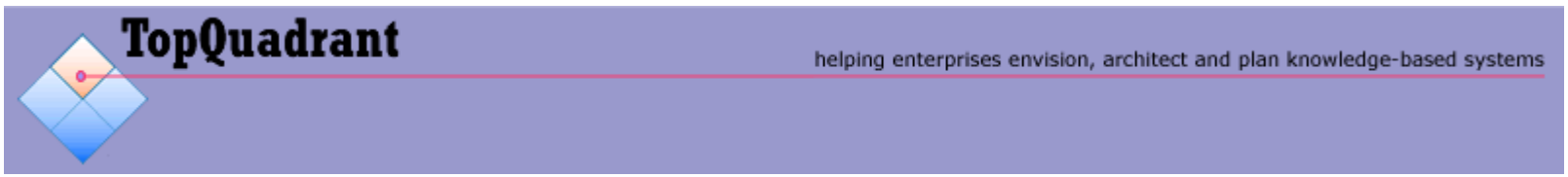
Semantic Web Rules

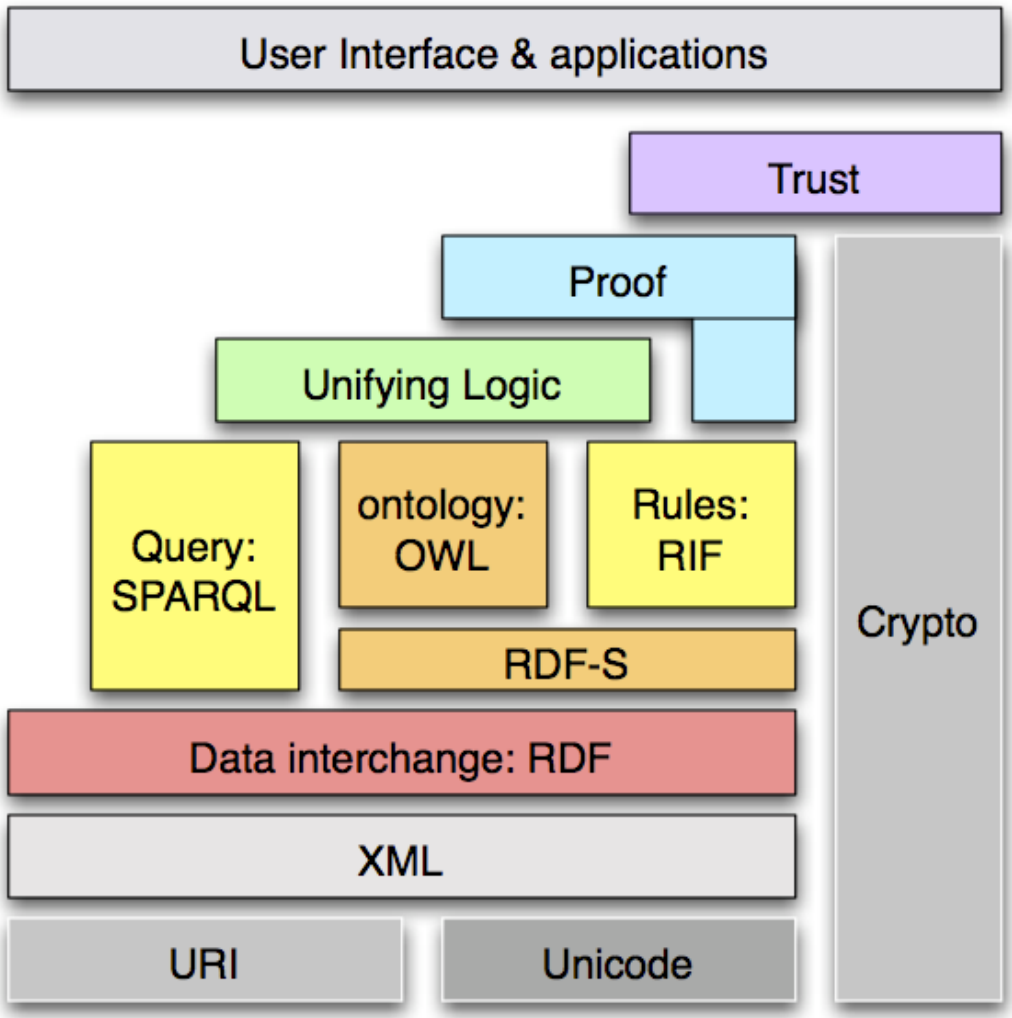
- Tools and Languages -

Tutorial at Rule ML 2006, Athens, GA

Holger Knublauch



Semantic Web

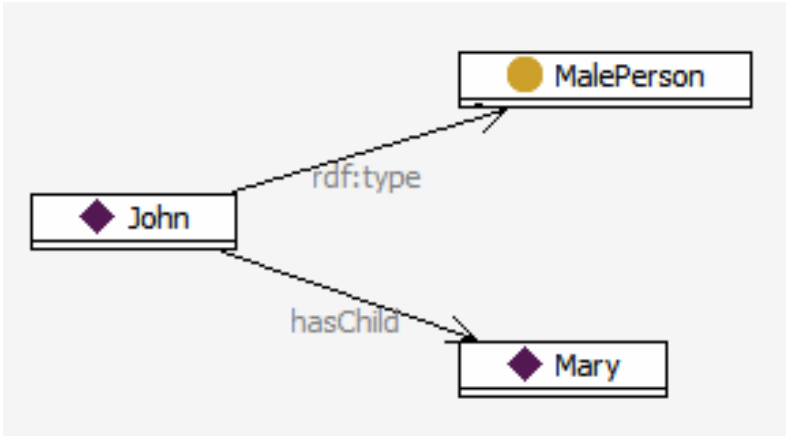


Languages

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

- RDF Triples are the common foundation

RDF Graphs and Triples



Subject	Predicate	Object
John	rdf:type	MalePerson
John	hasChild	Mary

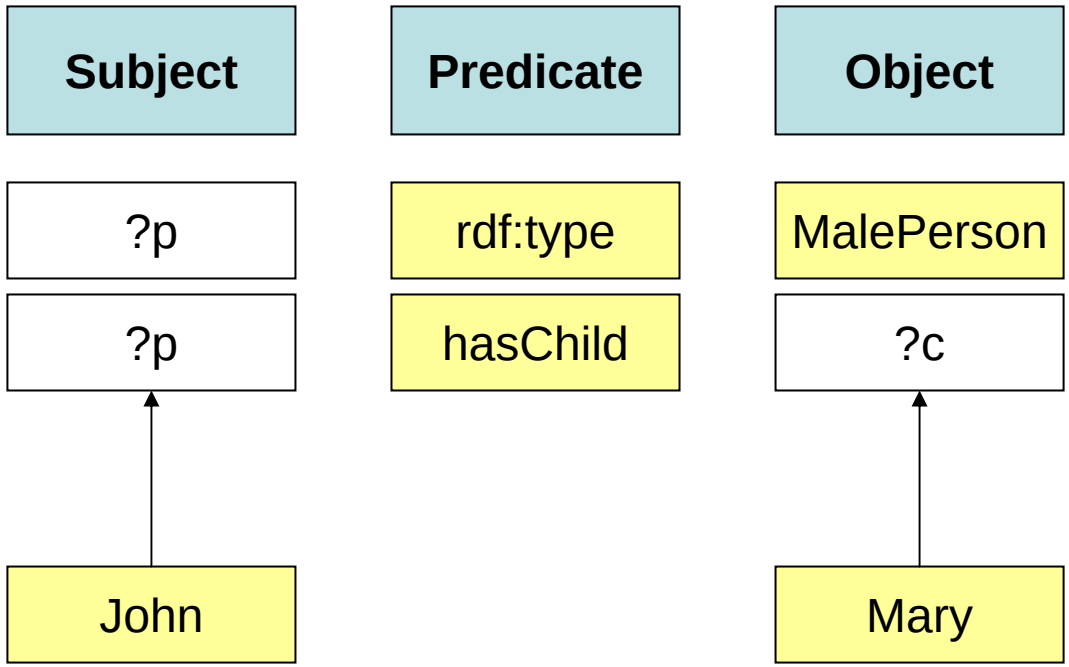
RDF/XML Serialization:

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

N3/Turtle Serialization:

```
:John  a       :MalePerson ;
       :hasChild :Mary .
```

