Semantic Web

User Interface & applications

Trust

Proof

Unifying Logic

Query: SPARQL

ontology: OWL

Rules: RIF

RDF-S

Data interchange: RDF

XML

URI

Unicode

Crypto
Languages

- RDF Schema
- OWL
- SWRL
- Jena Rules Language
- SPARQL

- RDF Triples are the common foundation
RDF Graphs and Triples

RDF/XML Serialization:

```xml
<MalePerson rdf:ID="John">
  <hasChild rdf:resource="#Mary"/>
</MalePerson>
```

N3/Turtle Serialization:

```turtle
:John a :MalePerson ;
  :hasChild :Mary .
```
Triple Pattern Matching

Subject: ?p

 Predicate: rdf:type MalePerson

 Object: hasChild ?c

 Subject: John

 Predicate: hasChild

 Object: ?c

 Subject: Mary
Rules & Triples

• Execution of rules infers new triples

[defineUncle:
  (?p :hasChild ?c)
  (?p :hasSibling ?s)
  (?s rdf:type :MalePerson)
  -> (?c :hasUncle ?s)]
Components of a Rule

- **Triple patterns** – like a triple, but with some named variables instead of fixed parts
  - `?company rdf:type :MajorCompany`
  - `Fortune500 :lists ?company`
  - `?company :hasCEO ?person`

- **Rule “Body”**
  - Set of triple patterns, all of which must match
  - Each variable must be ‘bound’ to the same item at every occurrence
    - `HP rdf:type :MajorCompany`.
    - `Fortune500 :lists HP`.
    - `HP :hasCEO Fiorina`.

- **Rule “Head”**
  - Set of triple patterns that will be asserted, when the body matches
  - Variables in these patterns have values that were bound in the body
Demos

• Tools
  – Protégé + JESS
  – TopBraid Composer + Jena

• Example use cases
  – Family relationships
  – Real estate business
  – Ontology Mapping
Example Scenario

- Real Estate agents
  - “Database” of available properties
  - Properties are updated continuously
  - Customers have specific search patterns
  - The rule system shall notify the agent if a matching property has been added
Design

• OWL Ontology with domain concepts
  – Real Estate Properties
  – Characteristics of properties
  – Suburbs
  – Local attractions of the suburbs

• (Jena) Rules to drive matching
Ontology Overview (1)

- Real Estate properties are located in Suburbs
- Suburbs have local attractions (Beaches etc)
Ontology Overview (2)

- We have various types of Real Estate properties
- Properties are suggested to Customers
Instance Database
Example Instance
Rule 1: Convert Currencies

• Property prices are in Australian Dollars
• Customers may ask for prices in $US

[convertAU2USDollar:
  (?p :priceAU ?aud)
  product(?aud 0.7745 ?usd)
  -> (?p :priceUS ?usd)]
Rule 2: Simple Matching

- Customer Mike Turner is looking for a three-bedroom house

```
[findMatchesForMikeTurner:
 (?p rdf:type :House)
 (?p :bedrooms 3)
 -> (:MikeTurner :suggestedProperty ?p)]
```
Rule 3: Matching

- Rebecca is looking for a property close to a shopping mall

```
[findMatchesForRebeccaSmith:
    (?p :location ?l)
    (?l :attraction ?a)
    (?a rdf:type :ShoppingMall)
  -> (:RebeccaSmith :suggestedProperty ?p)]
```
Rule 4: Classification

• Find all properties that are located in a suburb that has a beach with a swimming enclosure

[findSafeSwimmingInstances:
  (?p rdf:type :RealEstate)
  (?p :location ?s)
  (?s :attraction ?a)
  (?a rdf:type :Beach)
  (?a :swimmingEnclosure "true"^^xsd:boolean)
 -> (?p rdf:type :PropertyWithSafeSwimmingBeach) ]
Rule 5: Complex Matching

- John Doe is looking for a property with a safe swimming beach, at least 4 bedrooms and less than US$ 900,000

[findMatchesForJohnDoe:
  (?p rdf:type :PropertyWithSafeSwimmingBeach)
  (?p :priceUS ?usd)
  (?p :bedrooms ?b)
  lessThan(?usd 900000)
  greaterThan(?b 3)
  -> (:JohnDoe :suggestedProperty ?p)]
Rule Chaining

• Rule 5 depends on Rules 1 & 4

[findMatchesForJohnDoe:
(?p rdf:type :PropertyWithSafeSwimmingBeach)
(?p :priceUS ?usd)
(?p :bedrooms ?b)
lessThan(?usd 900000)
greaterThan(?b 3)
-» (:JohnDoe :suggestedProperty ?p)]

[findSafeSwimmingInstances:
(?p rdf:type :RealEstate)
(?p :location ?s)
(?s :attraction ?a)
(?a rdf:type :Beach)
(?a :swimmingEnclosure "true"^^xsd:boolean)
-» (?p rdf:type :PropertyWithSafeSwimmingBeach) ]

[convertAU2USDollar:
(?p :priceAU ?aud)
product(?aud 0.7745 ?usd)
-» (?p :priceUS ?usd)]
Rule 6: Copying Values

• Whenever something is a suggestedProperty, then we want to copy its image into suggestedPropertyImage

[copyImages:
    (?c :suggestedProperty ?p)
    (?p :image ?i)
-> (?c :suggestedPropertyImage ?i)]
Executing Rules
Browsing Suggestions
OWL DL and Rules

• Rules can be executed “on top of” DL
• DL can be implemented by Rules
# OWL vs. Rules

<table>
<thead>
<tr>
<th>OWL</th>
<th>SWRL / RIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>W3C Recommendation</td>
<td>Standard in Progress</td>
</tr>
<tr>
<td>Recent implementations</td>
<td>&gt;20 years technology</td>
</tr>
<tr>
<td>Formal decidability</td>
<td>Possibility of Spaghetti code</td>
</tr>
<tr>
<td>Restriction language highly constrained</td>
<td>Powerful pattern language</td>
</tr>
</tbody>
</table>
SPARQL

- Not designed as a rule language
- W3C Standard query language for RDF
- Triple matching
  - SELECT
  - CONSTRUCT
- “Pragmatic” rule language
Schema Mapping with SPARQL
Schema Mapping (2)
### SPARQL and RULES (SWRL)

<table>
<thead>
<tr>
<th>SPARQL</th>
<th>RULES (SWRL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex patterns with ?variables</td>
<td>Complex patterns with ?variables</td>
</tr>
<tr>
<td>Defaults, options, boolean operations</td>
<td>AND only</td>
</tr>
<tr>
<td>Filters with math</td>
<td>SWRLb built-ins for math</td>
</tr>
<tr>
<td>Run under user/program control</td>
<td>chaining opportunistically</td>
</tr>
<tr>
<td>Optimized for a single query</td>
<td>Optimized for groups of rules</td>
</tr>
</tbody>
</table>