Revelytix
Emergent Analytics

Ontology Integration

SPARQL Federation and SWRL Rules

Alex Hall

Mike Lang Jr.

May 18, 2009

Agenda

Revelytix
Emergent Analytics

- Overview of semantic technologies – RDF, OWL, and SPARQL
- Introduction and overview of ontology integration using OWL equivalence properties
- Discussion of the use of SPARQL queries to access and join the federated information
- Introduction to SWRL rules as a means of integrating information between ontologies

Semantic Technology Overview

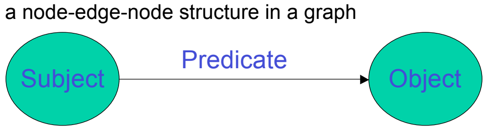
- RDF – data model, serialized in a variety of formats, including XML
- URI – universal identifiers
- SPARQL – query language for RDF data
- OWL – language for defining ontologies; schema layer
- SWRL – language for representing rules

RDF (Resource Description Framework)

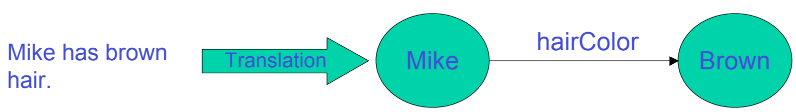
"The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web."

- W3C RDF Primer

- RDF enables the description of things (resources) in the form of graphs
- Each RDF statement (triple) consists of 3 parts: subject, predicate, object
- A triple forms a node-edge-node structure in a graph



Example

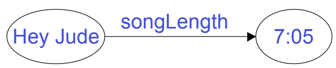


Examples

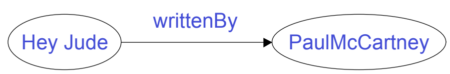
- Paul McCartney was a member of The Beatles

•Ringo Starr played the Drums

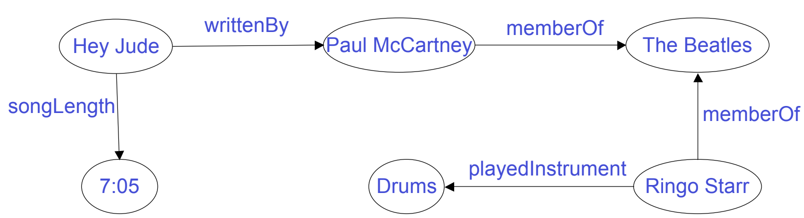
•“Hey, Jude” is 7 minutes and 5 seconds long.



•Paul McCartney wrote “Hey, Jude”



Examples cont’d – full graph



Main Benefits of RDF

Revelytix
Emergent Analytics

- **Extensibility**
  - Very limited schema design must be done prior to deployment
  - New concepts and relationships can easily be added to a graph at anytime
- **Integration – graph merging**
  - Graphs can be easily merged and then used as though they were a single graph
    - **A graph plus a graph plus a graph (…) equals a graph**

URIs (Uniform Resource Identifiers)

Revelytix
Emergent Analytics

- A URI acts as a unique identifier for a resource
  - Every resource either has a URI, is a blank node or is a literal (i.e. string, integer, date, etc.)
- Enables computers to understand when RDF statements are about the same thing or different things
  - Key to merging graphs and managing information in disparate locations
- A URI is not a name; it is simply an identifier
  - A URI should be opaque; it should not hold any meaning in
  - Examples in real life: SSN, VID

<u>Example</u>

Ringo Starr = http://www.example.com/ontology/music#person123

Aside: Namespaces

Revelytix
Emergent Analytics

- Namespaces are used to organize and abbreviate URIs
  - A prefix can be defined for a namespace

Example

PREFIX music: http://www.example.com/ontology/music#

Ringo Starr = music:person123

SPARQL (SPARQL Protocol and RDF Query Language)

Revelytix
Emergent Analytics

- SPARQL is the W3C standard language for querying RDF
  - SPARQL is also a protocol for sending queries and receiving results over HTTP
- A SPARQL query defines a graph pattern using variables
  - Matches to the graph pattern are returned as results to the query
  - Designed to be effective for distributed, ragged, unpredictable data

SPARQL Example

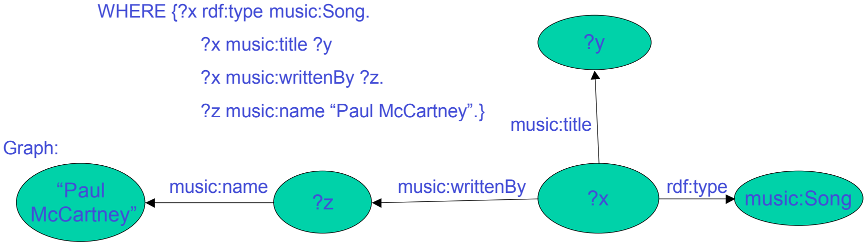
Revelytix
Emergent Analytics

Question: What are the titles of any songs that Paul McCartney wrote?