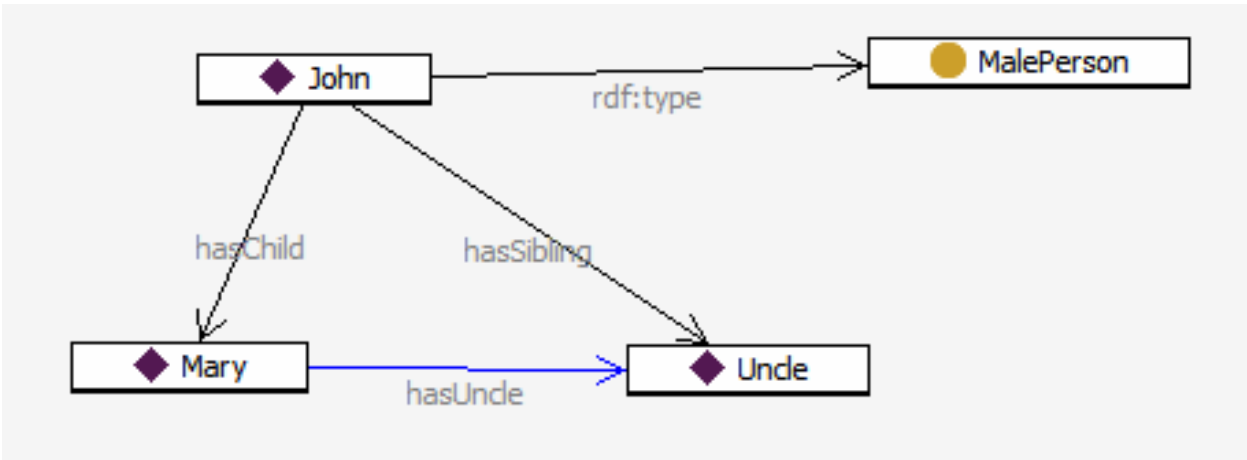
Triple Pattern Matching



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

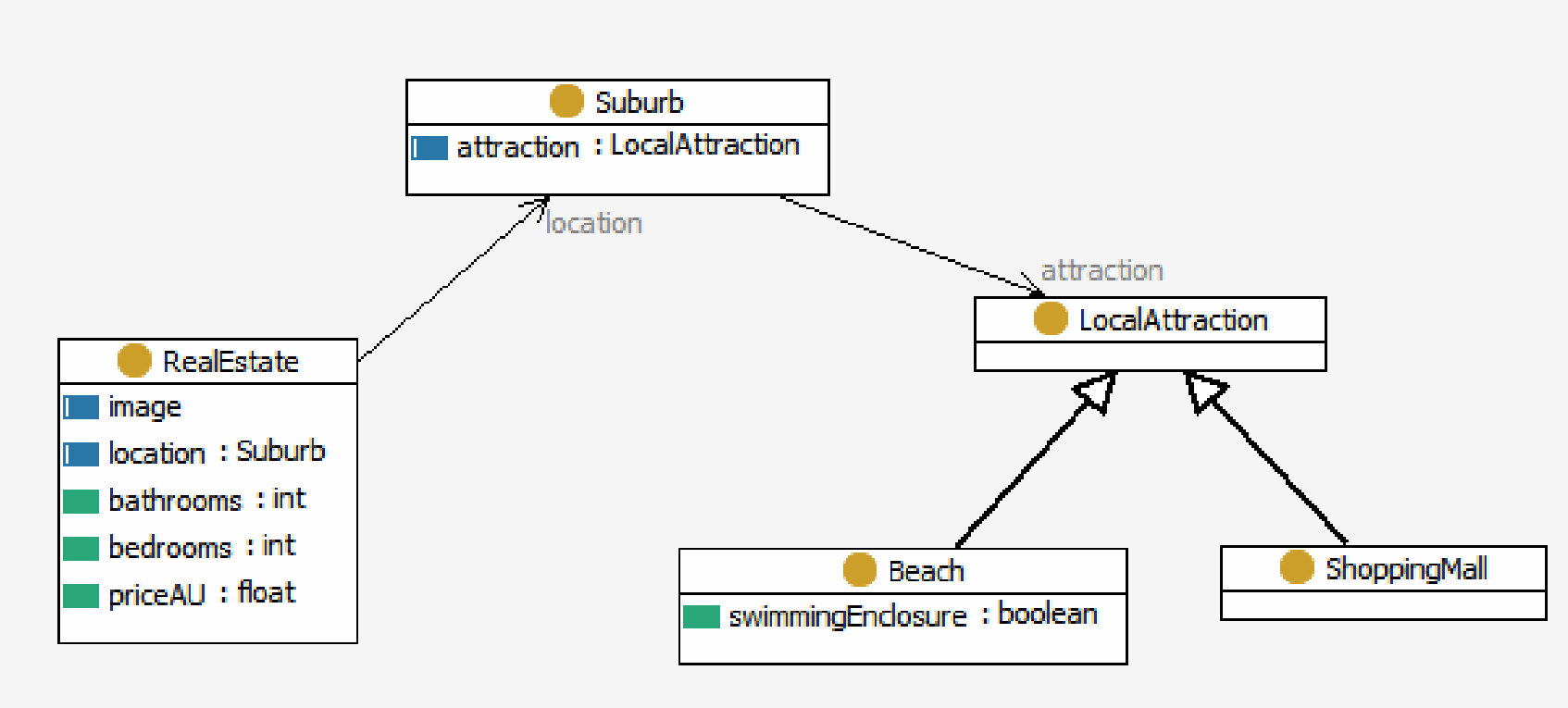
Example Scenario

- 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

