

Information Integration Intelligence with Semantic Technology

Ontolog Forum 2008-01-24

http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2008_01_24

Holger Knublauch



holger@topquadrant.com
<http://www.topquadrant.com>

About Myself



- Computer Scientist (PhD, 2002)
- 2003-05 Post-Doc at Stanford
 - Lead developer of Protégé-OWL
- 2006-now TopQuadrant, Inc.
 - VP, Product Development
 - Lead developer of TopBraid Suite

<http://www.knublauch.com>



About TopQuadrant



Headquarter: Alexandria, VA



Office: Mountain View, CA

Also: TopQuadrant Korea



TopBraid Suite

Design

TopBraid
Composer™



Deploy

TopBraid
Live™



Collaborate

TopBraid
Ensemble™



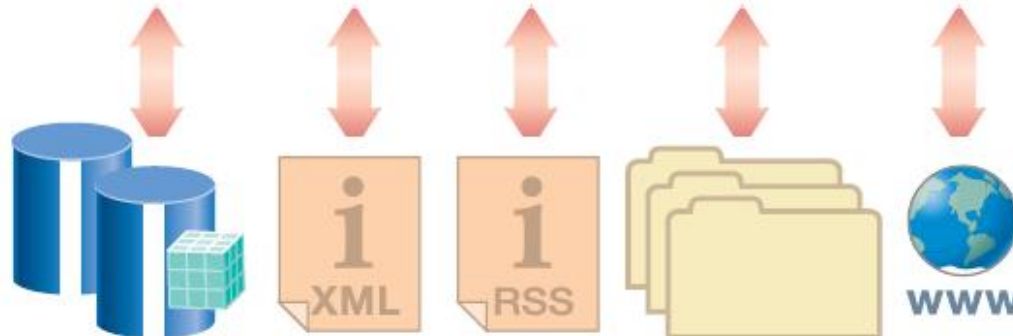
*Merging, Parsing, Event Handling,
UI Generation, Imports/Exports,
Co-ordination, Data Bridges, ...*

*Optional RDF Stores,
Optional Reasoners,
Custom Third-party Extensions*

**TopBraid
Common Services**

**Pluggable
Components**

TopBraid Platform



Information Integration Intelligence

- Heterogeneous data and schemas
 - Databases
 - Spreadsheet files
 - XML files
 - Newsfeeds
 - Online resources (HTML, GRDDL, RDFa)
 - Web services and data endpoints
- How to get integrated views to support business intelligence?



Why Semantic Technology

- Class and property definitions (RDFS)
- Open architecture (URIs, triples, etc)
- Designed for linking (sameAs etc)
- Schema reuse (subClassOf etc)
- Explicit definitions of “semantics” (DL)
- Self-describing data (generic tools, discovery, schema evolution)
- Cross-schema querying (SPARQL)



Semantic Technology Examples (1)

- Major retailer with an established name in Housewares, Lawn and Garden, Automotive and other products. Can we give our shoppers an integrated way to deal with warranties, service records, proofs of purchase, etc. for all our product lines? An "Orbitz of Housewares"
- But they have hundreds of product lines, and new ones every day. How can they do this on this scale?
- TopBraid Semantic Technologies provides seamless integration of many and varied product lines
- Customers come to this retailer instead of competitor to get integrated support of new appliances with old



Semantic Technology Examples (2)

- Consumer Electronics. Marketing and distributing information about products. Consumer electronics is notorious for new product categories with new features (game boxes? entertainment centers? HDTV? DVR?) and compatibility dependencies.
- How do we present our customer base with a seamless integrated picture of all possible products and how they combine, in the face of such a large set of product lines, with changing requirements?
- TopBraid / Semantic Technologies provides a flexible, extensible way to manage multiple products seamlessly



Semantic Technology Examples (3)

- Health care solution built by CTG. Health care providers as well as patients require a seamless, integrated view of all health care information and services:
 - tests
 - available drugs
 - insurance information
 - clinic availability, etc
- Information is available for these things, but cannot be managed in a single seamless way.
- CTG is using TopBraid to create a seamless health care dashboard.

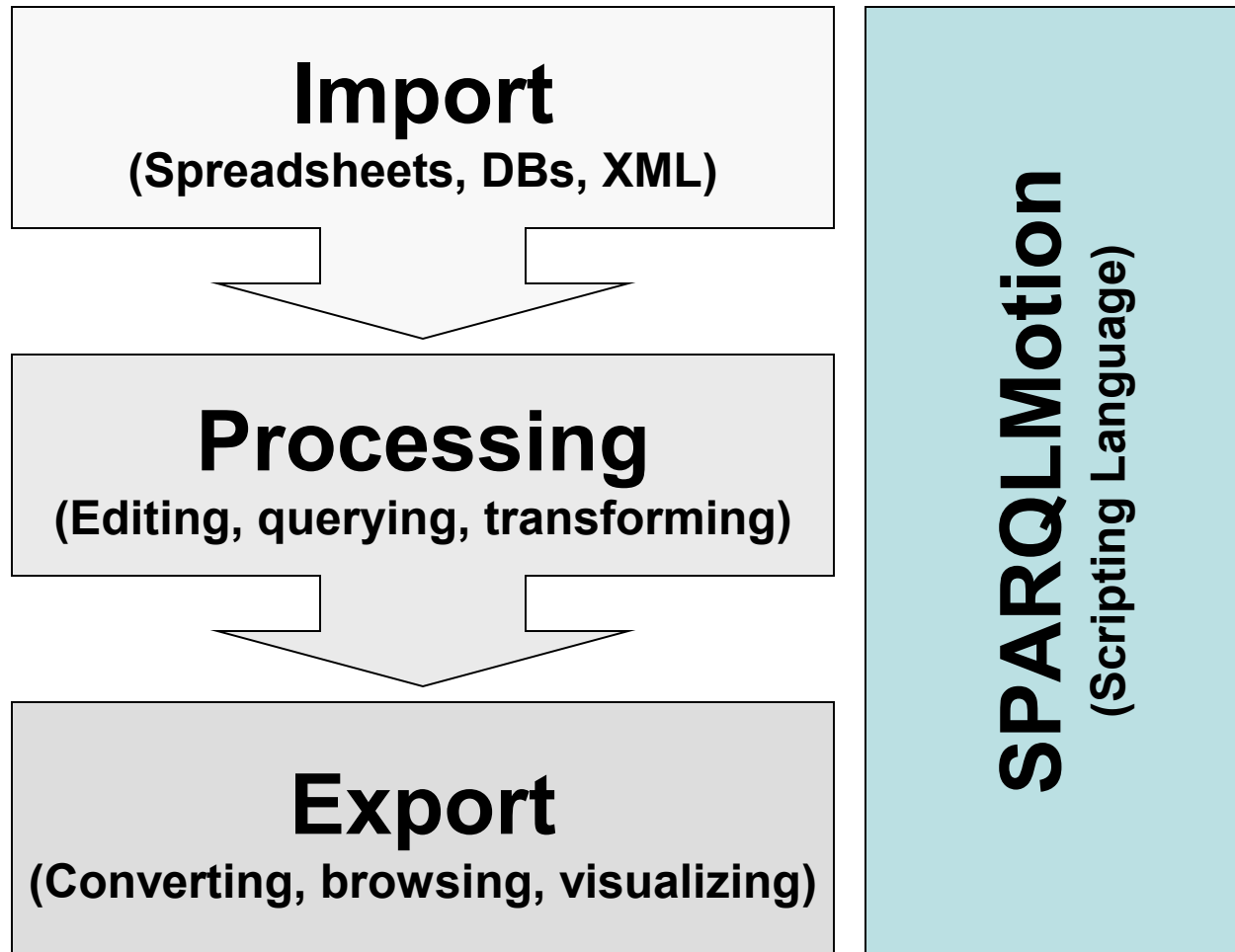


Semantic Technology Examples (4)

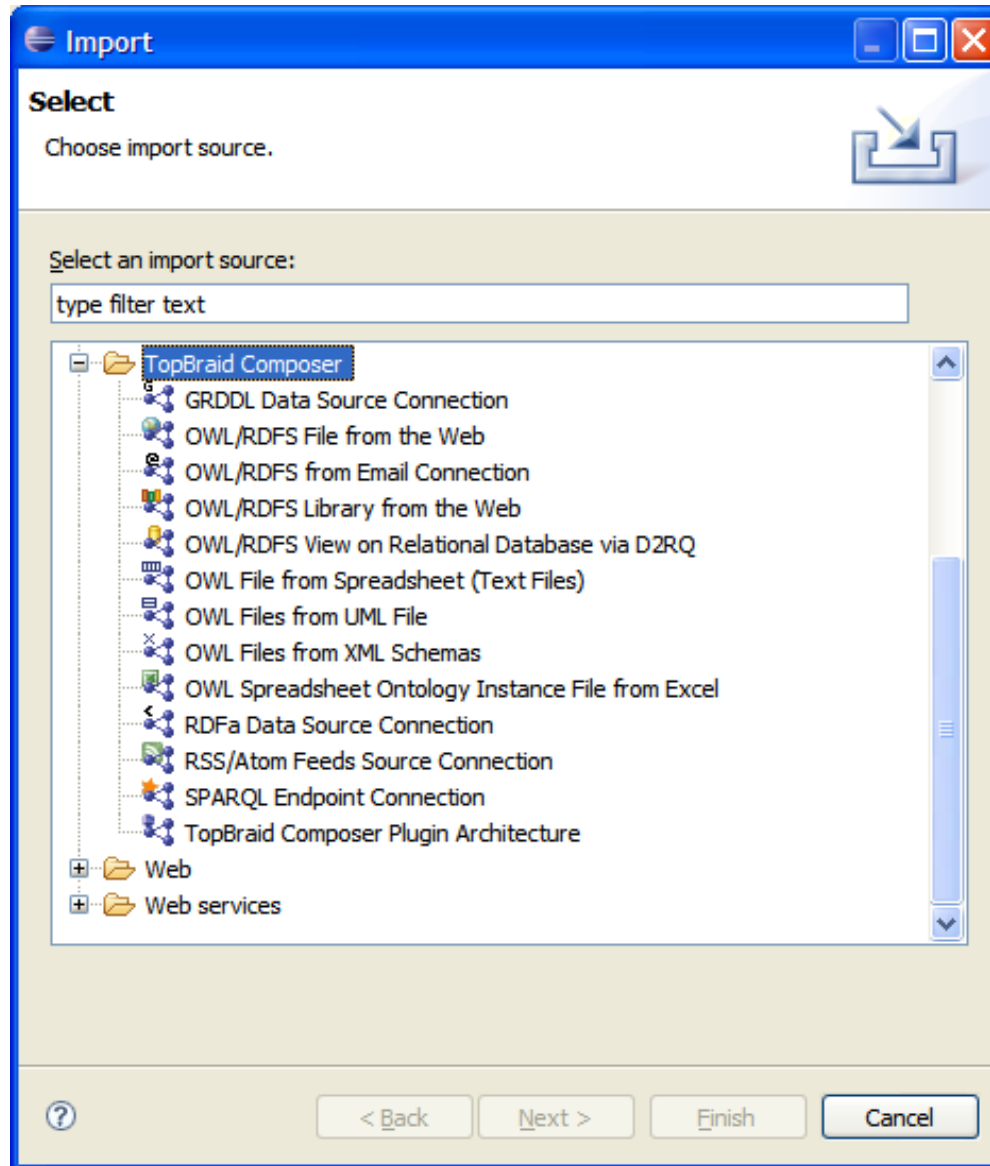
- NASA Constellation project requires integration of information from an astonishingly wide variety of sources - different disciplines (hydraulics, electronics, mechanics, avionics, aerodynamics . . .).
- In the design stage any particular simulation (testing or evaluating design alternatives for a space system) will require a seamless view of a component from any number of perspectives. Even within a single discipline, different groups have information that contributes to a decision.
- Considering the operations and longevity requirements: Constellation project creates data that will be used 30 years into the future - think about the form of data 30 years ago (A lesson learned with the Space Shuttle, in which line drawings for designs had to be consulted 25 years later). The information architecture has to be flexible enough to withstand the passage of all those years.
- NASA is using TopBraid / Semantic Technologies to make flexible, future proof data systems to take a person to Mars.