Query:

```
PREFIX  music: http://www.example.com/ontology/music#
SELECT ?x ?y
WHERE {?x rdf:type music:Song.
       ?x music:title ?y
       ?x music:writtenBy ?z.
       ?z music:name “Paul McCartney”.}
```

Graph:



Result:

x	y
music:song123	“Hey Jude”

SPARQL - Advanced Features

Revelytix
Emergent Analytics

- OPTIONAL
  - Lets you specify statements in your query which will only be returned in the results if present
- UNION
  - Acts as logical-or
  - Lets you specify multiple queries and receive results if any match
- FILTER
  - Lets you further restrict the results of your query

RDFS (RDF Schema)/OWL (Web Ontology Language)

Revelytix
Emergent Analytics

- RDFS and OWL are vocabularies defined using RDF
- Together they define a standard vocabulary which is used to define vocabularies using RDF
  - RDFS is a vocabulary which can be used to define a fully functional model for describing RDF data
  - OWL is another vocabulary which extends and adds to RDFS and gives the ability to define much more complex, intricate models for describing RDF data
- Can be viewed as the schema layer in the semantic web stack
  - However, in RDF, classes, relations, and instances are all "first class" entities
  - "Schemas" are known as ontologies

OWL Ontologies

Revelytix
Emergent Analytics

- An ontology is a machine-readable description of a domain
  - Model
    - Information model, Metadata model, Logical model
  - Vocabulary
- Defines Classes, Properties, and Instances which exist in the domain and the relationships between them
  - What types of things exist?
    - Classes
  - What types of relationships exist between things?
    - Properties
  - What is the meaning of a given term?
    - How are Classes and Properties related to each other?

Classes

Revelytix
Emergent Analytics

- A class is a group of similar instances
  - i.e. Horses, Mammals, Cars, Trees, Pine Trees
- OWL enables classes to be defined based on:
  - Property restrictions which describe conditions for membership in the class
  - Logical relationships to other classes
    - Sublclass
    - Equivalent
    - Union
    - Intersection
    - Complement

Properties

Revelytix
Emergent Analytics

- Properties are kinds of relationships which can exist between to resources
  - i.e. height, weight, name, date of birth
- Properties are used to describe general facts about Classes and specific facts about Instances
- RDFS/OWL define various properties which can be used to properties based on their logical characteristics
  - i.e. subproperty, equivalent, functional, inverse, transitive

Instances

Revelytix
Emergent Analytics

- An instance represents something that exists
  - The instantiation of a class
  - i.e. Mike Lang (Person), Revelytix (Company)
- There are three basic categories of facts that are asserted about instances
  - Class membership
    - Mike Lang is a Person
  - Identity – Equivalent or Different from another instance
    - Mike Lang is the same as Michael Lang
  - Generic property assertions
    - Mike Lang works for Revelytix

Revelytix
Emergent Analytics

Ontology Integration

Introduction

Revelytix
Emergent Analytics

•When different ontologies contain facts about the same resources, we can find new and interesting relationships between other resources in those ontologies.

•Ontology integration is:

- The process of identifying common concepts and resources shared between ontologies.
- Expressing these mappings between ontology concepts and resources using a common language.
- Accessing the distributed contents of these ontologies in a manner that takes advantage of these mappings.

Benefits of Semantics

Revelytix
Emergent Analytics

•Federation of information across multiple sources is not a new problem.

•Using semantics offers distinct benefits:

- Universal identifiers
- Open world assumption
- Structure decoupled from semantics

Benefits of Semantics

Revelytix
Emergent Analytics

•Universal Identifiers:

- Resources on the semantic web are identified using globally unique Uniform Resource Identifier (URI) references; different organizations can assert facts about common resources.
- Different data sources which make statements about the same URI are, by definition, referring to the same resource.
- No knowledge of internal database identifiers or foreign keys is required to refer to remote resources.
- *Caveat:* A URI can only refer to one resource, but one resource can be referenced using multiple URIs.