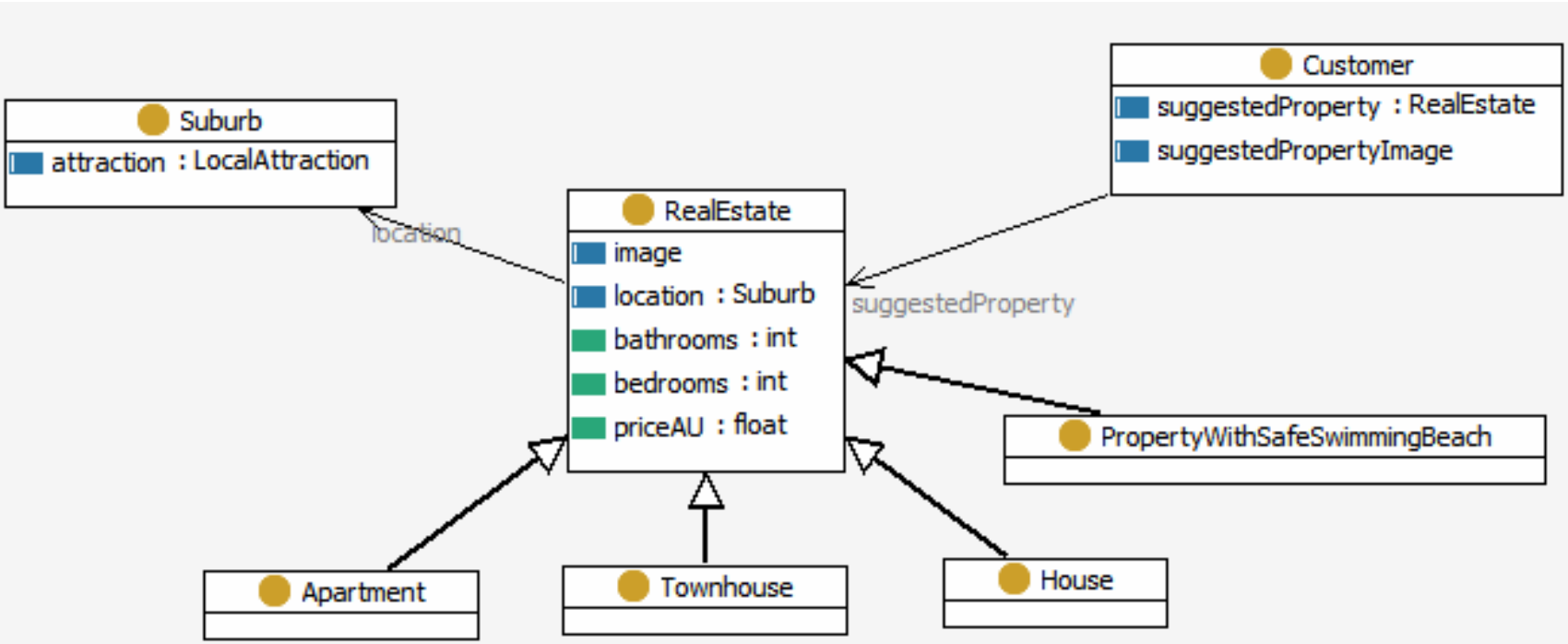
- OWL Ontology with domain concepts
  - Real Estate Properties
  - Characteristics of properties
  - Suburbs
  - Local attractions of the suburbs
- (Jena) Rules to drive matching