Structure of this Talk



TopBraid Import Features



Spreadsheet Import in TopBraid

- In practice a lot (!) of useful data resides in spreadsheets
- Excel Spreadsheets can be quite sophisticated (programs on their own)
- TopBraid has two importing options
 - Excel files, each cell becomes an instance
 - Text files, each row becomes an instance

Excel Import in TopBraid

- Sometimes, spreadsheets are not just single tables
- Each cell may have a distinct meaning
- Information about cell position must be preserved

»» Start Here! ««

Atm Supply and Pressure Control

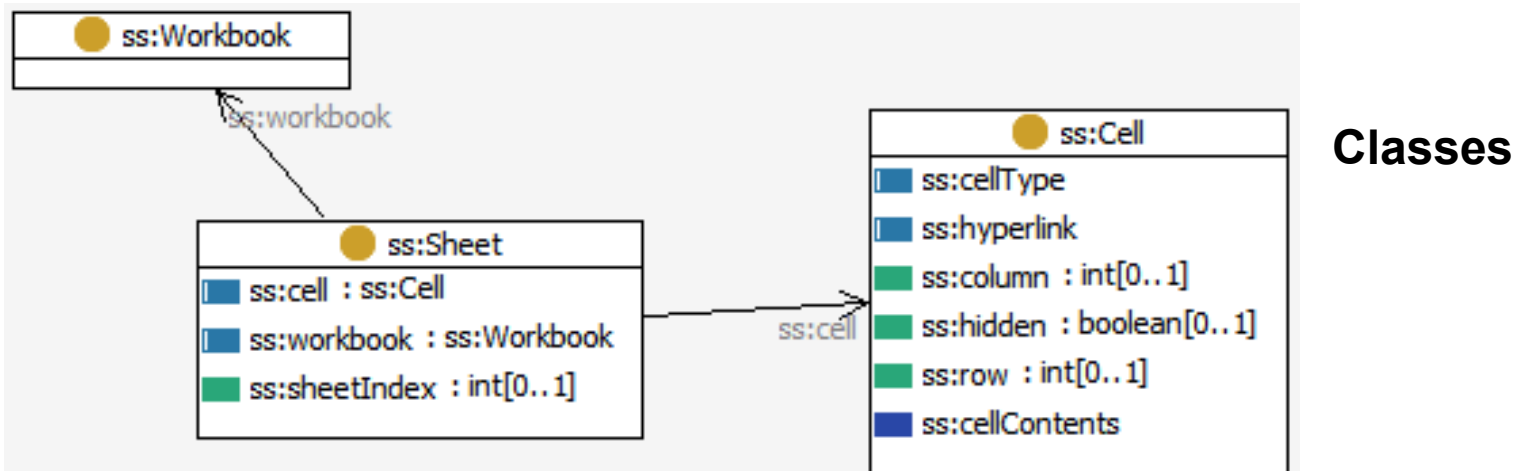
| | |
|-------------------|--|
| Mixed | Atmosphere Type |
| Included in Other | Oxygen Storage |
| 0.8 | kg/p/day Oxygen Consumption Rate |
| 1 | # Allowable Number of Full Cabin Repressurizations |
| 21.1 | °C Cabin Temperature |
| 10 | °C Cabin Dew Point |

Atm Contaminant Control and Ventilation

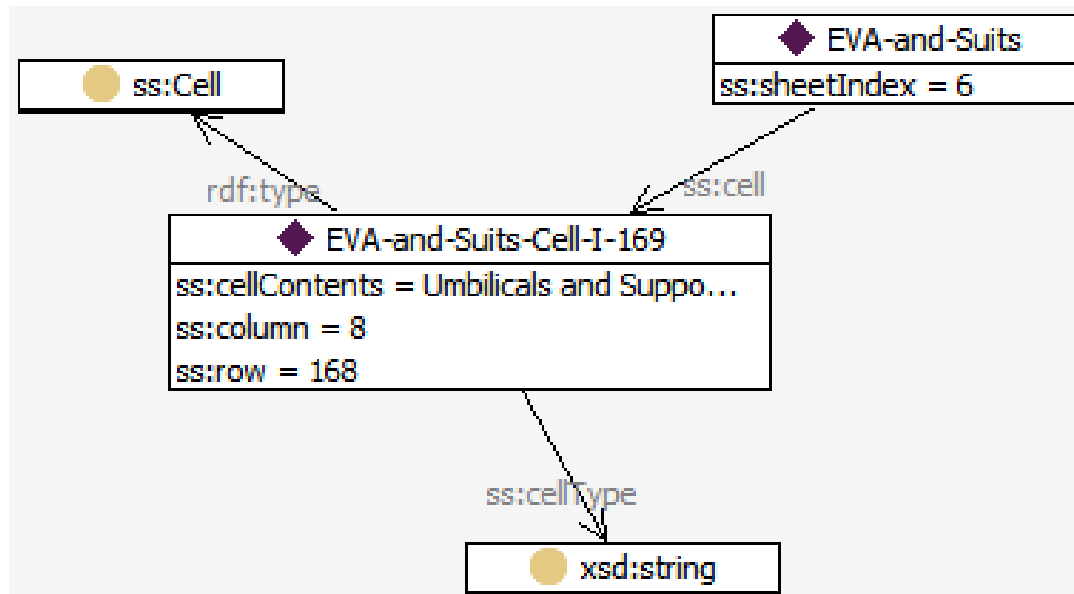
| | |
|------|------------------------------|
| None | Regenerative CO2 Removal |
| None | Non-regenerative CO2 Removal |

TopQuadrant™

TopBraid Spreadsheet Ontology



Example Instance



Spreadsheet Import

- Input: Tab-separated text files
- Table is interpreted as class
- Columns can be mapped into properties
- Rows become instances
- Import wizard can be used to fine tune



TopBraid Spreadsheet Import (1)

The image shows two overlapping dialog boxes from the TopBraid application. The background dialog is 'Import Spreadsheet' and the foreground dialog is 'Preview Spreadsheet'.

Import Spreadsheet Dialog:

- Title:** Import Spreadsheet
- Description:** This wizard can be used to load the contents of a spreadsheet into RDF/OWL files which can be opened with TopBraid.
- Spreadsheet File:** C:\Airports.txt
- Buttons:** Browse File System..., Browse Workspace...
- Base namespace as sections:**
 - General section: http://www.mycompany.com/
 - Local section: /Features
- Base namespace:** http://www.mycompany.com/Features/Airports.owl#
- Import to the current ontology (uses the current namespace):**
- Navigation:** < Back, Next >

Preview Spreadsheet Dialog:

- Title:** Preview Spreadsheet
- Logo:** RDF
- Table:**

| Column # | Column in spreadsheet | Property in ontology | Property data type |
|---------------------------------------|-----------------------|----------------------|--------------------|
| <input checked="" type="checkbox"/> 1 | Code | code | xsd:string |
| <input checked="" type="checkbox"/> 2 | Name | name | xsd:string |
| <input checked="" type="checkbox"/> 3 | Address | address | xsd:string |
| <input checked="" type="checkbox"/> 4 | Weight (lbs) | weightlbs | xsd:string |
| <input checked="" type="checkbox"/> 5 | Type | type | xsd:string |
| <input checked="" type="checkbox"/> 6 | Type2 | type2 | xsd:string |
| <input checked="" type="checkbox"/> 7 | Boardings | boardings | xsd:string |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Pattern for instance names: %1

Class Name: Airport

Navigation: < Back, Next >, Finish, Cancel

TopBraid Spreadsheet Import (2)

The screenshot displays the TopBraid Eclipse SDK interface. The main window is titled "TopBraid - Features/Airports.owl - Eclipse SDK". The interface is divided into several panes:

- Classes:** A tree view on the left showing the ontology structure. It includes `owl:Thing (847)`, `Airport (847)`, and `owl:Nothing`.
- Resource Form:** The central pane shows the "Resource Form" for the `LAX` resource. It includes fields for `Name: LAX`, `rdfs:label: LAX`, `address: Los Angeles CA`, and `boardings`. There are tabs for "Form", "Graph", and "Source Code".
- Properties:** A list of properties on the right, including `address`, `boardings`, `code`, `name`, `type`, `type2`, `weightlbs`, `owl:versionInfo`, `rdfs:comment`, `rdfs:label`, and `rdfs:seeAlso`.
- Instances:** A table at the bottom displays a list of instances. The table has columns for `[Resource]`, `rdf:type`, `rdfs:label`, and `rdfs:comment`.
- Navigator:** A pane on the bottom left showing the project structure, including folders like `Features`, `.settings`, `BIRT`, `D2RQ`, `Forms`, and `SemanticXML`.
- TopBraid Tools:** A toolbar at the top right includes icons for `TopBraid`, `Report Design`, and `Java`.

| [Resource] | rdf:type | rdfs:label | rdfs:comment |
|------------|----------|------------|--------------|
| LAS0 | Airport | LAS | |
| LAW | Airport | LAW | |
| LAX | Airport | LAX | |
| LAX0 | Airport | LAX | |
| LBB | Airport | LBB | |
| LBB0 | Airport | LBB | |
| LBE | Airport | LBE | |
| LBF | Airport | LBF | |

Relational Database Import

- Much enterprise data resides (and needs to stay) in relational databases
- Relational database importer (D2RQ) built into TopBraid
- Static import of schema
 - Tables become classes
 - Columns become properties
 - Link tables become object properties
- Dynamic import of actual data
 - Rows become instances
 - On the fly, i.e. data can stay where it is



Database Import in TopBraid (1)

Specify database connection

This wizard will generate OWL files so that you can handle your relational database as a virtual RDF store.

File name (without suffix):

Base URI of Instances (Database):

Base URI of generated Schema File:

Base URI of generated Mapping File:

Generate OWL Schema (otherwise: RDFS)

Database URL:

Database User Name:

Database Password:

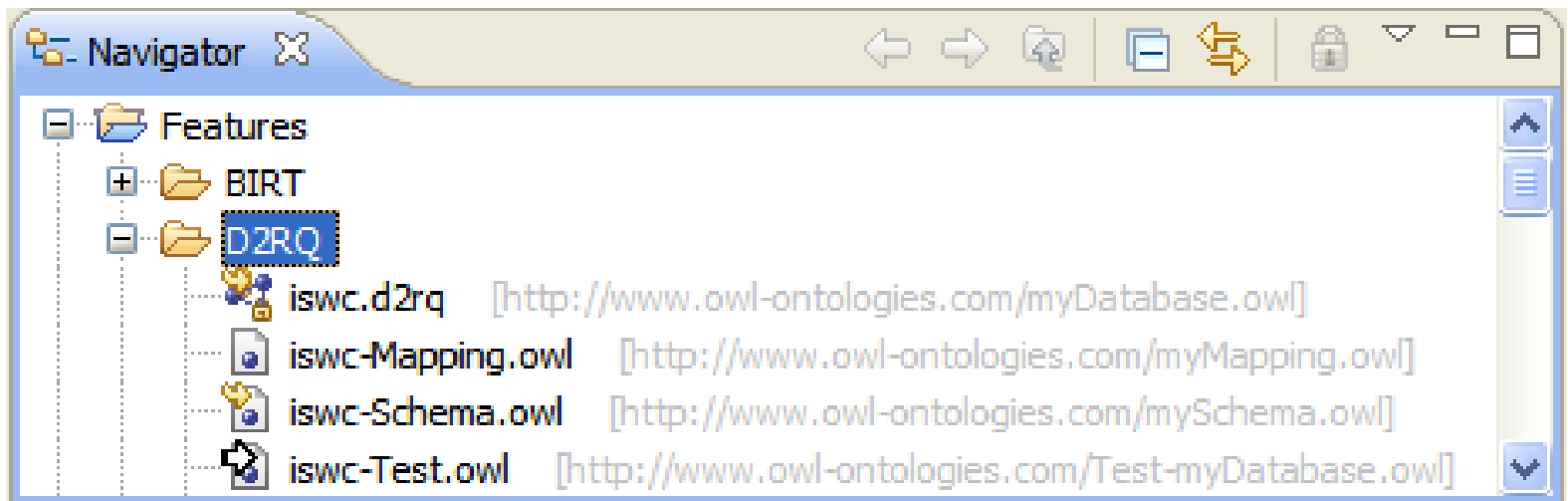
Driver Class:

Driver Jar URL:

