDL
- How do you describe an RDF graph?
  - a graph is just a set of (possibly disconnected) triples
  - not necessarily OO
  - constraints may depend on the REST operation (GET, PUT, POST)
  - constraints may involve more than one graph (request/response, application, world)
  - it is also highly desirable to efficiently check these constraints
  - RDFS and OWL are not a solution (see Backup)

# Use cases

- Consumers of a REST service API need to:
  - prepare data that the service provider will accept
  - accept data that the service provider returns
- Service providers need to:
  - validate incoming request data
  - ensure that outgoing response data satisfies the contract
- Data format descriptions act as **metadata** for generic tools
  - generate human-readable API documentation
  - generate input forms
  - generate report definitions
  - provide user assistance for query and report builders
  - etc.
- SPARQL is a good match for expressing the semantics of constraints and implementing a constraint checker
- SPARQL is a poor match for providing **metadata** to form builders, query builders, etc.

# OSLC ResourceShape vocabulary

- Open Service for Lifecycle Collaboration (OSLC) is an organization that develops Linked Data specifications for tool integration
  - e.g. Requirements, Architecture, Configuration, Change, Quality
- OSLC experience formed basis for Linked Data Basic Profile submission to W3C
  - now under standardization by the Linked Data Platform working group
- OSLC ResourceShape is an RDF vocabulary for expressing common RDF graph constraints
  - high-level, developer-friendly way to specify typical data constraints
- The semantics of ResourceShape is expressible in terms of SPARQL ASK
  - returns **true** if constraint is satisfied, **false** if violated
  - implementers are not required to use SPARQL
- ResourceShape may be used to describe:
  - a resource (What is its structure?)
  - a creation service (What input is expected?)
  - a query capability (What properties are available?)

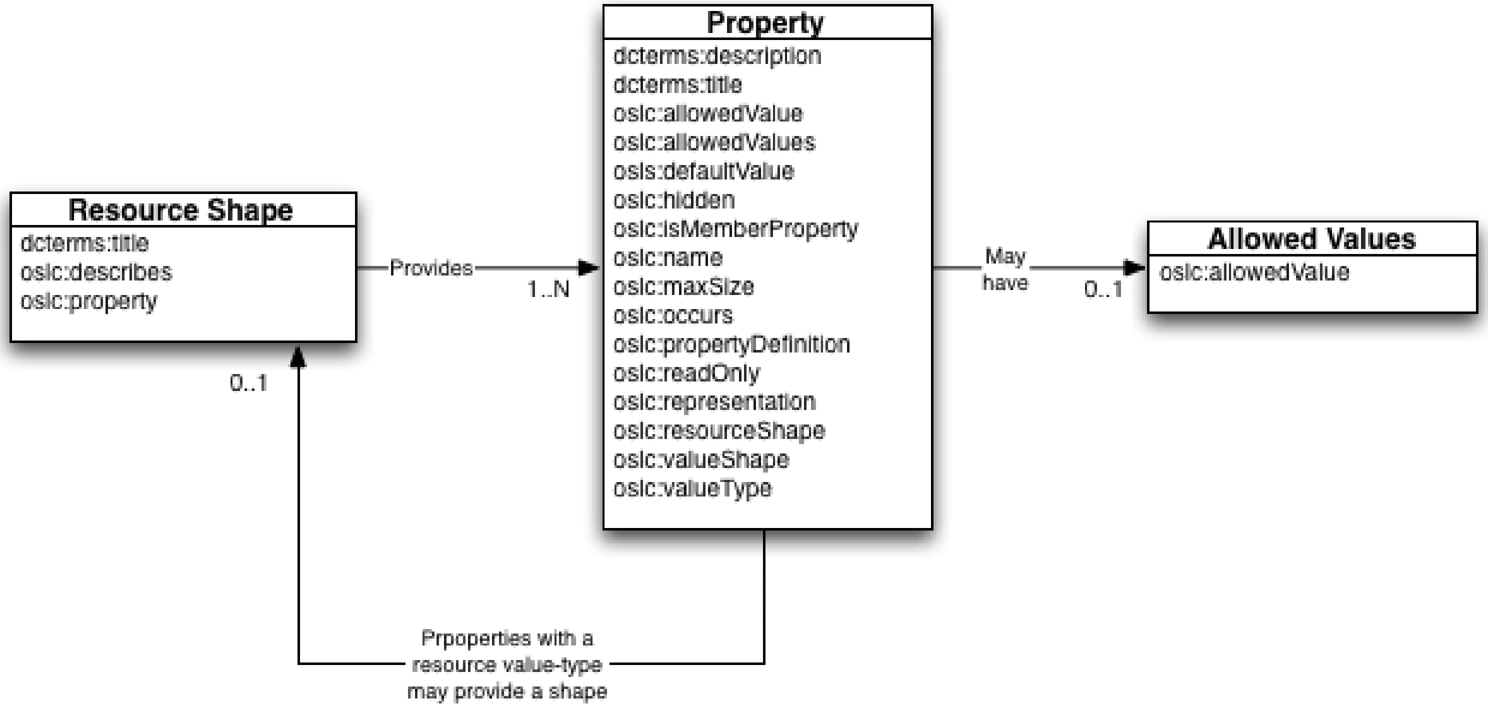
## Example ResourceShape with occurence constraints

```
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix oslc: <http://open-services.net/ns/core#> .
@prefix oslc_cm: <http://open-services.net/ns/cm#> .
@base <http://example.com/shape/oslc-change-request> .

<> a oslc:ResourceShape ;
        dcterms:title "Creation shape of OSLC Change Request" ;
        oslc:describes oslc_cm:ChangeRequest ;
        oslc:property <#dcterms-title>, <#oslc_cm-status> .

<#dcterms-title> a oslc:Property ;
        oslc:propertyDefinition dcterms:title ;
        oslc:occurs oslc:Exactly-one .

<#oslc_cm-status> a oslc:Property ;
        oslc:propertyDefinition oslc_cm:status ;
        oslc:occurs oslc:Zero-or-one .
```