Ontology Overview (1)



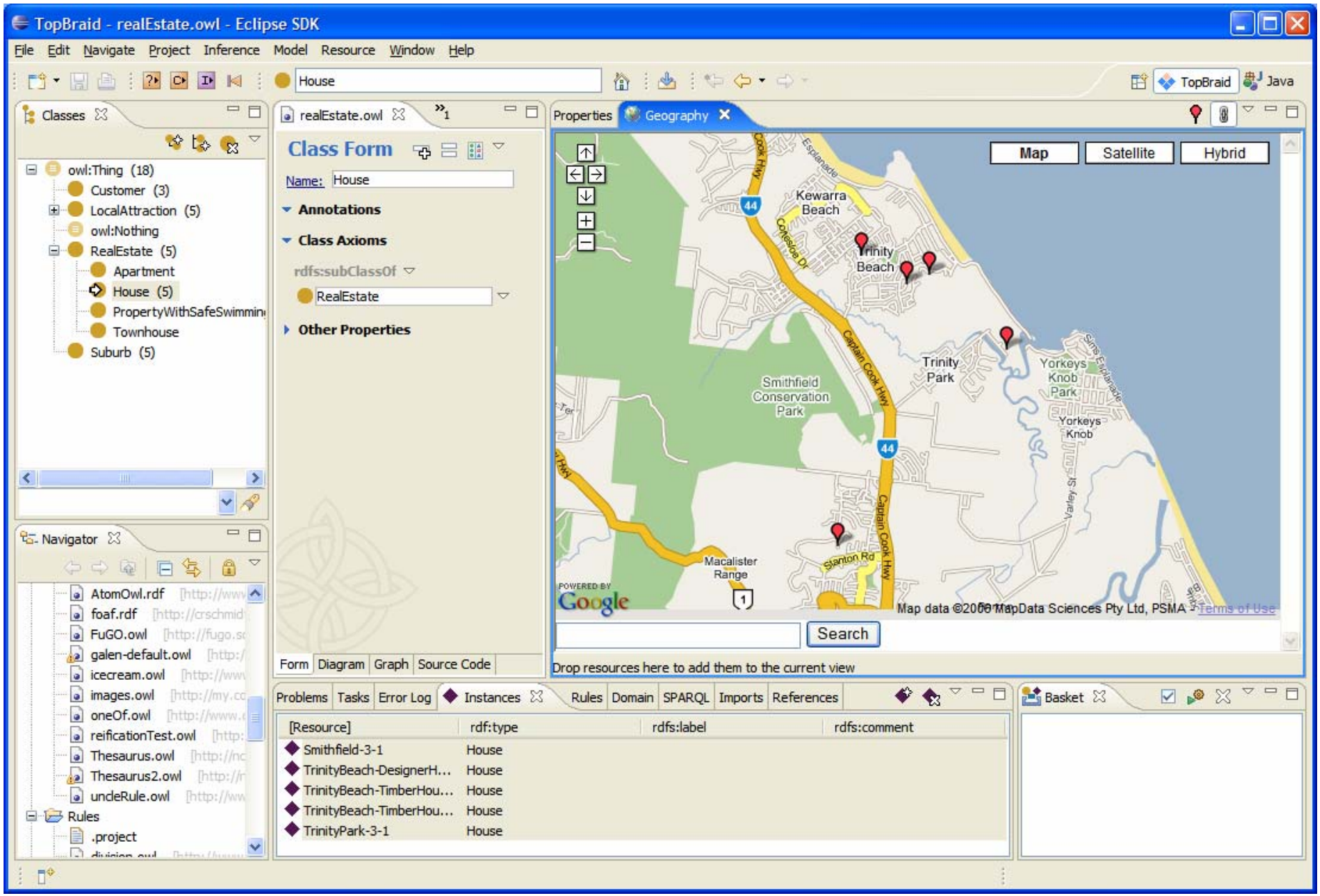
- Real Estate properties are located in Suburbs
- Suburbs have local attractions (Beaches etc)

Ontology Overview (2)