Benefits of Semantics

Revelytix
Emergent Analytics

•Open world assumption:

- Never assume that any one data source contains all the information there is to know about a resource.
- Our current understanding of the world can always be augmented with new facts from new sources.
- Any application can assert information about any resource and publish it on the semantic web. Client applications decide for themselves what, if anything, to do with this new information.

Benefits of Semantics

Revelytix Emergent Analytics

•Structure decoupled from semantics:

- All information is stored using a simple, universal data structure (RDF triples)
- Don't need to know about another organization's database schema in order to work with their information.
- Information in an RDF graph is self-describing: concept descriptions are encoded as an ontology that can be stored along with the data itself.

Challenges of Information Management

Revelytix Emergent Analytics

•Requires flexibility: schemas can change, and information can be incomplete.

•Differences in terminology: organizations can (and often do) choose different identifiers to reference the same real-world concepts and resources.

•Conceptual differences: real-world concepts can be modeled differently by different organizations.

- Example: “F-22” can be an instance (when describing capabilities) or a class (when listing inventories).

Trade-offs

Revelytix
Emergent Analytics

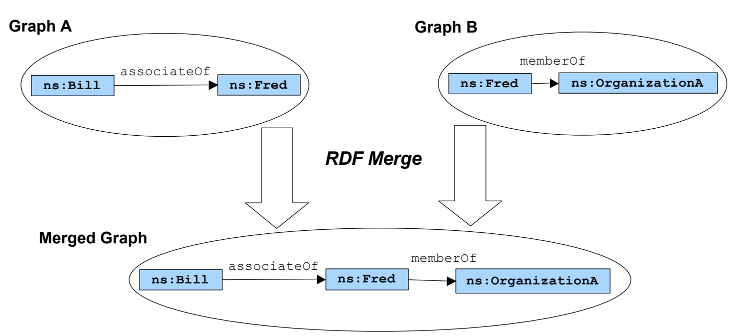
•Correctness, Completeness, Performance: Choose two.

- Correctness: Every result is a valid relationship supported by the base facts.
  - This is usually a prerequisite for most applications.
- Completeness: Every valid relationship is found for a given set of base facts.
  - This is the realm of description logics, i.e. OWL DL.
- For distributed applications, simple OWL Lite semantics evaluated with rules (backwards or forward chaining) represents a good compromise between completeness and performance.

Example

Revelytix
Emergent Analytics

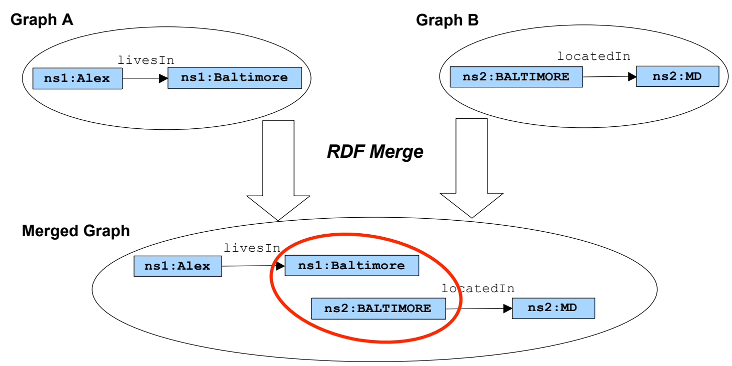
•Simple case: two ontologies asserting facts about the same resource, using the same identifier:



Another Example

Revelytix
Emergent Analytics

•Same resource appears in different ontologies, but referred to with different identifiers:



OWL Equivalence

Revelytix
Emergent Analytics

•Solution for linking different URI's which refer to the same concept or resource.

•Different types of equivalence:

- Class and property equivalence (owl:equivalentClass, owl:equivalentProperty axioms)
- Individual equality (owl:sameAs axioms)

•Asserting equivalence using properties from the OWL vocabulary allows applications to understand and find new connections involving the related resources.

- Use of the OWL vocabulary doesn't automatically mean we're using OWL inference engines.

Integration Tasks

Revelytix
Emergent Analytics

- Mapping of relationships between concepts (classes and properties) in ontologies.
    - Often done manually by an ontologist or architecture committee.
- Classification of individuals from one ontology using concepts described in another hierarchy.
    - Existing OWL reasoners are good at this.
- Identifying common individuals that appear in ontologies with different identifiers.
    - Too many to do this manually, and OWL provides only limited support for inferring individual identity.