# SPARQL ASK Semantics for oslc:occurs oslc:Zero-or-one example

```
prefix oslc_cm: <http://open-services.net/ns/cm#>

ask {
        select ?resource
        where {
                ?resource a oslc_cm:ChangeRequest.
                ?resource oslc_cm:status ?status
        }
        group by ?resource
        having (count(?status) <= 1)
}
```

# Overview of oslc:ResourceShape



# Going forward

- Constraint checking is part of data validation
- Proposed W3C RDF Validation Workshop
- 10-11 September 2013, Cambridge, MA, USA
- Topics for position papers may include, but are not limited to:
  - usage scenarios, e.g. data deployment or input validation.
  - schema expressivity, e.g. SPARQL ASK compared to a grammar language like XML Schema or RelaxNG.
  - schema distribution.
  - distributed validation in collaborative environments.
  - performance.
  - management of schema evolution.
- Related work (see Backup)
  - OWL Integrity Constraint Checking
  - SPARQL Interencing Notation (SPIN)

# Summary

- Linked Data application designers need a way to specify REST APIs, including constraints on RDF representations
- Generic tools for Linked Data applications need high-level metadata about the REST API
- RDFS and OWL do not satisfy this need
- OSLC ResourceShapes provides a high-level vocabulary for expressing constraints and can be implemented as SPARQL ASK
- The proposed W3C RDF Data Validation Workshop offers a forum where we can establish use cases and work towards a new standard

# BACKUP

# Why RDFS and OWL are not a solution

- Both RDFS and OWL have annotation properties
  - e.g. rdfs:label, rdfs:comment
  - useful for describing data
- However, the bulk of RDFS and OWL are about inferences, not constraints
  - e.g. rdfs:domain, rdfs:range
  - Reasoners use these interference rules to add more triples to a graph
- OWL reasoners will go to great lengths to reconcile apparently inconsistent data
  - Non-Unique Naming Assumption
  - Open World Assumption
- OWL reasoners use inconsistency to approximate constraint checking in some irreconcilable cases
  - e.g. x owl:differentThan x .

## OWL infers: ex:Joe owl:sameAs ex:Bob .

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix ex: <http://example.org/ns#> .

ex: a owl:Ontology .
ex:ChangeRequest a owl:Class .
ex:Owner a owl:Class .
ex:hasOwner a owl:ObjectProperty, owl:FunctionalProperty .

ex:Joe a ex:Owner .
ex:Bob a ex:Owner .
ex:MyRequest a ex:ChangeRequest ;
        ex:hasOwner ex:Joe, ex:Bob .
```

# Related work

- OWL Integrity Constraint Semantics, Clark & Parsia
  - – proposes alternate, non-inferencing semantics via a different import mechanism
  - – similar motivation and use cases
  - – *dual semantics may cause confusion*
  - – *as an OWL extension, it brings in all the complexity of OWL*
  - – *not graph oriented*
  - – *not REST operation oriented*

- SPARQL Inferencing Notation (SPIN), TopBraid
  - – express constraints on RDF classes as SPARQL Ask queries
  - – very flexible, expressive, and powerful
  - – *requires SPARQL for implementation*
  - – *not high level*
  - – *not graph oriented*
  - – *not REST operation oriented*