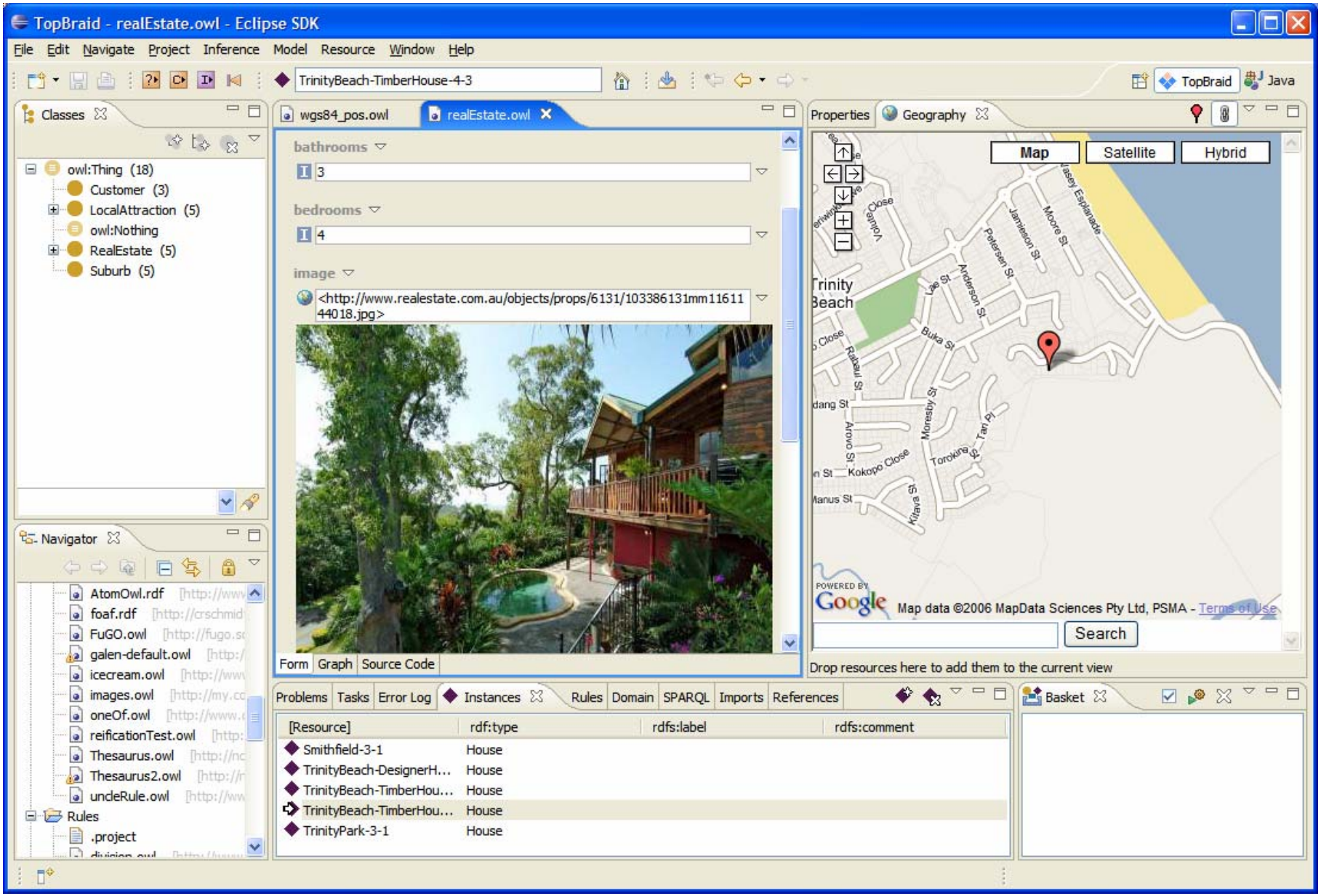


- We have various types of Real Estate properties
- Properties are suggested to Customers

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

```
[convertAU2USDollar:
        (?p :priceAU ?aud)
        product(?aud 0.7745 ?usd)
->      (?p :priceUS ?usd)]
```

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

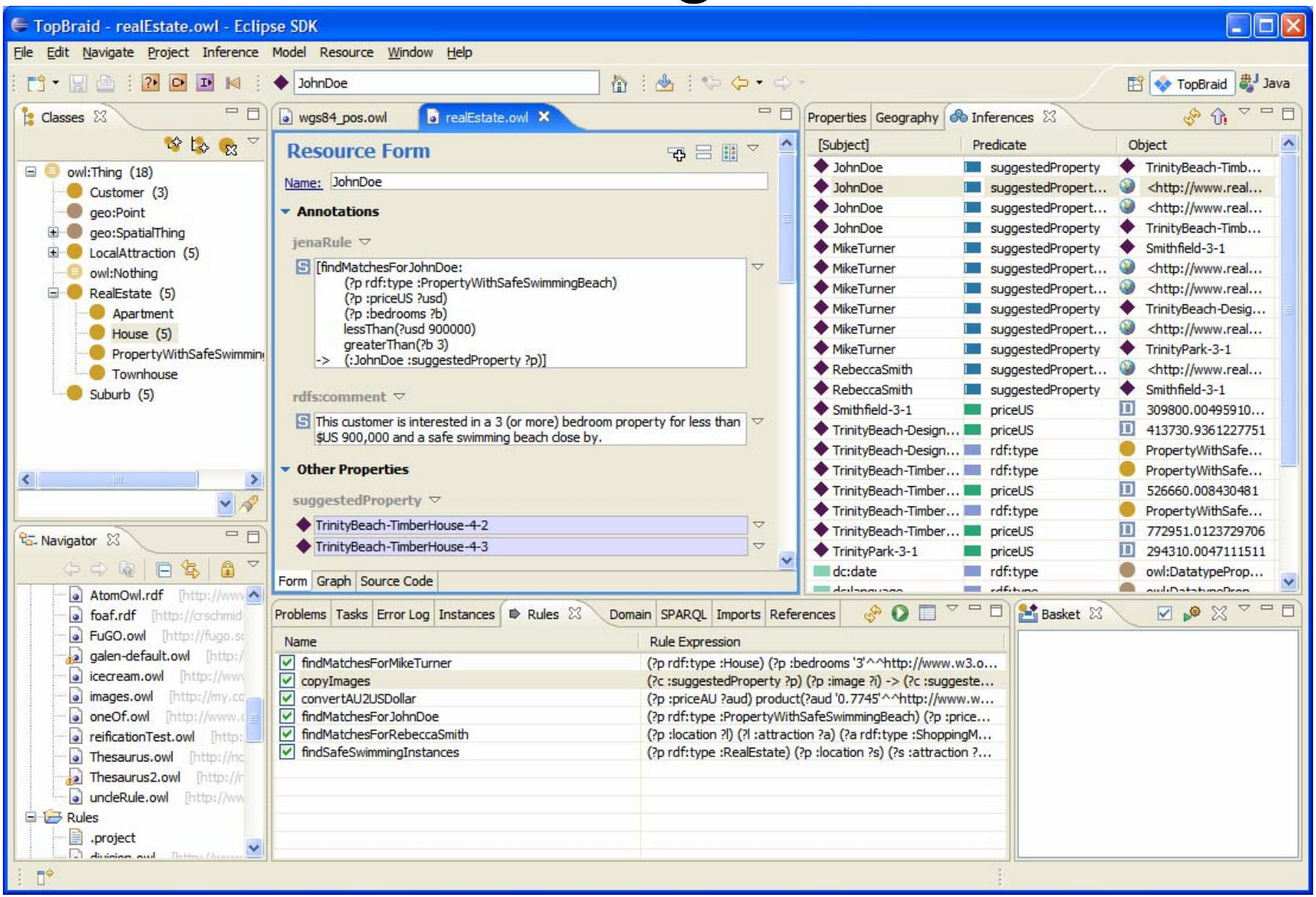
```
[findMatchesForJohnDoe:
        (?p rdf:type :PropertyWithSafeSwimmingBeach)
        (?p :priceUS ?usd)
        (?p :bedrooms ?b)
        lessThan(?usd 900000)
        greaterThan(?b 3)
->      (:JohnDoe :suggestedProperty ?p)]
```