Revelytix
Emergent Analytics

Querying Federated Graphs

Merging Graphs

Revelytix Emergent Analytics

- To find new relationships between resources in different graphs, query against the *RDF merge* of the graphs.
  - RDF merge is the logical union of the statements from multiple graphs, with special handling for blank nodes.
- SPARQL lets you specify an RDF merge as the target dataset for a query using multiple "FROM" clauses.
- Each FROM clause specifies a graph URI that contributes to the merge. These URIs are usually either:
  - An internal graph identifier.
  - The location of an external document containing RDF triples. (e.g. in RDF/XML or N3 format)

SPARQL Graph Merge

Revelytix Emergent Analytics

```
SELECT ?person
FROM <ex:graph:A>
FROM <ex:graph:B>
WHERE { <ex:Alex> foaf:knows ?x . ?x foaf:knows ?person }
```

- All existing SPARQL implementations can handle this query when graphs A and B are both internal.
- Problem: What if graphs A and B are stored in separate systems accessible only with distinct SPARQL services?
  - BAD IDEA: Download all the data and store it locally.
    - You'll have multiple copies of the data to synchronize.
    - Your data will quickly become out-of-date. This is a data warehouse.
  - Better idea: Go directly to the source.

SPARQL Federation

Revelytix
Emergent Analytics

•If the SPARQL query service knows the address and graph IRI used to access the externally stored graph, then it can include its contents in the query results:

- Triple patterns being resolved against the external graph are translated to SPARQL queries and sent to the remote service.
- Results from the remote service are appended or joined with results from the internal graphs as necessary.
- This strategy applies regardless of the native storage format for the remote graph, since SPARQL is a standard.

SPARQL Federation

Revelytix
Emergent Analytics

•Given a FROM clause containing the URI of an external graph, a query service needs to:

- Identify that the graph is stored externally.
- Find the address of the remote SPARQL service and (optionally) the graph URI needed to access the graph.

•Options for finding service address and graph URI:

- Configure them in a set of mappings internal to the local service.
- Set up an external registry service.
- Use a naming scheme for the graph URI that encodes the service location and graph name.

Federation Example

Revelytix
Emergent Analytics

•Sample query:

```
SELECT ?emp ?homepage
FROM <ex:local:HR> FROM <ex:remote:FOAF>
WHERE { ?emp hr:reportsTo hr:JSmith .
        ?emp foaf:homepage ?homepage }
```

•Expanded query plan:

```
((?emp hr:reportsTo hr:JSmith IN <ex:local:HR>)
   UNION (?emp hr:reportsTo hr:JSmith IN <ex:remote:FOAF>))
JOIN ((?emp foaf:homepage ?homepage IN <ex:local:HR>)
   UNION (?emp foaf:homepage ?homepage IN <ex:remote:FOAF))
```

•Remote queries:

```
SELECT ?emp WHERE { ?emp hr:reportsTo hr:JSmith }
SELECT ?emp ?homepage WHERE { ?emp foaf:homepage ?homepage }
```


Optimization

Revelytix
Emergent Analytics

•The more you know about the contents and structure of the federated graphs, the more efficient your queries.

•Previous example: the two graphs use different ontologies to describe their information.

- The graph containing HR information won't use FOAF properties, and vice versa.
- Use the GRAPH keyword to evaluate triple patterns against only the graph that will satisfy them.

```
SELECT ?emp ?homepage
FROM <ex:local:HR>
WHERE { ?emp hr:reportsTo hr:JSmith .
        GRAPH <ex:remote:FOAF> {
              ?emp foaf:homepage ?homepage } }
```

Challenges of Federation

Revelytix
Emergent Analytics

- Resolving patterns against very large external graphs can return lots of unnecessary data across the network
  - Compounded by the verbose SPARQL XML results format.
- Complex queries can cause a large number of remote SPARQL requests to be executed if complete results are required.
- Efficient processing of federated queries requires an advanced query planner.
  - Designed specifically for a federated environment.
  - Ideally incorporate dynamic means of retrieving statistics and other metadata from remote services.

Existing Federation Technology

Revelytix
Emergent Analytics

- Sesame RDF framework (http://www.openrdf.org/):
  - Federation SAIL is part of 3.0 alpha release
- Jena (http://jena.sourceforge.net/):
  - Extends SPARQL with a “SERVICE” keyword (similar to GRAPH) for matching query patterns against remote services.
- Mulgara (http://www.mulgara.org/):
  - Remote resolver can federate queries among remote Mulgara servers (but not generic SPARQL services).
- Other RDF stores: Oracle, AllegroGraph, OpenLink

Revelytix
Emergent Analytics

Ontology Integration with SWRL

Mapping Individuals

Revelytix
Emergent Analytics

•Our goal: Identify common resources that are shared between graphs and mark them as equal using owl:sameAs

- Too many individuals to do this manually.
- Better approach is to determine resource equality dynamically by examining properties of the resources.