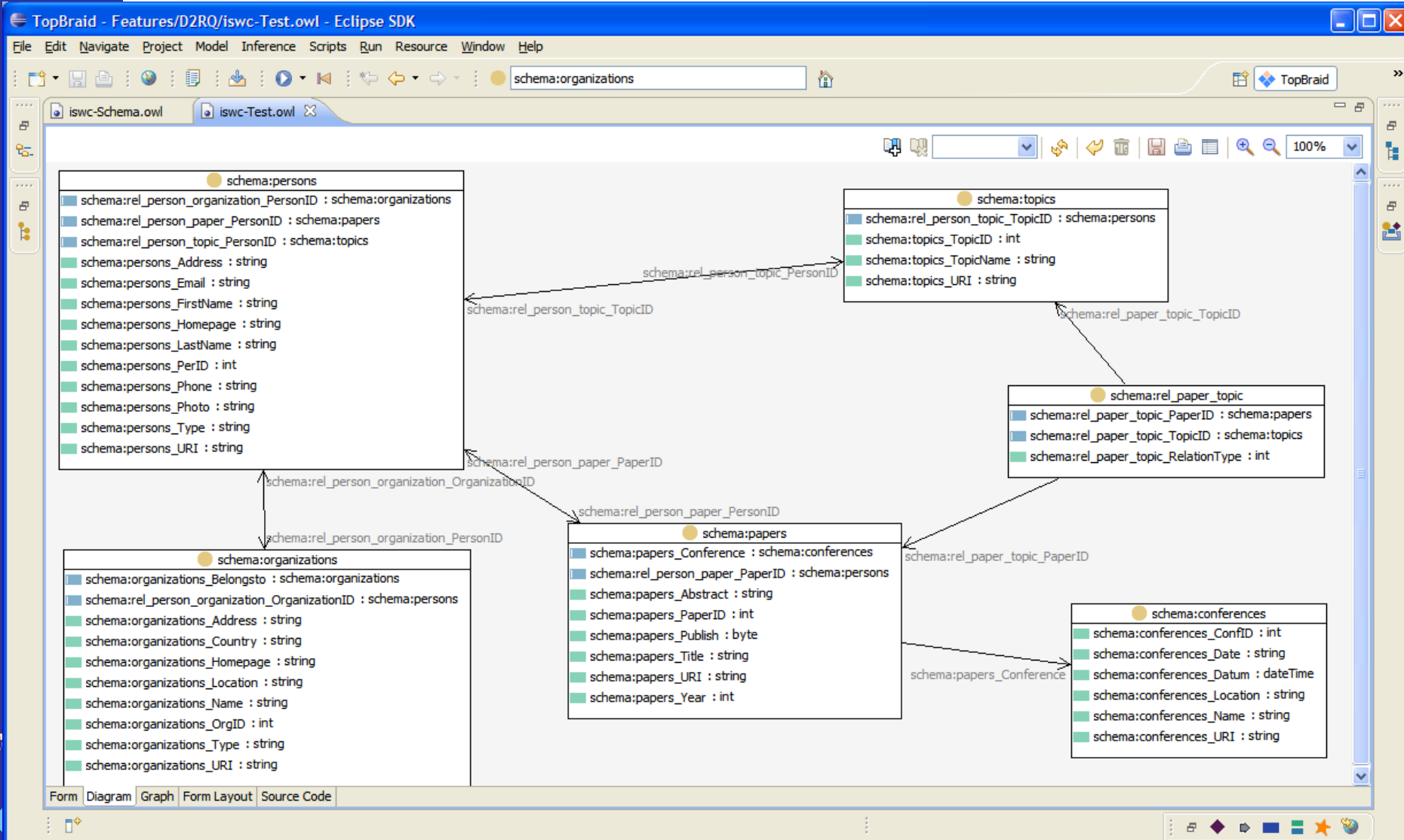


Database Import in TopBraid (2)

- TopBraid automatically generates
 1. Schema
 2. Instances placeholder file (.d2rq)
 3. Mapping file (table-to-class mapping)
 4. Test file that imports 1. and 2. based on 3.



Database Import in TopBraid (3)



Database Import in TopBraid (4)

The screenshot displays the TopBraid Eclipse SDK interface. The main window is titled "TopBraid - Features/D2RQ/iswc-Test.owl - Eclipse SDK". The interface is divided into several panes:

- Classes:** A tree view showing the ontology structure, including classes like owl:Thing (52), schema:conferences (1), schema:organizations (9), schema:papers (5), schema:persons (9), schema:rel_paper_topic (13), and schema:topics (15).
- Resource Form:** A form for editing the resource "db:persons10". It includes sections for Annotations and Other Properties. The "Other Properties" section is expanded, showing fields for schema:persons_Address (DII- Università di Modena e Reggio Emilia via Vignolese 905 41100 Modena), schema:persons_Email (bergamaschi.sonia@unimo.it), schema:persons_FirstName (Sonia), schema:persons_Homepage (http://www.dbgroup.unimo.it/Bergamaschi.html), and schema:persons_LastName (Bergamaschi).
- Properties:** A list of properties from the ontology, including schema:organizations_Belongsto, schema:papers_Conference, schema:rel_paper_topic_PaperID, schema:rel_paper_topic_TopicID, schema:rel_person_organization_OrganizationID, schema:rel_person_organization_PersonID, schema:rel_person_paper_PaperID, schema:rel_person_paper_PersonID, schema:rel_person_topic_PersonID, schema:rel_person_topic_TopicID, schema:conferences_ConfID, schema:conferences_Date, schema:conferences_Datum, schema:conferences_Location, schema:conferences_Name, schema:conferences_URI, schema:organizations_Address, and schema:organizations_Country.
- Instances:** A table showing instances of the class schema:persons. The table has two columns: "[Resource]" and "rdf:type".
- Navigator:** A tree view showing the project structure, including Features, BIRT, D2RQ, and iswc.d2rq.
- Basket:** A pane for managing the current workspace.

| [Resource] | rdf:type |
|--------------|----------------|
| db:persons1 | schema:persons |
| db:persons10 | schema:persons |
| db:persons11 | schema:persons |
| db:persons2 | schema:persons |
| db:persons3 | schema:persons |

Database Import in TopBraid (6)

- Relational databases imported by D2RQ become triple sources like any other – but original data can stay where it is
- Resulting mapping can be fine-tuned
- Full range of generic RDF/OWL tools can be executed
 - Inferencing
 - Merging
 - Mapping
 - Querying
- Not all of these perform equally well

XML Import/Export

- XML is the favourite syntax in many areas, e.g. data exchange between tools, web services
- TopBraid supports two approaches
 - XML Schema import to ontology
 - Semantic XML



XML Schema Import/Export

The screenshot displays the TopBraid Eclipse SDK interface for an OWL ontology. The main window shows a class diagram for the `CommandVerifier` class, which is part of the `xtce.owl` ontology. The diagram includes the following classes and their properties:

- Comparison**:
 - `comparisonOperator` : [0..1]
 - `value` : string[1..1]
- ComparisonList**:
 - `hasComparison` : Comparison[1..]
- ContainerRef**:
 - `containerRef` : string[1..1]
- CommandVerifier**:
 - `hasBooleanExpression` : BooleanExpression[0..1]
 - `hasComparison` : Comparison[0..1]
 - `hasComparisonList` : ComparisonList[0..1]
 - `hasContainerRef` : ContainerRef[0..1]
 - `hasCustomAlgorithm` : InputAlgorithm[0..1]
 - `hasParameterValueChange` : ParameterValueChange[0..1]
 - `timeToWait` : duration[1..1]

Relationships shown in the diagram include:

- `BooleanExpression` `hasBooleanExpression` `CommandVerifier`
- `Comparison` `hasComparison` `CommandVerifier`
- `ComparisonList` `hasComparisonList` `CommandVerifier`
- `ContainerRef` `hasContainerRef` `CommandVerifier`
- `Comparison` `hasComparison` `ComparisonList`
- `ContainerRef` `hasContainerRef` `ContainerRef`

The interface also shows a **Classes** sidebar on the left with a tree view of the ontology's classes, and a **Properties** sidebar on the right listing various property types. The bottom status bar includes tabs for `Instances`, `Rules`, `Domain`, `Relevant Properties`, `SPARQL`, and `Imports`.

Semantic XML

- Converts arbitrary XML to OWL
- Keeps reverse-engineering info in the resulting ontology, using annotation properties
- Can create XML files from OWL
- Lossless round-tripping of XML
- Mapping ontologies can be edited



Semantic XML Example

```
sxml.owl  plugin.xml
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.2"?>
<plugin>
  <extension
    point="org.topbraided.inference.inferenceEngines">
    <inferenceEngine
      class="org.topbraided.sparql.inference.SPARQLInferenceEngine"
      id="org.topbraided.sparql.inference"
      name="SPARQL CONSTRUCTs"/>
    <inferenceEngine
      class="org.topbraided.sparql.inference.SPARQLResourceConstructorInferenceEngine"
      id="org.topbraided.sparql.constructors"
      name="SPARQL Resource Constructors">
    </inferenceEngine>
  </extension>
  <extension
    point="org.topbraided.strings.rowEditorDrivers">
    <rowEditorDriver class="org.topbraided.sparql.editor.SPARQLRowEditorDriver"/>
  </extension>
</plugin>
```

- Each element name becomes a class
- Each attribute becomes datatype property
- Nesting is mapped into a dedicated object property (composite:child)

Semantic XML Classes

The screenshot displays a software interface with three main panels. The left panel, titled 'Classes', shows a tree view of classes including owl:Thing (7), extension (2), inferenceEngine (2), owl:Nothing, plugin (1), rowEditorDriver (1), and sxml:Node (1). The middle panel, titled 'Class Form', shows the configuration for a class named 'extension'. It includes a 'Name' field with the value 'extension', an 'Annotations' section with a dropdown menu set to 'sxml:element' (circled in red), and a 'Class Axioms' section with a dropdown menu set to 'owl:Thing'. The right panel, titled 'Properties', shows a tree view of properties including composite:child, sxml:root, composite:parent, class-inferenceEngine, class-rowEditorDriver, composite:index, id-inferenceEngine, name-inferenceEngine, point-extension, sxml:text, owl:versionInfo, rdfs:comment, rdfs:label, rdfs:seeAlso, sxml:attribute, and sxml:element.

- Each generated class contains an annotation that points back to the XML element where it came from
- Similarly for the properties

Semantic XML Instances

The screenshot displays an IDE interface for editing Semantic XML. The left pane, titled 'Classes', shows a tree view of a `<plugin>` element. It contains two children: an `<extension point="org.topbraided.inference.inferer">` element with two `<inferenceEngine name="SPARQL CONSTRUCT" />` children, and an `<extension point="org.topbraided.strings.rowEditorDrivers">` element with one `<rowEditorDriver class="org.topbraided.sparql.editor.SPARQLRowEditor" />` child.

The right pane, titled 'Resource Form', shows the details of the selected `<plugin>` element. The 'Name' field is set to 'r'. Under 'Annotations', the `rdf:type` is set to 'plugin'. Under 'Other Properties', the `composite:child` property is expanded to show two children: `<extension point="org.topbraided.inference.inferenceEngines">` and `<extension point="org.topbraided.strings.rowEditorDrivers">`. The second child is further expanded to show its `point-extension` property set to `org.topbraided.strings.rowEditorDrivers`, its `composite:index` property set to '1', and its `composite:parent` property set to the `<plugin>` element. The `rdf:type` of the second child is set to 'extension', and its `composite:child` property is set to `<rowEditorDriver class="org.topbraided.sparql.editor.SPARQLRowEditor" />`.

The bottom pane, titled 'Navigator', shows a list of files: `person2.owl.tbc`, `pizza-1.owl` [http://www.owl-ontologies.com], `plugin.xml` [file://ZZZ/plugin.xml], and `registeredUsers.owl` [http://www.mycompar].

Semantic XML Profiles

- The Semantic XML class models can be edited and fine-tuned
- TopBraid provides a couple of standard profiles
 - XHTML to open .html files (including tidy)
 - XSD to open XML Schemas
- More profiles are planned/prepared
 - X3D
 - SVG



Semantic XML Profile for HTML

The screenshot displays the Eclipse SDK interface for editing a Semantic XML profile. The main window is titled "TopBraid - Features/SemanticXML/realEstate.html - Eclipse SDK".

Class Explorer: Shows a hierarchy of classes for the HTML profile, including `xhtml:Form.content` (2), `xhtml:Head` (1), `xhtml:Hr`, `xhtml:Html` (1), `xhtml:Img` (15), `xhtml:Inline` (99), `xhtml:Abbr`, `xhtml:Acronym`, `xhtml:Address` (1), `xhtml:B`, `xhtml:Bdo`, `xhtml:Big`, `xhtml:Caption`, `xhtml:Cite`, `xhtml:Code`, `xhtml:Dfn`, `xhtml:Dt` (14), `xhtml:Em` (12), `xhtml:H1` (1), `xhtml:H2` (17), and `xhtml:H3` (16).

Properties View: Shows the configuration for the `Resource Form` (Name: `r-1-17-2-1-2-0`). It includes annotations and other properties such as `composite:index` (0), `composite:parent` (`<div id="propertyDetail">`), `rdf:type` (`xhtml:H1`), and `composite:child` (`<"Search Results">`). The `composite:child` section shows `sxml:text` (Search Results), `composite:index` (0), and `rdf:type` (`sxml:TextNode`).

Navigator: Shows the project structure, including `pizzaProcess.xml`, `process.owl`, `realEstate.html` (selected), `sxsd.owl`, `sxsdNexiom.owl`, `SparqlMotion`, `.project`, and `XSD_NEXIOM-asme-v1.9.xml`.

Instances Table: Shows the instances of the profile, with columns for `[Resource]`, `rdf:type`, `rdfs:label`, and `rdfs:comment`. The table contains one instance: `<h1>` with `rdf:type` `xhtml:H1`.