Rule 6: Copying Values

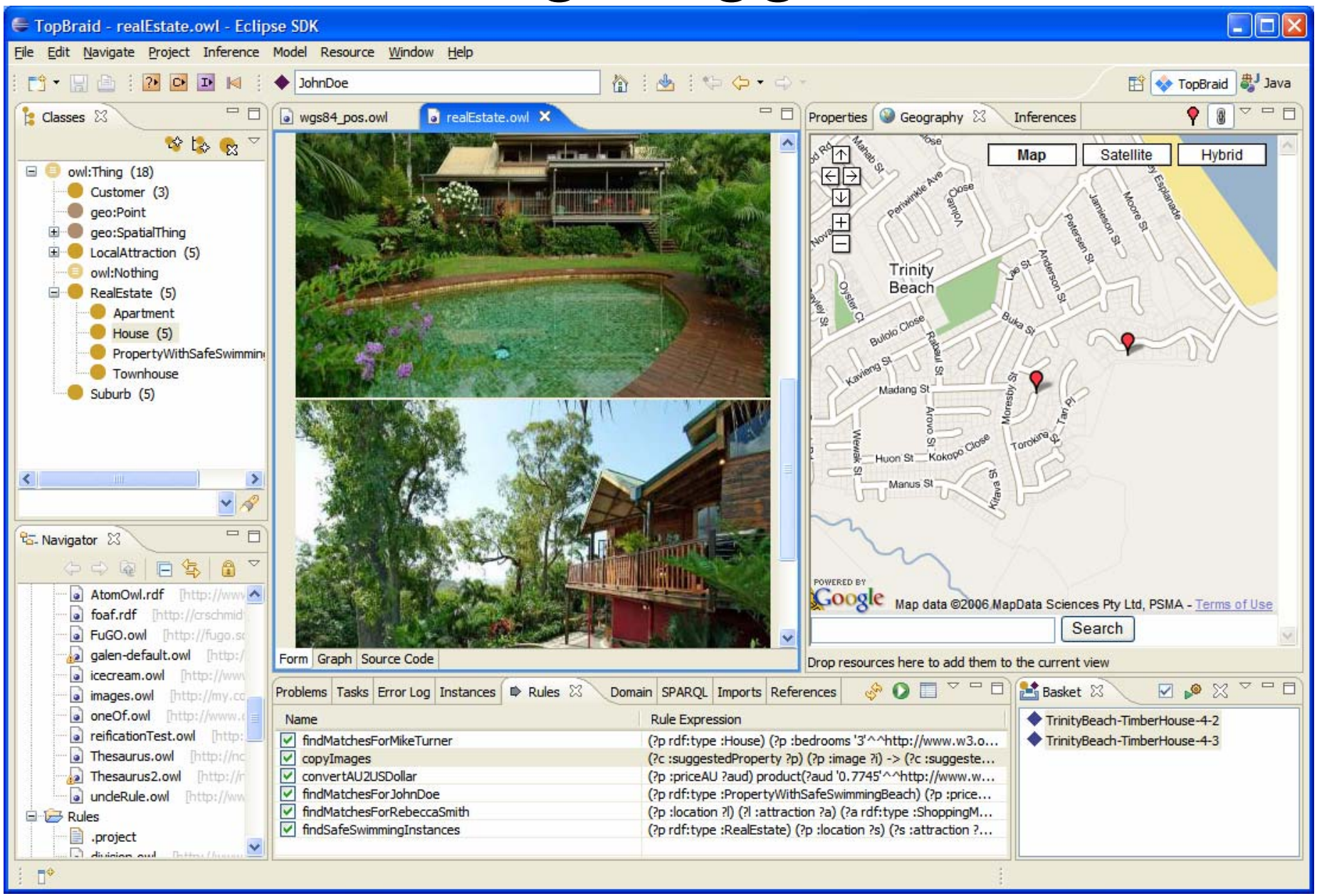
- Whenever something is a suggestedProperty, then we want to copy its image into suggestedPropertyImage

```
[copyImages:
        (?c :suggestedProperty ?p)
        (?p :image ?i)
->      (?c :suggestedPropertyImage ?i)]
```

