Identity Crisis in Linked Data

Ora Lassila (Nokia)
Ryan J. McDonough (Nokia)
Susan Malaika (IBM)

Position Statement for the W3C Workshop on
Linked Enterprise Data Patterns
2011-12-06

IBM

NOKIA
Connecting People

Background: Web Architecture and Identity

- Ability to identify things (resources, objects, concepts, etc.) is a cornerstone of the Web Architecture
- Identity via URIs is critical for the Semantic Web and Linked Data
- W3C's guidance is clear: All things "worth talking about" deserve to have a URI as a global, unambiguous identifier
- But, problems exist, all related to URIs as identifiers:
  1. identity vs. location
  2. missing or ambiguous identity
  3. versioning of data and identity
  4. lack of stable identity
- More guidance is needed…

Problem #1: Identity vs. Location

- Two mutually confusing observations:
  - URIs are used to provide **identity** for resources
  - URLs are used as **queries** (= "locators")
- Many different query URLs can "yield" the same resource
  - (i.e., a resource whose identity is a particular URI)
- Particularly difficult with REST
  - identity URI often "embedded" in a query URL
- Protocol URIs confuse the hell out of Web developers

Problem #1: Identity vs. Location
URL Query with Internal Identity

http://a.com/things?id=13

- Exposes an internal identifier (often a primary key in some database) – identity (as a URI) now tied to a particular domain or server
- There may be practical reasons why developers prefer internal identifiers in databases
  - no tie-ins with servers/domains
  - no perceived need of "external" linking
  - efficiency…?

Problem #1: Identity vs. Location
"RESTish" URL Query

http://a.com/things/13

- As an identity URI, people typically do not treat this as opaque
- The identity is still internal, but is now embedded in the URL as a path parameter

Problem #1: Identity vs. Location
URL Query with URI Identifier

http://b.org/things?id=http%3A//a.com/foo/13

- Makes sense, but does not look particularly “pretty”

Problem #1: Identity vs. Location
URL Query with URN Identifier

http://a.com/things?id=urn%3Acom%3Aa%3Athings%3A13

- Still not so pretty, but makes sense

Problem #1: Identity vs. Location
*URL Query with URI Identifier **(same domain)***

http://a.com/things?id=http%3A//a.com/foo/13

- Confusing to many as it does not appear to make sense
- Has the perception of being somehow inefficient

Problem #1: Identity vs. Location
Separating Identity from Location

- Web Developers have a difficult time with protocol URIs
- If a URI starts with “http”, it must be dereferencable

- NB: It seems that the new JSON specs (json-schema, json-ref, etc.) do not even attempt to make the distinction between identity and location

Problem #2: Missing or Ambiguous Identity

- Many "real-world" objects do not have a URI, but they can be uniquely identified via some attribute
  - e.g., SSN for people living in the US
  - use via `owl:InverseFunctionalProperty`, but this implies the need for some kind of reasoning services
- Some attributes normally thought of as unique in fact are not
  - e.g., ISBN numbers for books
- A set of attributes can uniquely identify an object, but is this a real identity or merely a query?

Problem #3: Versioning of Data and Identity

- W3C has largely ignored versioning
  - certainly true of versioning of data
  - sometimes also true of specifications (e.g., RDF)
- Should version information be part of the identity of an object?
- Related to “lifecycle” issues
- Note that recent SQL standardization has added features that may help in resource (data) versioning

Problem #4: Lack of Stable Identity

- W3C proclaims that "cool URIs do not change"
  - (but in reality we know they do…)
  - actually, we see a confusion between URIs and URLs here
- Mitigating lack of stability via **redirection** (or "address resolution")
  - PURLs tied to network access, not database access
  - lifecycle issues might benefit as well
  - "hash vs. slash", httpRange-14, …
- "Local" URLs are not globally unique and lend themselves to ambiguity, errors and confusion
  - e.g., file:///C:/My Documents/Resume.doc
- With reasoning support, `owl:sameAs` could be used as a means of declaring mappings (some caveats apply)

Conclusions

- Confusion in matters of identity leads to a lack of interoperability
- Consequently, this delays and hinders the deployment of Semantic Web and Linked Data systems
- We would like to alleviate the confusion, and are seeking for discussion, guidance, etc.