Properties List: A scrollable list of properties including `composite:child`, `composite:parent`, `composite:index`, `language-script`, `style-ul`, `sxml:text`, `target-a`, `title-a`, `xhtml:accept`, `xhtml:accept-charset`, `xhtml:accesskey`, `xhtml:action`, `xhtml:align`, `xhtml:alt`, `xhtml:archive`, `xhtml:axis`, `xhtml:border`, `xhtml:cellpadding`, `xhtml:cellspacing`, `xhtml:char`, `xhtml:charoff`, `xhtml:charset`, `xhtml:checked`, and `xhtml:class`.

Semantic XML Profile for XSD

The screenshot displays the TopQuadrant software interface, showing a Semantic XML Profile for XSD. The left pane, titled 'Classes', shows a hierarchy of classes under 'owl:Thing (1135)'. The selected class is 'xsd:SimpleType (22)'. The right pane, titled 'Resource Form', shows the configuration for a resource named 'r-3'. The 'Annotations' section is expanded, showing 'xsd:name' set to 'ArrestTypeCodeType'. The 'Other Properties' section includes 'xsd:name', 'composite:index' (set to 3), 'composite:parent' (set to a schema URI), 'rdf:type' (set to 'xsd:SimpleType'), and 'composite:child' (set to '<xsd:annotation>' and '<xsd:restriction base = "xsd:token">').

Classes

- owl:Thing (1135)
 - owl:Nothing
 - sxml:Node (279)
 - sxsd:Annotation (278)
 - sxsd:Attribute
 - sxsd:Choice
 - sxsd:ComplexContent
 - sxsd:ComplexType
 - sxsd:Documentation (278)
 - sxsd:Element
 - sxsd:Enumeration (254)
 - sxsd:Extension
 - sxsd:Import (1)
 - sxsd:Restriction (22)
 - sxsd:Schema (1)
 - sxsd:SimpleType (22)

Resource Form

Name: r-3

Annotations

Other Properties

sxsd:name: ArrestTypeCodeType

composite:index: 3

composite:parent: <xsd:schema xmlns:i="http://www.it.ojp.gov/jxdm/appinfo/1" elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema" ...>

rdf:type: xsd:SimpleType

composite:child: <xsd:annotation>, <xsd:restriction base = "xsd:token">

Form Graph Source Code

Semantic XML Summary

- Load, query and generate arbitrary XML files (even without XSD)
- Generated schema can then be fine tuned and reused for other XML files of the same kind
- SPARQL, rules and inferencers can be used to extract or convert the XML



Other Importers

- UML Class Diagrams
- Direct Triple Sources
 - Files (RDF/XML, N3/Turtle, N-Triples)
 - RSS/Atom Feeds
 - GRDDL
 - RDFa
 - SPARQL Endpoints
 - RDF databases (Oracle 11g, Jena, AllegroGraph, Sesame)

Data Processing

- So far: data physically converted to a uniform language (RDF/OWL)
- Semantic integration
 - Ontology editing
 - Mapping by built-in inferences
 - Mapping by constructing new triples



Ontology Editing

- TopBraid Composer is the most sophisticated professional editor for OWL and RDF on the market
- Modular ontologies
- Refactoring
- Form-based & visual editing
- Customizable and extensible
- Driven by requirements from real-world projects (NASA etc)
- Several hundred users (and counting)

TopBraid Composer

The screenshot displays the TopBraid Composer interface within the Eclipse SDK. The main workspace shows a graph view of the ontology `KennedyDemo.owl`. Two instances are visible: `simple:person27` (Maria Shriver) and `simple:person46` (Arnold Schwarzenegger). They are connected by a `simple:spouse` property. The graph view includes a `Palette` on the left with options like `Select`, `Create node`, and `Add connection`. The `Instances` view at the bottom shows a table of instances.

| Resource | rdf:type | [rdfs:label] | rdfs:comment |
|-----------------|---------------|-----------------------|--------------|
| simple:person46 | simple:Person | Anne Kelly | |
| simple:person33 | simple:Person | Anthony Shriver | |
| simple:person27 | simple:Person | Arnold Schwarzenegger | |
| simple:person20 | simple:Person | Caroline Kennedy | |
| simple:person23 | simple:Person | Carolvn Bessette | |

The `Classes` view on the left shows a hierarchy of classes, including `owl:Thing` (135) and `simple:Person` (75). The `Properties` view on the right lists various properties like `simple:alma-mater`, `simple:has-child`, and `simple:spouse`. The `Navigator` view at the bottom left shows the project structure, including folders like `Common`, `DOJ`, and `Examples`.

Ontology Mapping via RDFS/OWL

- `rdfs:subClassOf/owl:equivalentClass`
- `rdfs:subPropertyOf`
- Then run inferencing
- Only suitable for trivial cases
- Limited expressivity



Ontology Mapping with SPARQL

TopBraid - Ontologies-tbl/spider/bugzillaImport.owl - Eclipse SDK

File Edit Navigate Project Model Inference Scripts Run Resource Window Help

user:User

issue.owl bug-export.sesan project.owl sparql.owl topbraid.owl dc-1.1.rdf user.owl

bz:profiles

- bz:profiles_cryptpassword : string
- bz:profiles_disable_mail : byte
- bz:profiles_disabledtext : string
- bz:profiles_extern_id : string
- bz:profiles_login_name : string
- bz:profiles_mybugslink : byte
- bz:profiles_realname : string
- bz:profiles_userid : int

```
CONSTRUCT {  
  ?user a user:User .  
  ?user topbraid:email ?loginName .  
  ?user rdfs:label ?realName .  
}  
WHERE {  
  ?user a bz:profiles .  
  ?user bz:profiles_login_name ?loginName .  
  ?user bz:profiles_realname ?realName .  
}
```

topbraid:PrivilegedEntity

- topbraid:canEditIndividuals : topbraid:P
- topbraid:canRead : topbraid:Project

topbraid:User

- topbraid:email : string
- topbraid:group : topbraid:Group
- topbraid:password : string
- topbraid:userName : string

user:User

- user:roleOf : user:Role

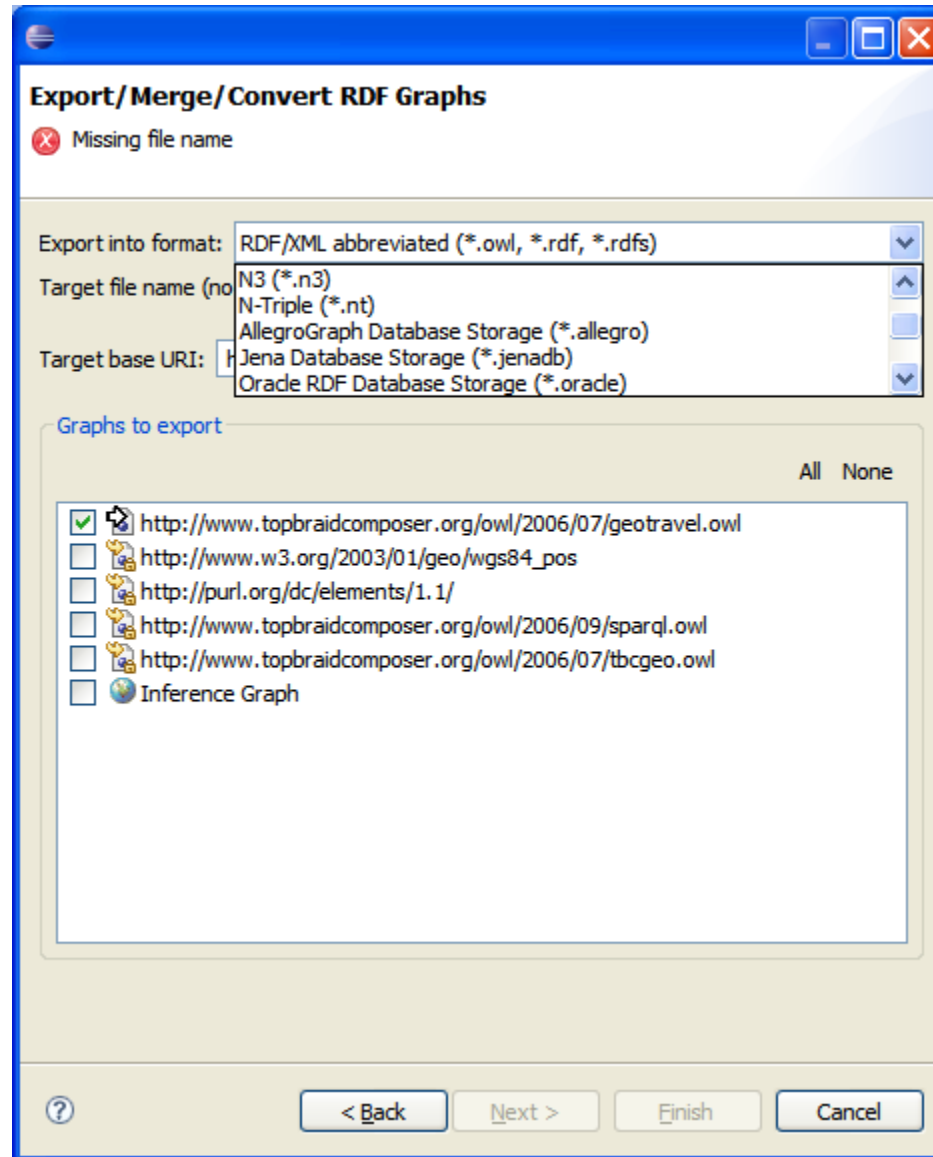
issue:reporter
issue:assignedTo
issue:notifiedUser

TopBraid Export Features

- Triples (databases, files)
- HTML documentation
- Semantic Java Server Pages
- Google Maps, Calendars
- Spreadsheets, Matrix
- Business Intelligence Reports
- Browsing and querying (TopBraid Live)



Export/Merge/Convert Triples




HTML Export

Ontology Documentation

Navigation

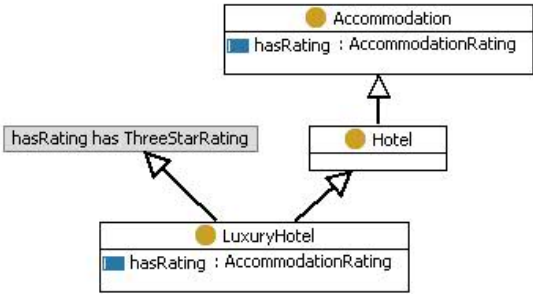
- [Ontologies](#)
- [Classes](#)
- [Datatype Properties](#)
- [Object Properties](#)
- [Individuals](#)
- [All Resources](#)



- [Destination](#)
- [FamilyDestination](#)
- [Farmland](#)
- [Hiking](#)
- [Hotel](#)
- [LuxuryHotel](#)
- [Motel](#)
- [Museums](#)
- [NationalPark](#)
- [QuietDestination](#)
- [Relaxation](#)
- [RetireeDestination](#)
- [RuralArea](#)
- [Safari](#)
- [Sightseeing](#)
- [Sports](#)
- [Sunbathing](#)
- [Surfing](#)
- [Town](#)

<http://www.owl-ontologies.com/travel.owl#LuxuryHotel>

Class LuxuryHotel



```
graph BT; Accommodation --> Hotel; Hotel --> LuxuryHotel; LuxuryHotel -->|hasRating has ThreeStarRating| Accommodation
```

Class Axioms

rdfs:subClassOf

- [hasRating](#) has [ThreeStarRating](#)
- [Hotel](#)