Executing Rules

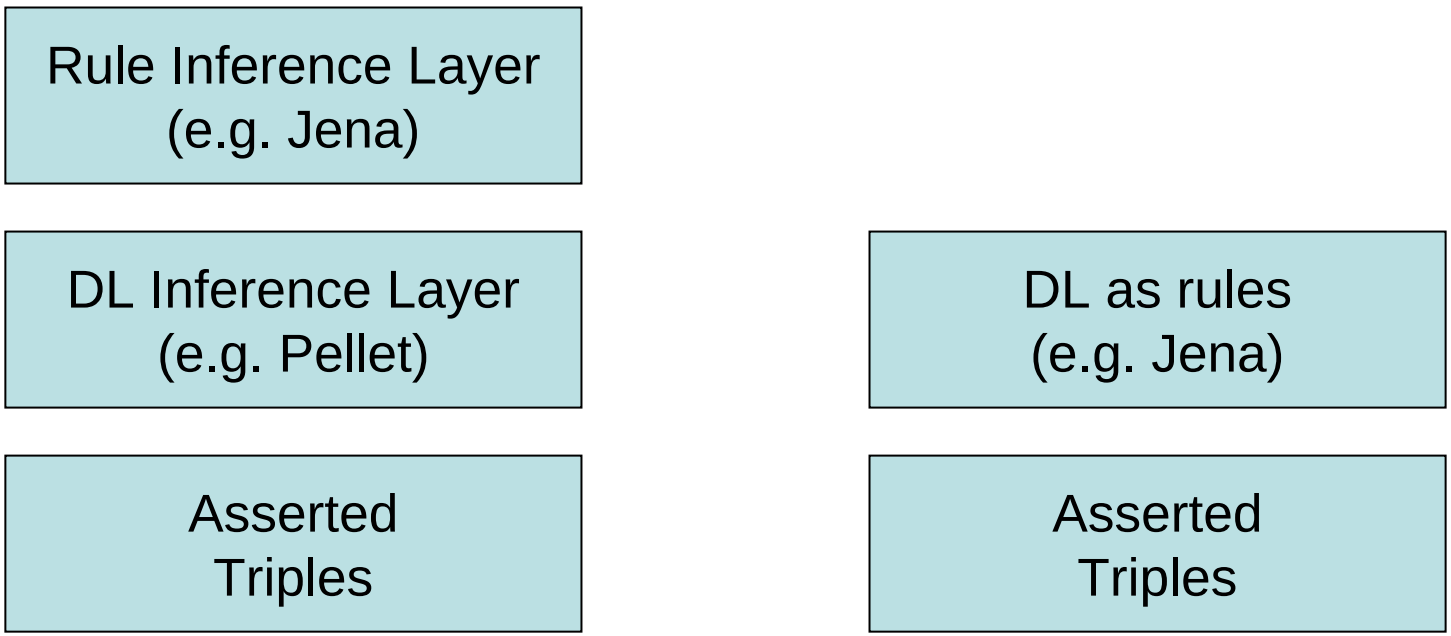


Browsing Suggestions



OWL DL and Rules

- Rules can be executed “on top of” DL
- DL can be implemented by Rules



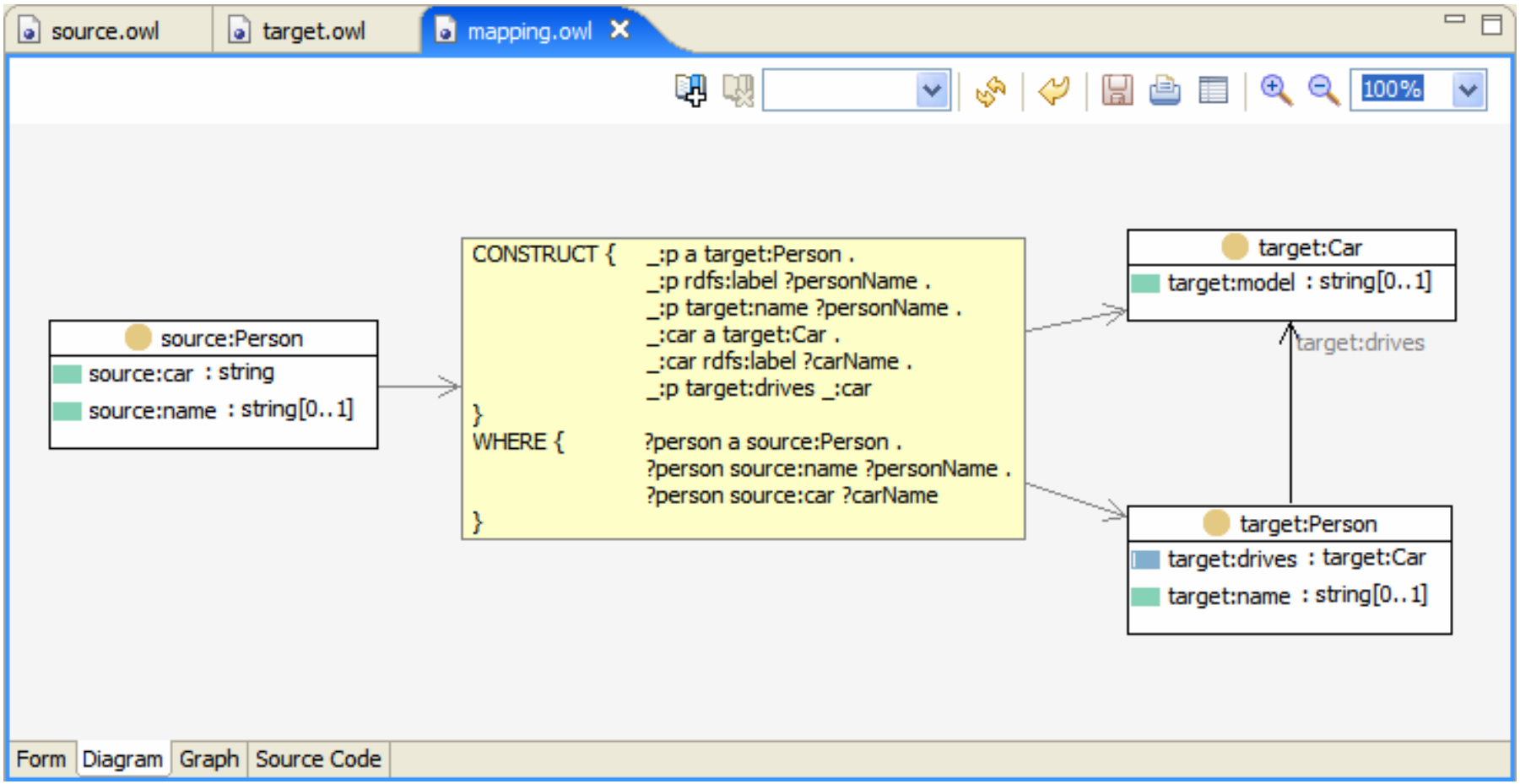
OWL vs. Rules

OWL	SWRL / RIF
W3C Recommendation	Standard in Progress
Recent implementations	>20 years technology
Formal decidability	Possibility of Spaghetti code