•Rules in the form of *Horn clauses* are well-suited to this purpose.

- Define an antecedent (body) which is a pattern match of triples from a graph and a consequent (head) which much be true whenever the antecedent is satisfied.
- "A(x) ^ B(x) → C(x)": If A(x) and B(x) are true, the so is C(x).

SWRL Rules: W3C Submission

Revelytix
Emergent Analytics

- SWRL: *Semantic Web Rules Language*
  - De facto standard exchange format for describing rules to apply to RDF graphs
  - Antecedent and consequent are composed of *atoms* which map to triple patterns in the RDF graph.
  - The body describes triple patterns to match in the graph, and the head describes derived statements.
- Example (using an informal SWRL syntax):

The rule:

```
hasParent(?x1,?x2) ^ hasBrother(?x2,?x3) → hasUncle(?x1,?x3)
```

Applied to the graph:

```
<ex:Mary> <ex:hasParent> <ex:John> .
<ex:John> <ex:hasBrother> <ex:George> .
```

Generates the statement:

```
<ex:Mary> <ex:hasUncle> <ex:George> .
```

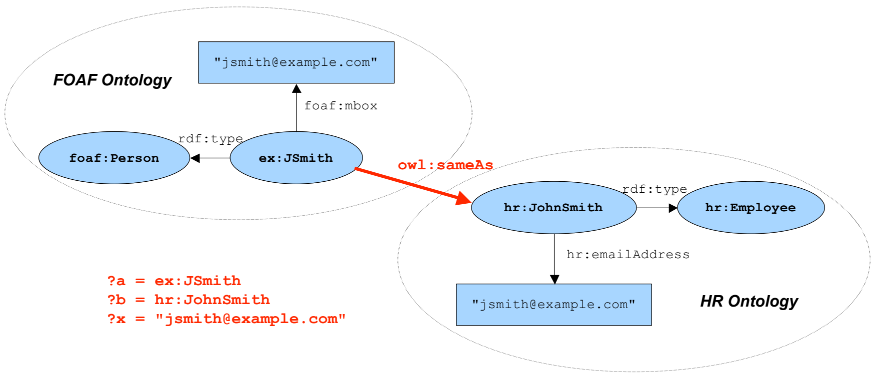

Mapping with Rules: General Approach

Revelytix
Emergent Analytics

1) Identify similar classes in the target ontologies whose instances are candidates for equivalence mapping.
   - Example: `foaf:Person` and `hr:Employee`
2) Choose criteria (usually property values) to use as the basis for determining equivalence of instances in these classes.
   - Example: `foaf:mbox` and `hr:emailAddress`
3) Express the criteria in the form of a rule.
   - Example: "If *A* is a `foaf:Person` with a `foaf:mbox` of *X*, and *B* is an `hr:Employee` with an `hr:emailAddress` of *X*, then *A* and *B* are the same individual."

Graphical Example

Revelytix Emergent Analytics



SWRL Rule:

```
foaf:Person(?a) ^ foaf:mbox(?a,?x) ^
hr:Employee(?b) ^ hr:emailAddress(?b,?x) → sameAs(?a,?b)
```


Applying SWRL Rules

Revelytix Emergent Analytics

•Once SWRL rules are written to express the equivalence mappings, several options for using them:

- Apply the rules to the merged graphs using SPARQL federation and forward-chaining, store the derived statements locally.
  - Good existing tool support for SWRL in forward-chaining rules engines for RDF graphs.
  - Existing tools need to be extended to apply rules to combinations of distributed graphs.
- Use a backwards-chaining engine to avoid having to store the derived statements locally.
- Convert the SWRL rules to SPARQL queries which can be saved and re-executed.

Using the Derived Equivalence Relations

Revelytix
Emergent Analytics

- Perform additional forward-chaining using OWL semantics to find new relationships which will be found by federated SPARQL queries.
- Use a backward-chaining engine to incorporate the new relationships into federated SPARQL queries.
- Add logic to the federated SPARQL queries to directly incorporate the OWL equivalence relationships:

```
SELECT ?person
WHERE { JSmith foaf:knows ?person
        UNION { Jsmith foaf:knows ?x .
                ?x owl:sameAs ?person } }
```