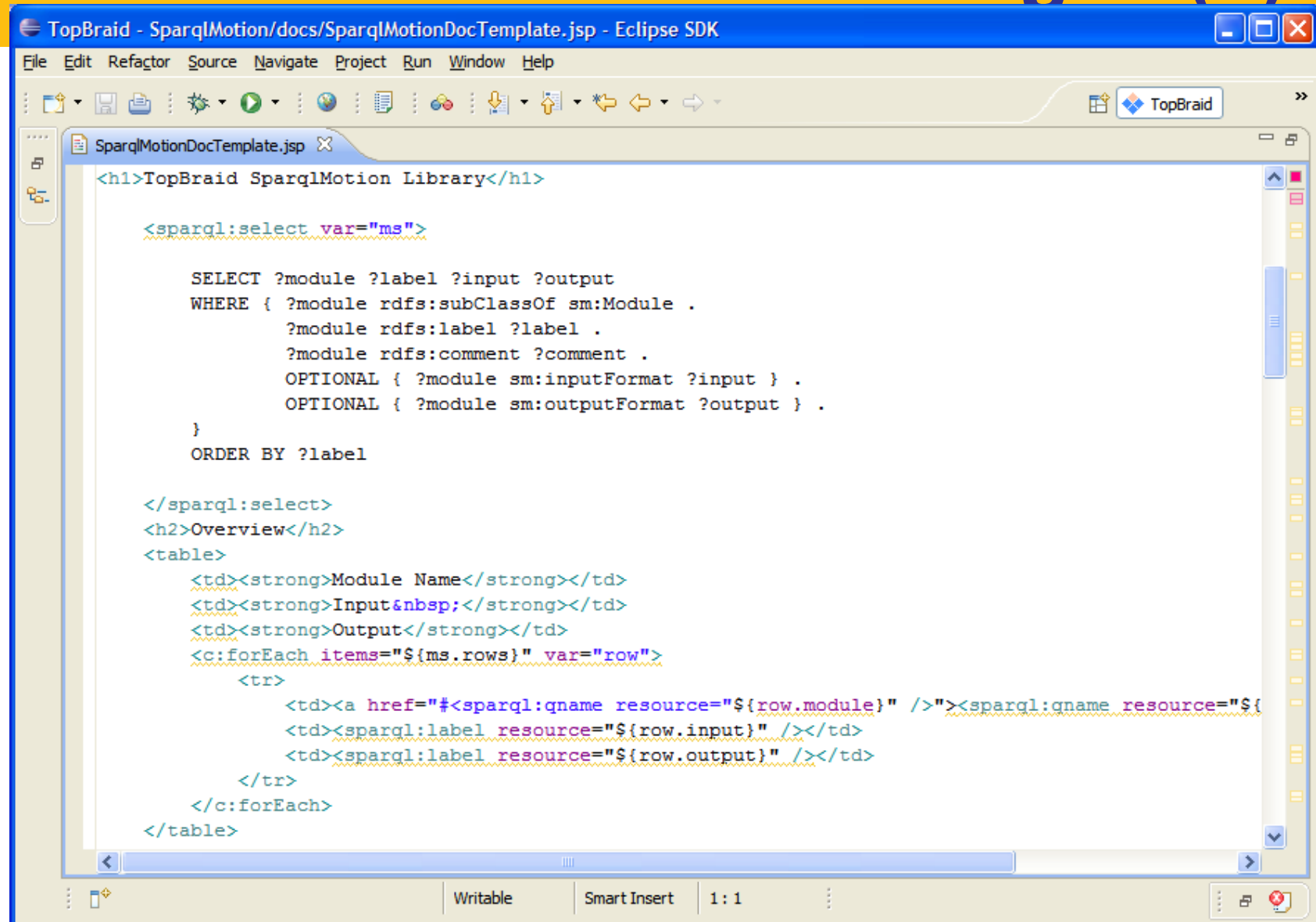
Other Properties

rdf:type

- owl:Class

Instances

Semantic Java Server Pages (1)



```
TopBraid - SparqlMotion/docs/SparqlMotionDocTemplate.jsp - Eclipse SDK
File Edit Refactor Source Navigate Project Run Window Help
SparqlMotionDocTemplate.jsp
<h1>TopBraid SparqlMotion Library</h1>

<sparql:select var="ms">

    SELECT ?module ?label ?input ?output
    WHERE { ?module rdfs:subClassOf sm:Module .
            ?module rdfs:label ?label .
            ?module rdfs:comment ?comment .
            OPTIONAL { ?module sm:inputFormat ?input } .
            OPTIONAL { ?module sm:outputFormat ?output } .
    }
    ORDER BY ?label

</sparql:select>
<h2>Overview</h2>
<table>
    <td><strong>Module Name</strong></td>
    <td><strong>Input<nbsp;</strong></td>
    <td><strong>Output</strong></td>
    <c:forEach items="${ms.rows}" var="row">
        <tr>
            <td><a href="#<sparql:qname resource="${row.module}" />"><sparql:qname resource="${
            <td><sparql:label resource="${row.input}" /></td>
            <td><sparql:label resource="${row.output}" /></td>
        </tr>
    </c:forEach>
</table>
```



Semantic Java Server Pages (2)

TopBraid SparqlMotion Library - Windows Internet Explorer

C:\TQ\TBC\runtimeWork Google

TopBraid SparqlMotion Library

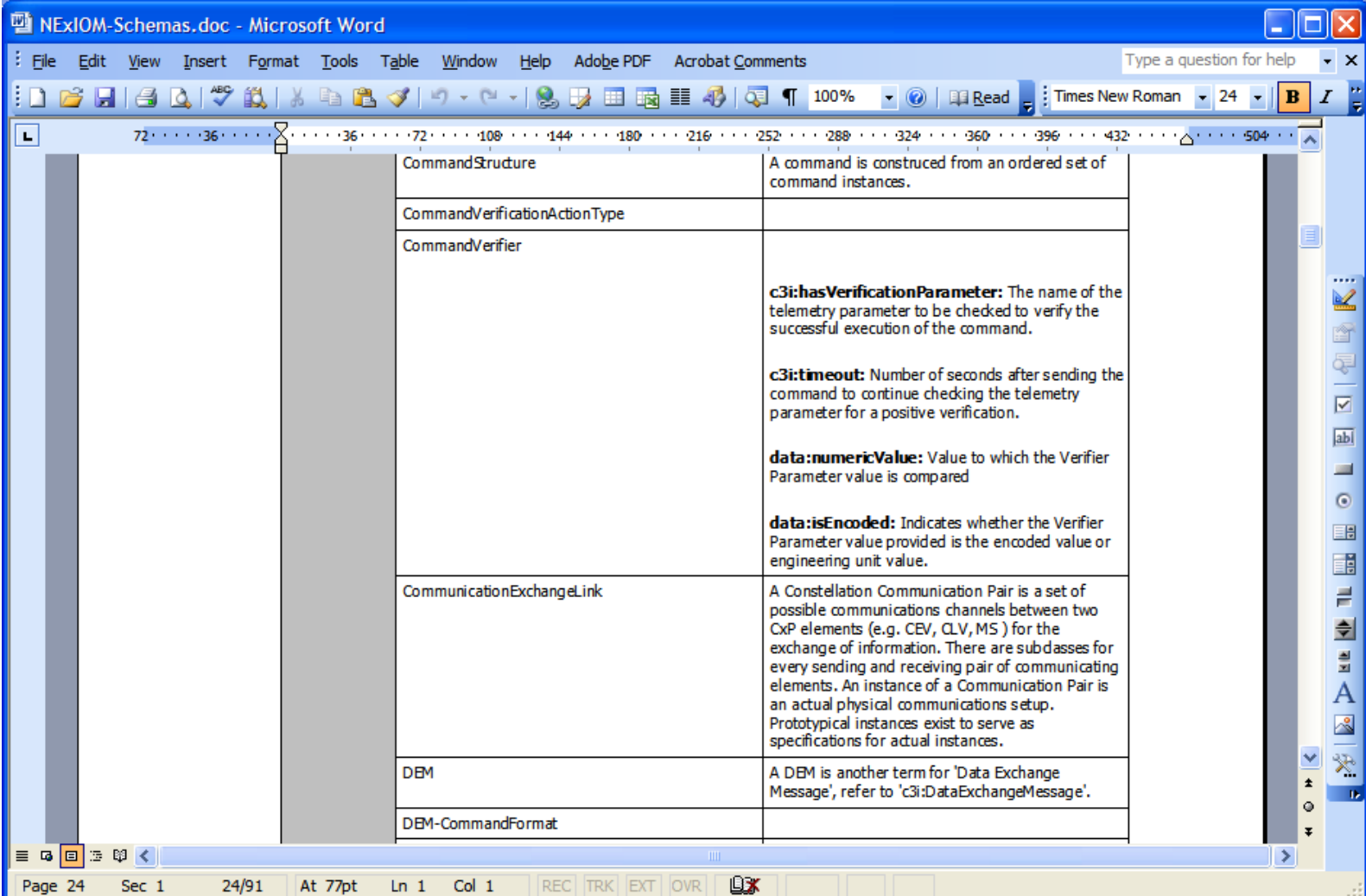
Overview

| Module Name | Input | Output |
|---|-------|--------|
| sml:ApplyConstruct | RDF | RDF |
| sml:ApplyJenaInferencing | RDF | RDF |
| sml:ApplyJenaRules | RDF | RDF |
| sml:ApplyOWLIM | RDF | RDF |
| sml:ApplyPellet | RDF | RDF |
| sml:ApplyResourceConstructors | RDF | RDF |
| sml:BindBySelect | RDF | RDF |
| sml:BranchByAsk | RDF | RDF |
| sml:ConvertRDFToXML | RDF | XML |

Done My Computer 100%

- Content driven by SPARQL queries.
- Layout defined by JSP template.

Semantic Java Server Pages (3)



| | |
|-------------------------------|--|
| CommandStructure | A command is constructed from an ordered set of command instances. |
| CommandVerificationActionType | |
| CommandVerifier | <p>c3i:hasVerificationParameter: The name of the telemetry parameter to be checked to verify the successful execution of the command.</p> <p>c3i:timeout: Number of seconds after sending the command to continue checking the telemetry parameter for a positive verification.</p> <p>data:numericValue: Value to which the Verifier Parameter value is compared</p> <p>data:isEncoded: Indicates whether the Verifier Parameter value provided is the encoded value or engineering unit value.</p> |
| CommunicationExchangeLink | A Constellation Communication Pair is a set of possible communications channels between two CxP elements (e.g. CEV, CLV, MS) for the exchange of information. There are subclasses for every sending and receiving pair of communicating elements. An instance of a Communication Pair is an actual physical communications setup. Prototypical instances exist to serve as specifications for actual instances. |
| DEM | A DEM is another term for 'Data Exchange Message', refer to 'c3i:DataExchangeMessage'. |
| DEM-CommandFormat | |

Used extensively to generate all kinds of documents, deliverables

Google Maps

The screenshot shows the Eclipse IDE with the TopBraid application. The main window displays the 'Resource Form' for 'BondiBeach'. The 'Annotations' section shows 'hasCity' with a dropdown menu. The 'Other Properties' section shows 'hasAccommodation', 'hasActivity', and 'tbgeo:zoom' (set to 14). The 'rdf:type' is set to 'Beach'. A Google Map of Bondi Beach is visible on the right, with a red pin marking the location. The map shows streets like Blair St and Curlew Rd, and landmarks like Royal Sydney Golf Course and Barracuff Park.

Resource Form

Name: BondiBeach

Annotations

- hasCity

Other Properties

- hasAccommodation
- hasActivity
- tbgeo:zoom: 14
- rdf:type: Beach

Instances

| [Resource] | rdf:type | rdfs:label | rdfs:comment |
|------------|----------|--------------|--------------|
| BondiBeach | Beach | | |
| Cairns | City | Cairns Label | |
| Canberra | Capital | | |

**Resources with
geo:lat geo:long
values**

Calendars

The screenshot shows a web-based calendar interface for July 2007. The interface includes a navigation bar with tabs for 'Instances', 'Rules', 'Domain', 'Relevant Properties', 'SPARQL', 'Imports', and 'Calendar'. The calendar grid shows days from Sunday to Saturday. Several events are marked with diamond icons: 'WOY_W...' on Monday the 9th, 'DANDEN...' on Tuesday the 10th, and 'WETHERI...' on Wednesday the 18th. A tooltip is displayed over the 18th, containing the text: 'WETHERILL_PARK(NSW) House-4/2: UNIQUE CAPE COD auction date = 2007-07-18T00:30:09.25Z'.

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----------------|-------------------|--|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 ◆ WOY_W... | 10 ◆ DANDEN... | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 ◆ WETHERI... | 19 | 20 | 21 |
| 22 | 23 | 24 | WETHERILL_PARK(NSW) House-4/2: UNIQUE CAPE COD auction date = 2007-07-18T00:30:09.25Z | | | 28 |
| 29 | 30 | 31 | | | | |
| | | | | | | |



BIRT Reports



Input: Tabular data from SPARQL queries



TopBraid Ensemble (1)

TopBraid Ensemble

TopQuadrant

Concept Navigator

- Agent (359)
- Query (2)
- ▼ Real estate (6)
 - Apartment (114)
 - House (343)
 - Townhouse (2)
- ▼ person:Person (1)
 - Customer
- ▶ world:Address (1)
- world:Country (5)

Results

| Name | Location | Bedroom | Bathroom | Price (\$A) |
|--|-------------------|---------|----------|-------------|
| TORQUAY(VIC) House-4/2: International Real Estate Network | TORQUAY VIC | 4 | 2 | 367785.0 |
| PACIFIC_PINES(QLD) House-4/2: Attention First Home Buyers. | PACIFIC PINES QLD | 4 | 2 | 369000.0 |
| KALLANGUR(QLD) House-3/2: ENTERTAINERS DELIGHT | KALLANGUR QLD | 3 | 2 | 369000.0 |
| TORQUAY(VIC) House-4/2: Hotham 24E | TORQUAY VIC | 4 | 2 | 372760.0 |
| TORQUAY(VIC) House-4/2: Loxley 20E | TORQUAY VIC | 4 | 2 | 373207.0 |
| PETRIE(QLD) House-4/2: A CUT ABOVE THE REST! | PETRIE QLD | 4 | 2 | 379000.0 |
| MARANGAROO(WA) House-4/2: GREAT LOCATION SET AT THE TOP OF A | MARANGAROO WA | 4 | 2 | 379000.0 |
| TORQUAY(VIC) House-4/2: International Real Estate Network | TORQUAY VIC | 4 | 2 | 383126.0 |
| NORTH_LAKES(QLD) House-4/2: Don't Miss This One | NORTH LAKES QLD | 4 | 2 | 385000.0 |
| GOONDMINDI(QLD) House-5/2: OPEN HOUSE - SATURDAY 12TH MAY, 2:3 | GOONDMINDI QLD | 5 | 2 | 385000.0 |

Geography



WEMBLEY(WA) House-3/2: Picture Perfect

Label
WEMBLEY(WA) House-3/2: Picture Perfect

Bedrooms 3 **Bathrooms** 2 **Carports** 3

Price (\$AU)
1050000.0

Location
WEMBLEY WA

Description
Discover modern living at 88#x2019;s best in this elegant 1940#x2019;s three bedroom, study, two bathroom home that has been extended and renovated to an excellent standard. Polished boards and warm light unite to create a stunning yet cosy lifestyle.