Restriction language highly constrained	Powerful pattern language

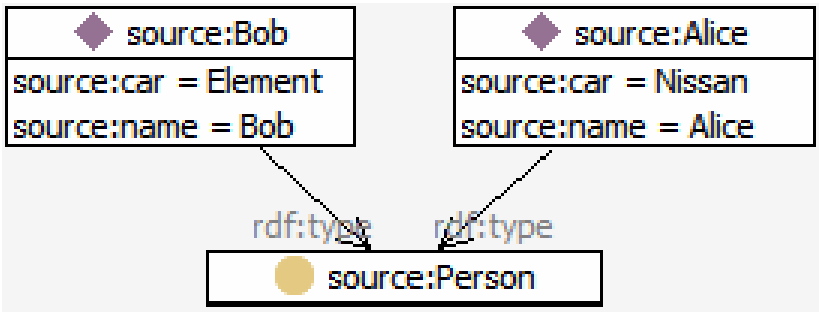
SPARQL

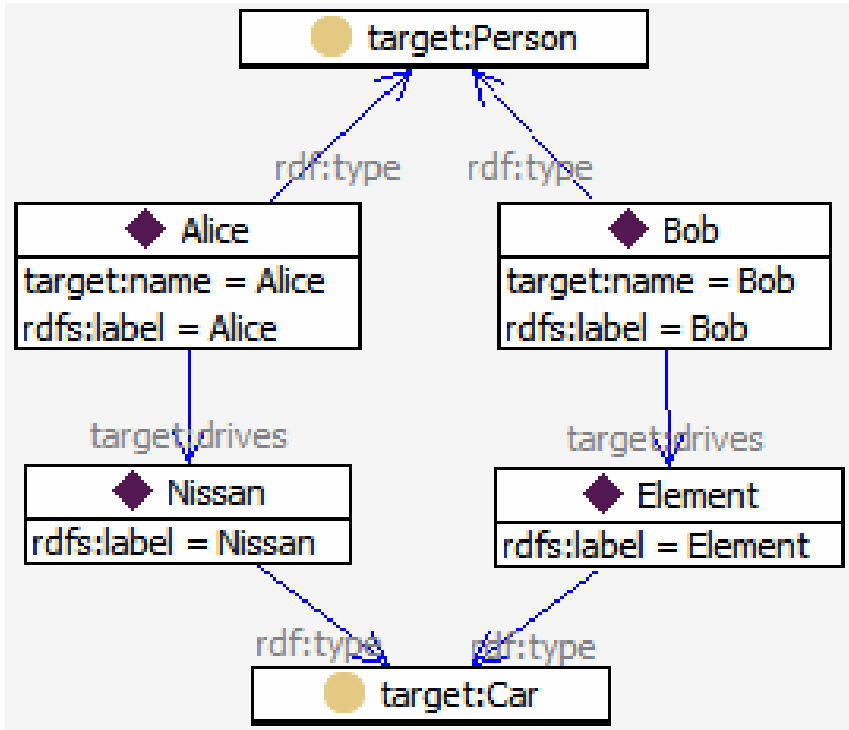
- Not designed as a rule language
- W3C Standard query language for RDF
- Triple matching
  - SELECT
  - CONSTRUCT
- “Pragmatic” rule language

Schema Mapping with SPARQL



Schema Mapping (2)





SPARQL and RULES

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