88#x2019;s beautifully presented, extremely well maintained and includes a magnificent sun-drenched meals/dining/family room that overlooks tranquil outdoor entertaining areas and a private back yard large enough for a pool or children#x2019;s play area. There is a

Image

Address
95 NANSON STREET WEMBLEY

Agent
Property Connection Wb, Elwood

GYMPIE(QLD) House-4/2: GREAT FAMILY HOME, GREAT POSITION

Label
GYMPIE(QLD) House-4/2: GREAT FAMILY HOME, GREAT POSITION

Bedrooms 4 **Bathrooms** 2 **Carports** 2

Price (\$AU)
325000.0

Location
GYMPIE QLD

Description
*Spacious 4 bedroom brick home boasts 2 bathrooms, new kitchen, dining, cosy lounge room with wood heater
*Spacious outdoor entertainmet area is ideal for outdoor living
*Functional office
*Large double garage and rumpus room
*All located on the Southside of Gympie on a 2152 m2 block
*Presently returning \$270 per week for rent
*10,000 gal inground water tank

Image

Address
GYMPIE

Agent
Dalyellside, Gympie

Search Basket SPARQL

Location:

Bedrooms: Min Max

Bathrooms: Min Max

Price (\$AU): Min Max

Rich Internet Application for browsing and editing RDF/OWL



TopBraid Ensemble (2)

TopBraid Ensemble User: test Logout Project: kennedys.owl **TopQuadrant**

Concept Navigator

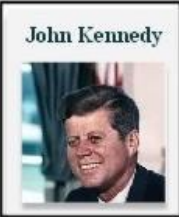
- College (19)
- Gender (2)
- ▼ ● Person (73)
 - Matriarch (1)
 - Patriarch (1)
- Profession (34)
- world:Address
- world:Area
- world:Country (5)
- world:State

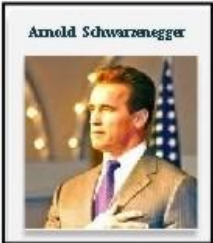
Results

| Name | Label | Comment | First name | Year of death | Year of birth |
|---------------|---------------|---------|------------|---------------|---------------|
| Alfred Tucker | Alfred Tucker | | Alfred | | 1967 |
| Alina Mojica | Alina Mojica | | Alina | | 1965 |
| Amanda Smith | Amanda Smith | | Amanda | | 1967 |
| Andrew Cuomo | Andrew Cuomo | | Andrew | | 1957 |

Jean Kennedy **Geography** **Graph** **Calendar**

Shortest Path

John Kennedy


Arnold Schwarzenegger


Search **Basket** **SPARQL**

Label:

Comment:

First name:

Year of death: Min Max

Year of birth: Min Max



TopBraid Ensemble (3)

TopBraid Ensemble User: test Logout Project: kennedys.owl **TopQuadrant**

Concept Navigator

- College (19)
- Gender (2)
- ▼ ● Person (73)
 - Matriarch (1)
 - Patriarch (1)
- Profession (34)
- world:Address
- world:Area
- world:Country (5)
- world:State

Results

| Name | Label | Comment | First name | Year of death | Year of birth |
|---------------|---------------|---------|------------|---------------|---------------|
| Alfred Tucker | Alfred Tucker | | Alfred | | 1967 |
| Alina Mojica | Alina Mojica | | Alina | | 1965 |
| Amanda Smith | Amanda Smith | | Amanda | | 1967 |
| Andrew Cuomo | Andrew Cuomo | | Andrew | | 1957 |

Jean Kennedy **Geography** **Graph** **Calendar**

```
graph TD; JohnKennedy[John Kennedy] -- has parent --> RoseFitzgerald[Rose Fitzgerald]; RoseFitzgerald -- has child --> EuniceKennedy[Eunice Kennedy]; EuniceKennedy -- has child --> ArnoldSchwarzenegger[Arnold Schwarzenegger]; ArnoldSchwarzenegger -- has spouse --> MariaShriver((Maria Shriver));
```

Search **Basket** **SPARQL**

Label:

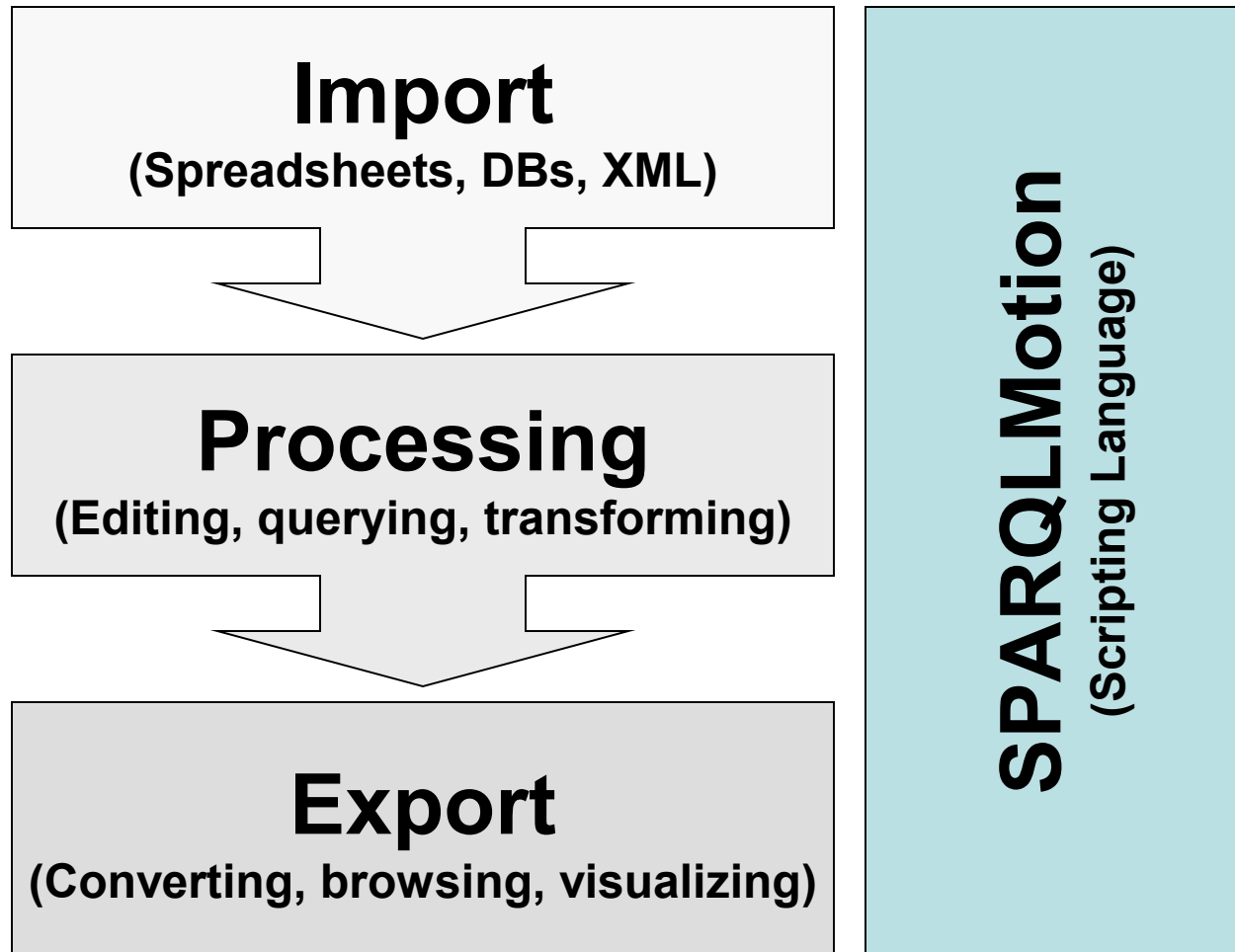
Comment:

First name:

Year of death: Min Max

Year of birth: Min Max

Structure of this Talk



SPARQLMotion

- A visual scripting language for Semantic Web Technology
- Import – Process – Export
- Use case: repeatable data processing and information integration tasks
- SPARQLMotion itself is defined as an OWL ontology
- Instance scripts can be edited with any OWL editing tool
- Has an extensible architecture



SPARQLMotion Example

The screenshot displays the SPARQLMotion application interface, which is used for creating and editing SPARQL workflows. The main window shows a workflow diagram with three nodes connected by 'sm:next' relationships:

- Load TopBraid blog**: URL = http://composing-the...
- Filter TopBraid articles**: construct query = CONSTRUCT { ?item...
- Create spreadsheet**: select query = SELECT DISTINCT ?tit...; target file path = /ZZZ/temp/spreadshee...

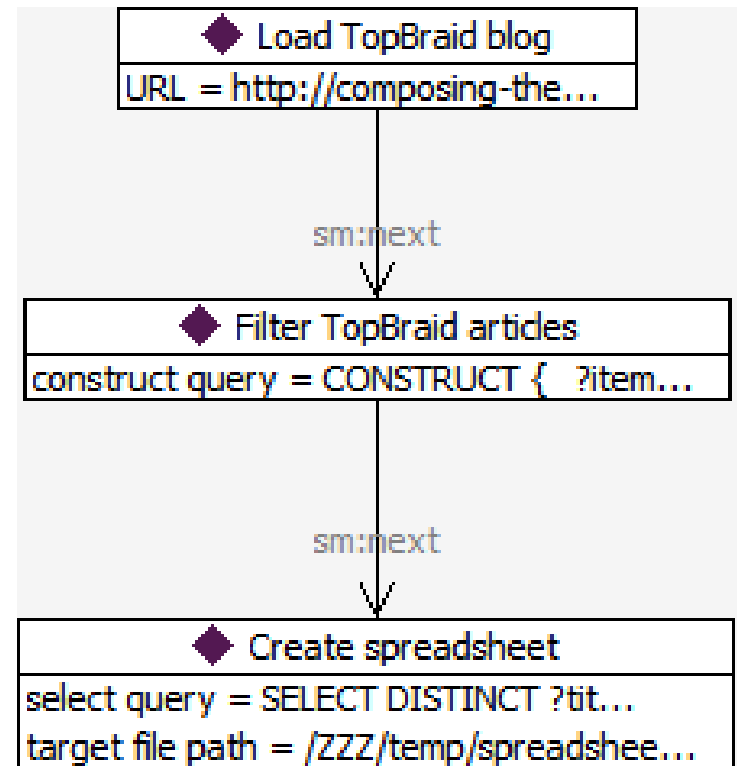
The right-hand pane shows the 'Resource Form' for the selected node, 'FilterTopBraidArticles'. The form includes the following fields:

- Name:** FilterTopBraidArticles
- Annotations:** rdfs:label Filter TopBraid articles
- Other Properties:** sm:next CreateSpreadsheet; sml:constructQuery CONSTRUCT { ?item rdfs:label ?label } WHERE { ?item rdfs:label ?label . FILTER (!regex(?label, "TopBraid")) . }
- rdf:type:** sml:FilterByConstruct

The bottom of the application shows tabs for 'Form', 'Graph', and 'Source Code', with 'Form' currently selected.

SPARQLMotion Language

- Scripts consist of modules
- Modules have a type (e.g. ApplyPellet)
- The output of one module is the input to its successors (RDF, XML and/or variable bindings)
- Branching (if-else), Iterations (while) and merging supported

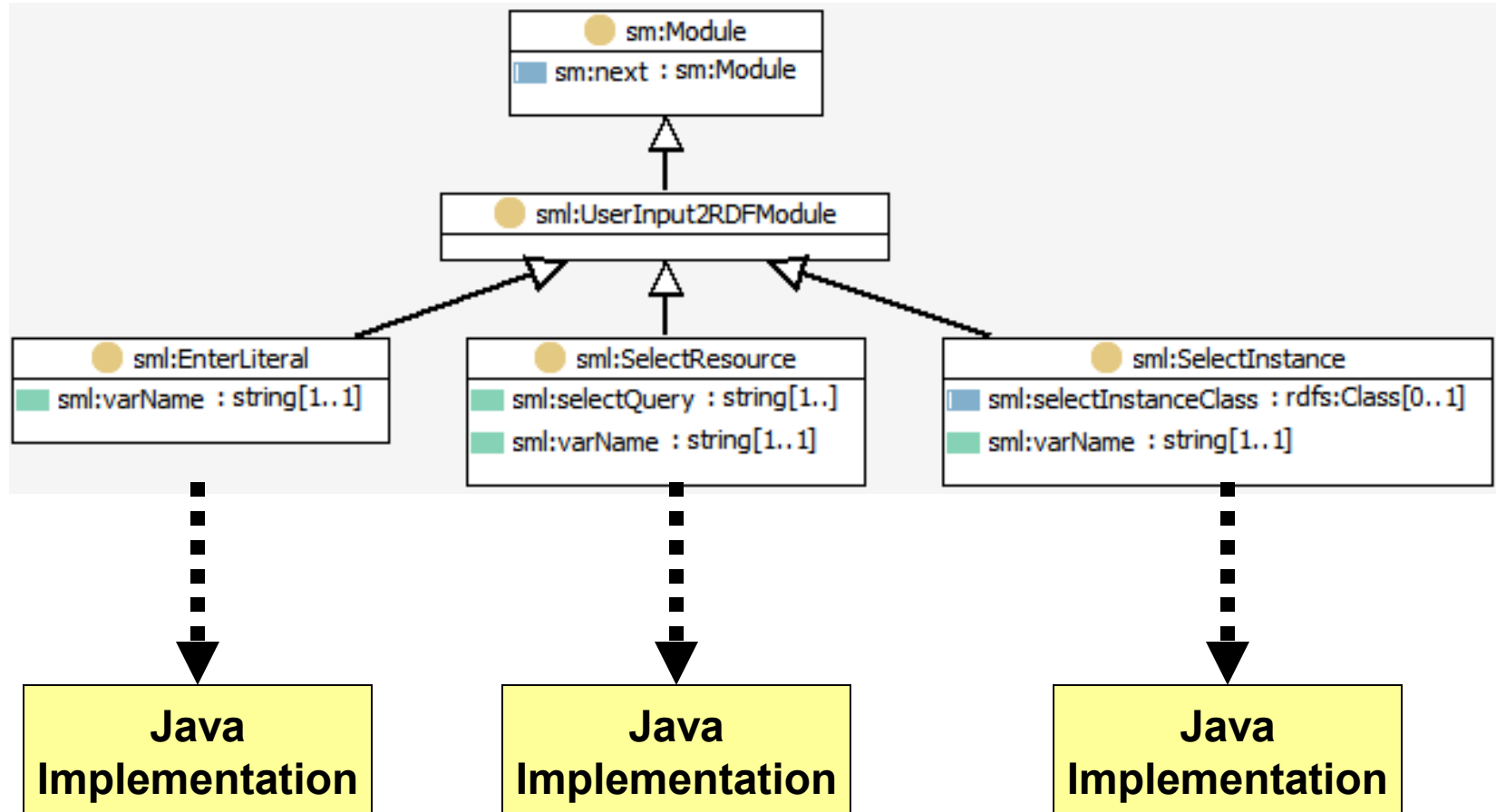


SPARQLMotion Module Types

- Input
 - Something-to-RDF
 - Something-to-XML
 - User Input
- Processing
 - RDF-to-RDF
 - XML-to-RDF
 - RDF-to-XML
- Output
 - RDF-to-Output
 - XML-to-Output



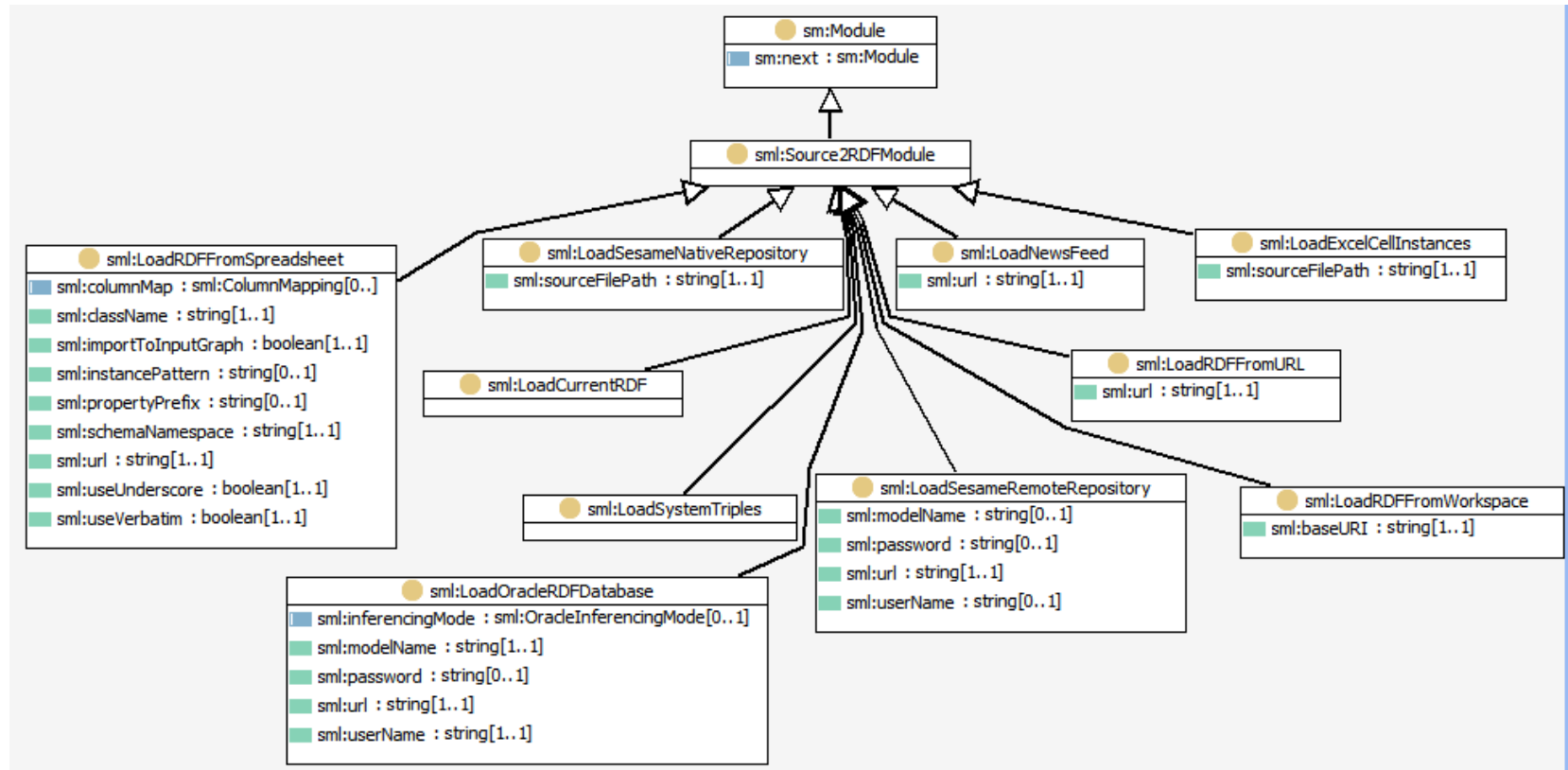
SPARQLMotion Module Library (1)



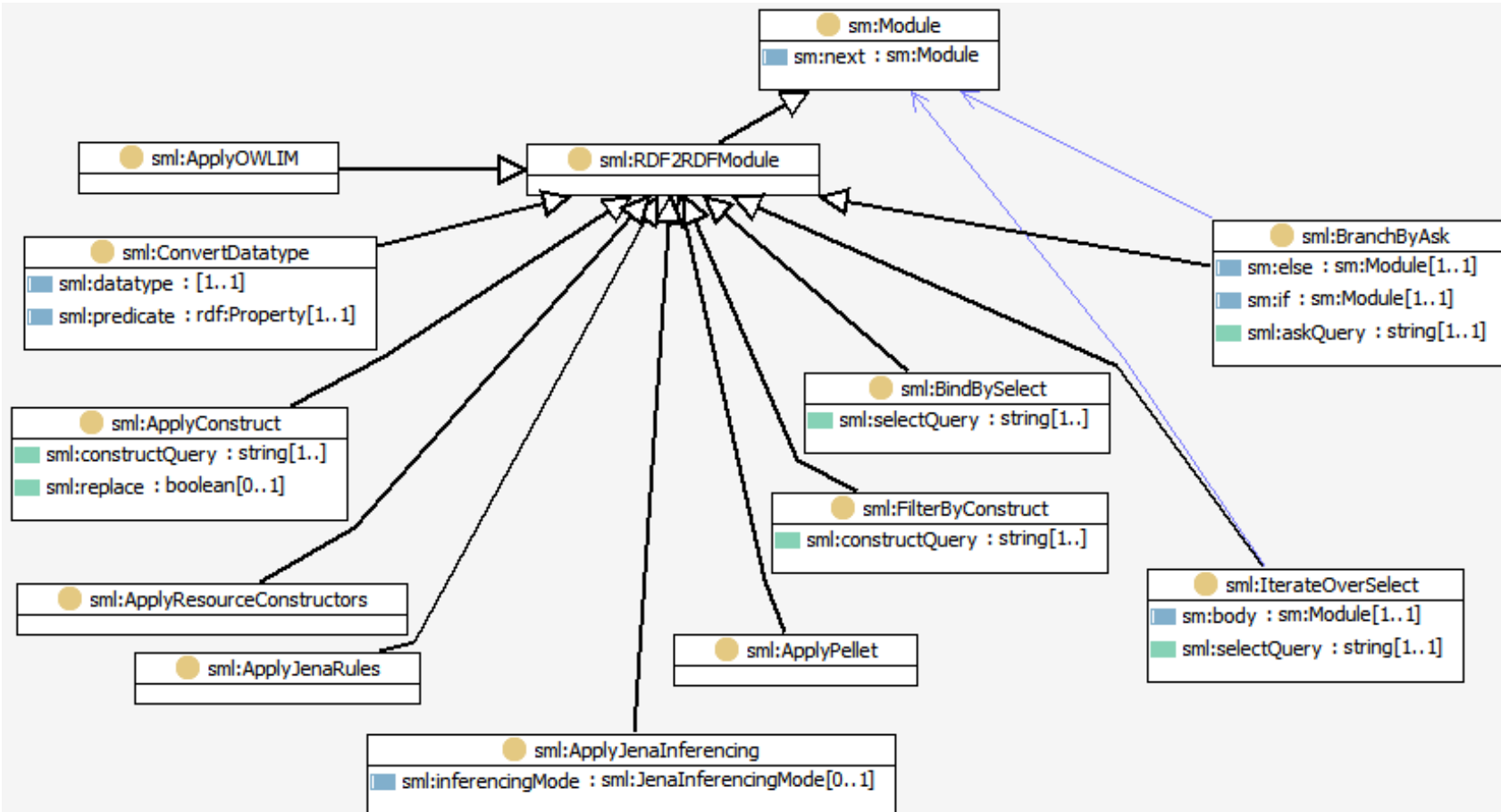
OWL Representation is backed by Java classes in execution engine
Module implementations are plug-ins to the engine



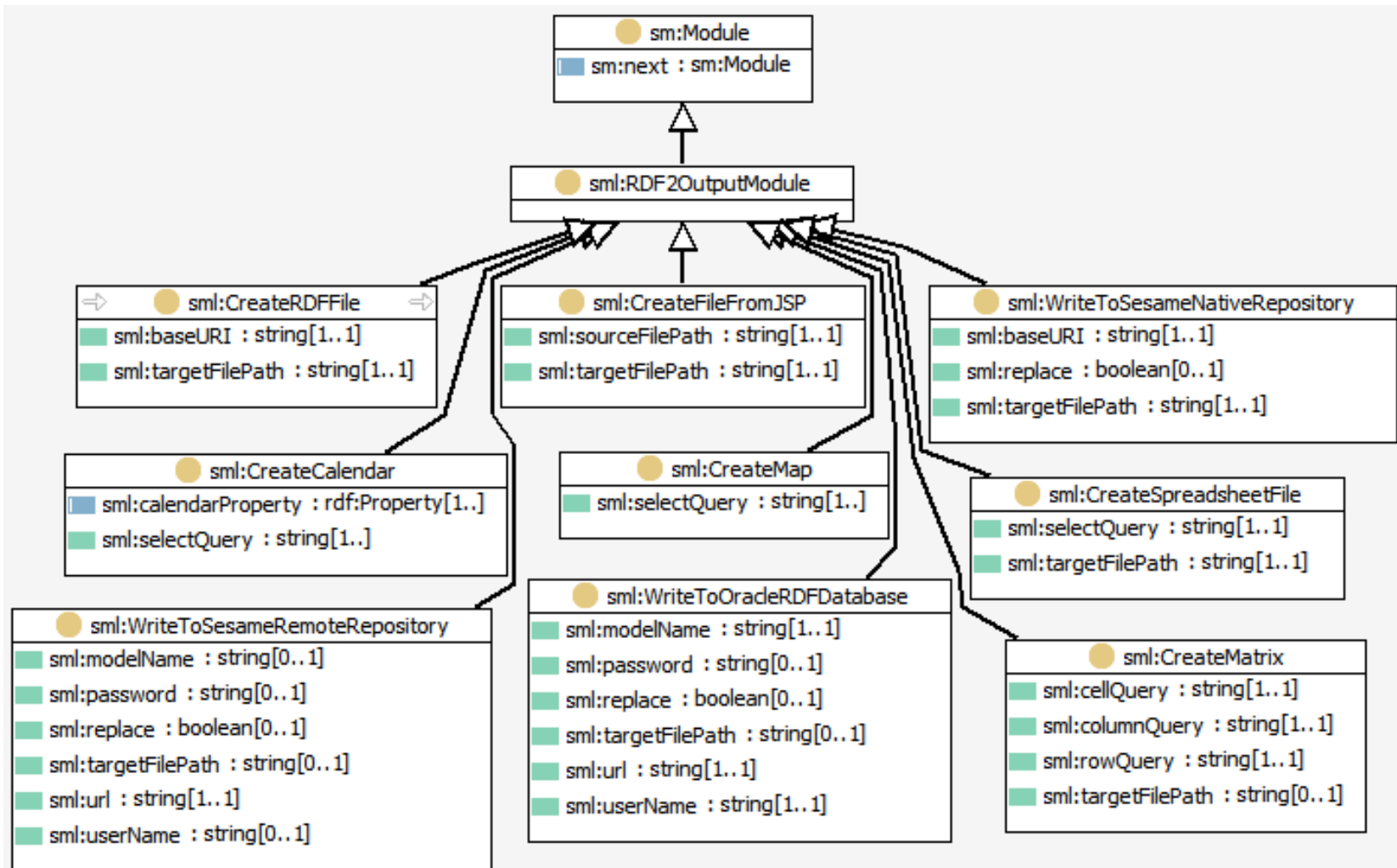
SPARQLMotion Module Library (2)



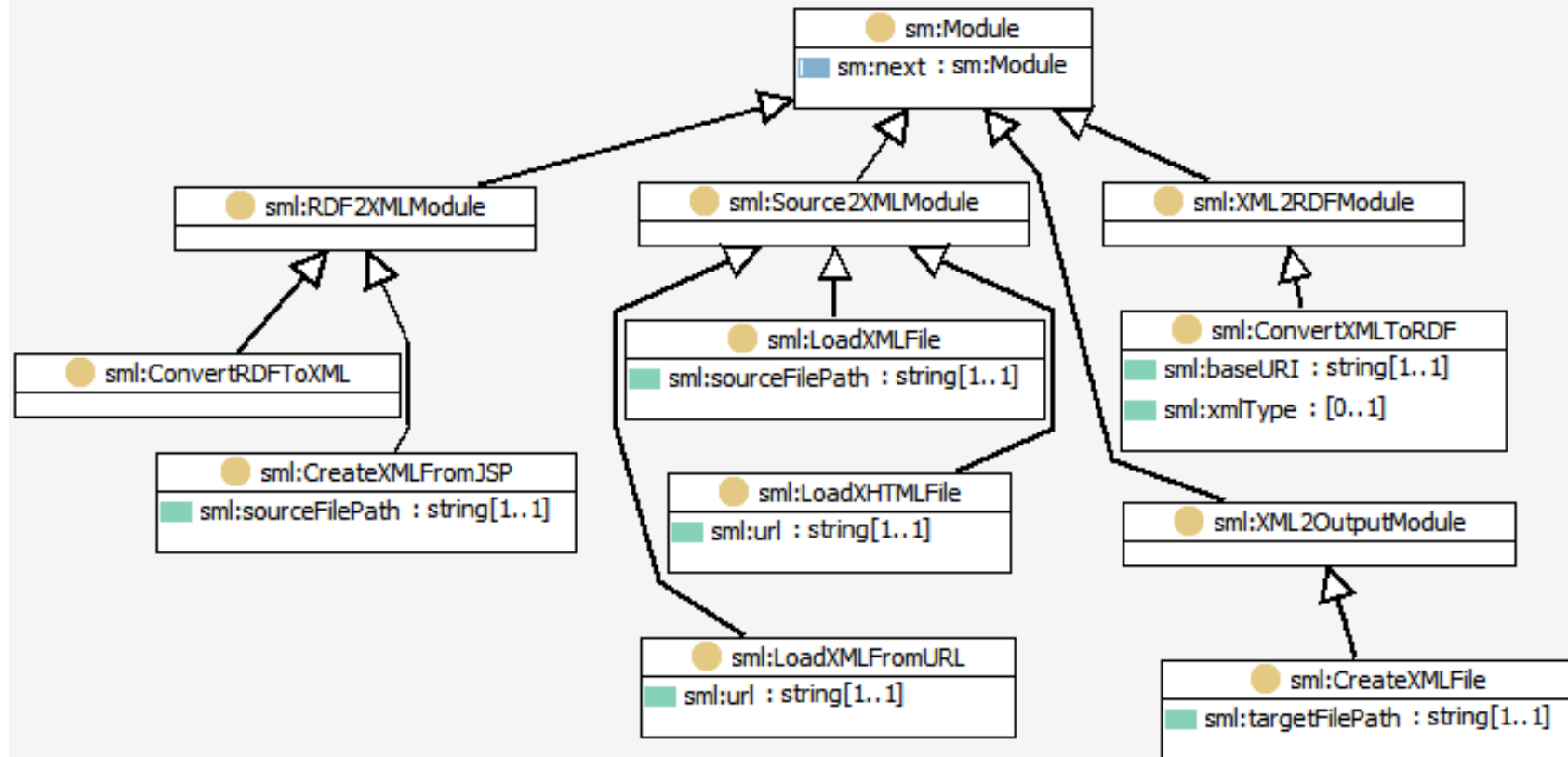
SPARQLMotion Module Library (3)



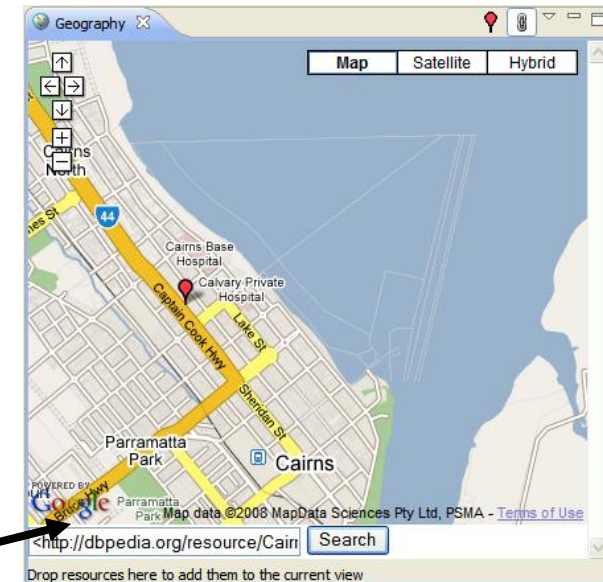
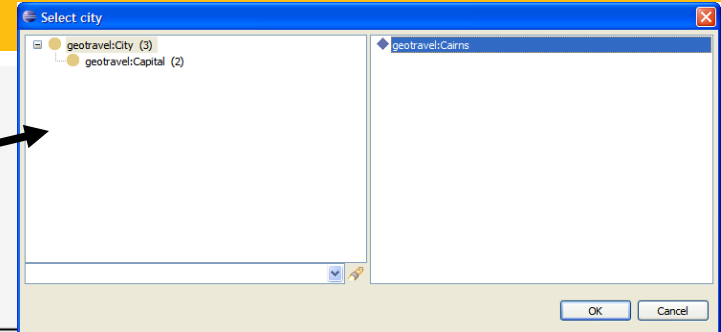
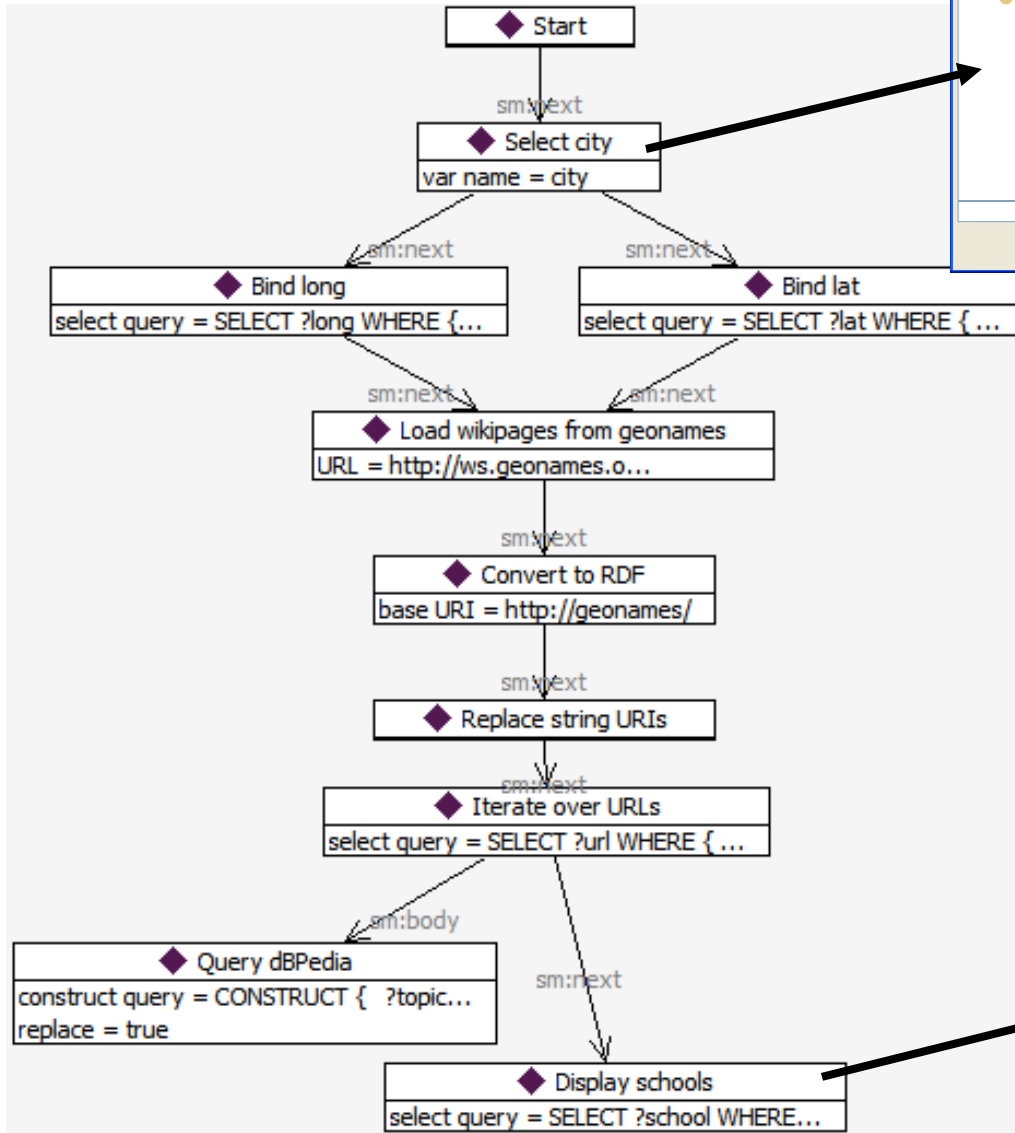
SPARQLMotion Module Library (4)



SPARQLMotion Module Library (5)



Complex SPARQLMotion Example



SPARQLMotion Use Cases

- Convert files to databases
- Combine multiple RSS feeds
- Create spreadsheets and charts
- Run periodic background checks
- Create XML input for other tools
- Control web pages
- Create maps and calendars
- Run inferences periodically
- ...



Summary

- Semantic Web languages are an attractive foundation for data integration tasks
- Generic methods and tools can be used, exploiting ontological metadata
- The TopBraid Suite product family is a comprehensive solution covering import, processing and export.



Extra Slides



TopBraid Suite

