Geosemantics & Open Source
@ IFGI, 52°North

Patrick Maué
Institute for Geoinformatics (IFGI)
pajoma@uni-muenster.de

Semantic Annotations API

http://purl.org/net/sapience/docs

Lightweight Java API supporting the dynamic and standard-compliant injection of references to external shared vocabularies.

Semantic Annotations Wrap Up

- **Semantic annotations**
  real world associations between (meta)data and shared vocabularies (addressable via URLs)
- **Semantic annotations in OGC Standards**
  Discussion Paper OGC 08-167r1
  - Best Practice Solutions to add references to existing standards (GML Schema, GML, KML, SensorML, OWS Common, …)
  - focus on compliance (re-using existing features in standards)

Examples

KML

```
<placemark>
    <name> some river </name>
    <ExtendedData>
        <Data name="urn: … modelReference">
            <value>http://sws.geonames.org/3020251/about.rdf</value>
        </Data>
    </ExtendedData>
    <Point> ....
```

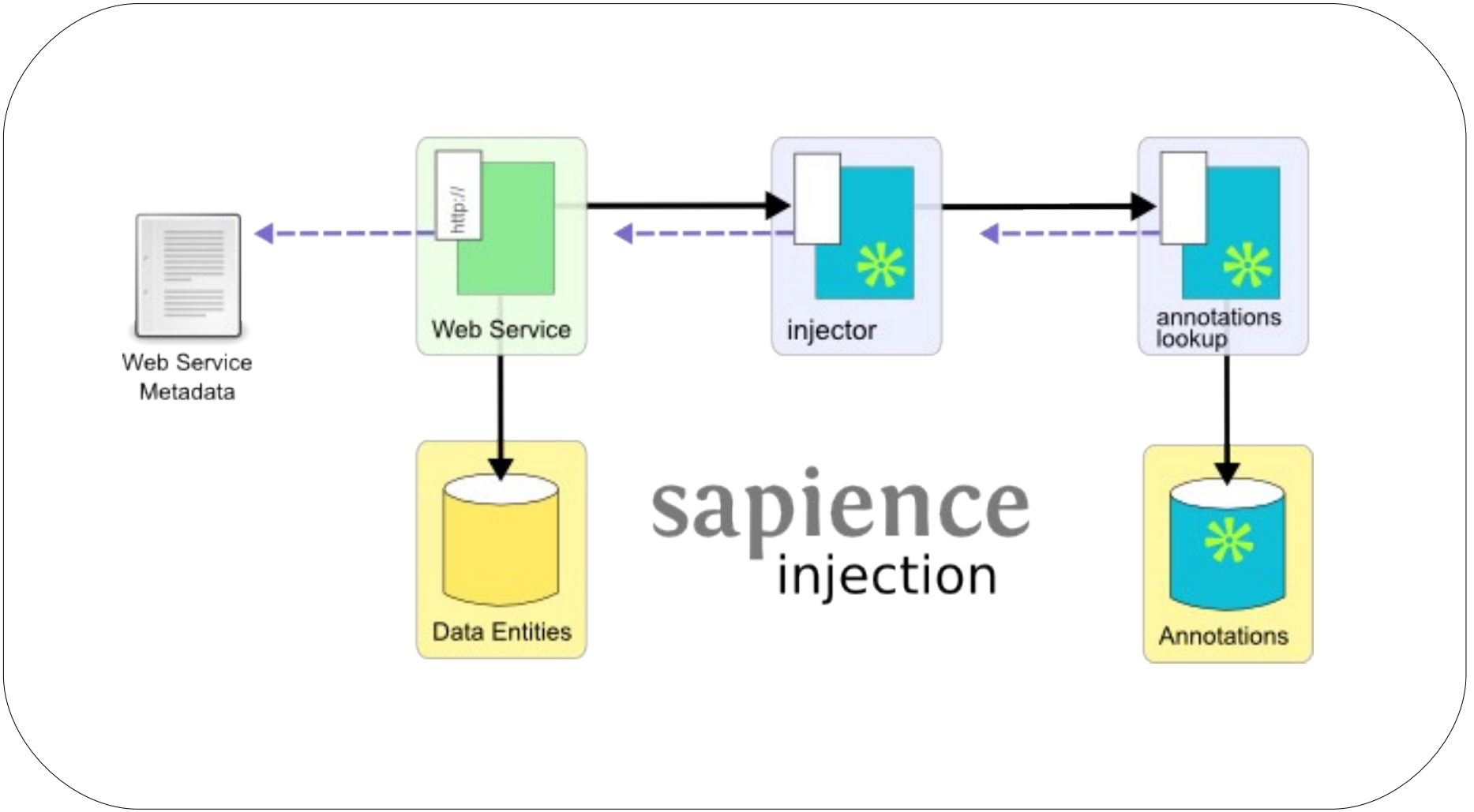
GML

```
<om:Observation>
    <om:observedProperty
        xlink:href="urn:ogc:def:phenomenon:WaterStage">
    <om:Procedure
        xlink:href="http://.../rdf/Gauge04"
        xlink:arcrole="urn:...:modelReference">
...
```

GML Schema

```
<element name="rivers"
    type  ="RiverType"
    sawsdl:modelReference="http://.../rdf/River">
<complexType name="banksType">
  …
```

Injecting external references into metadata



Status & Outlook

- Implementation ongoing
- Integration with deegree Framework (GDI-Grid Project)
  - Running version planned for February 2010
  - Focus on WSDL/WSRF (Grid context)
- What is needed
  - Feedback on best practices
  - URNs for references (domainReference, gazetteerReference, modelReference, ...?)

Concept Repository

http://purl.org/net/sapience/docs
http://purl.org/net/concepts/

Moving from an ontology repository to the more fine-grained and context-sensitive concept repository.

Background

- Ontology engineering tasks in projects Swing and GDI-Grid
- Ontology Repositories should support
  - fine-grained access to the models
  - filter mechanisms to select relevant concepts only
  - context-aware access methods
  - ontology evolution (and access to prior versions)
  - Consistency checks for relations
- Implementing“Best Practice Recipes for Publishing RDF Vocabularies” (W3C, http://www.w3.org/TR/swbp-vocab-pub/)

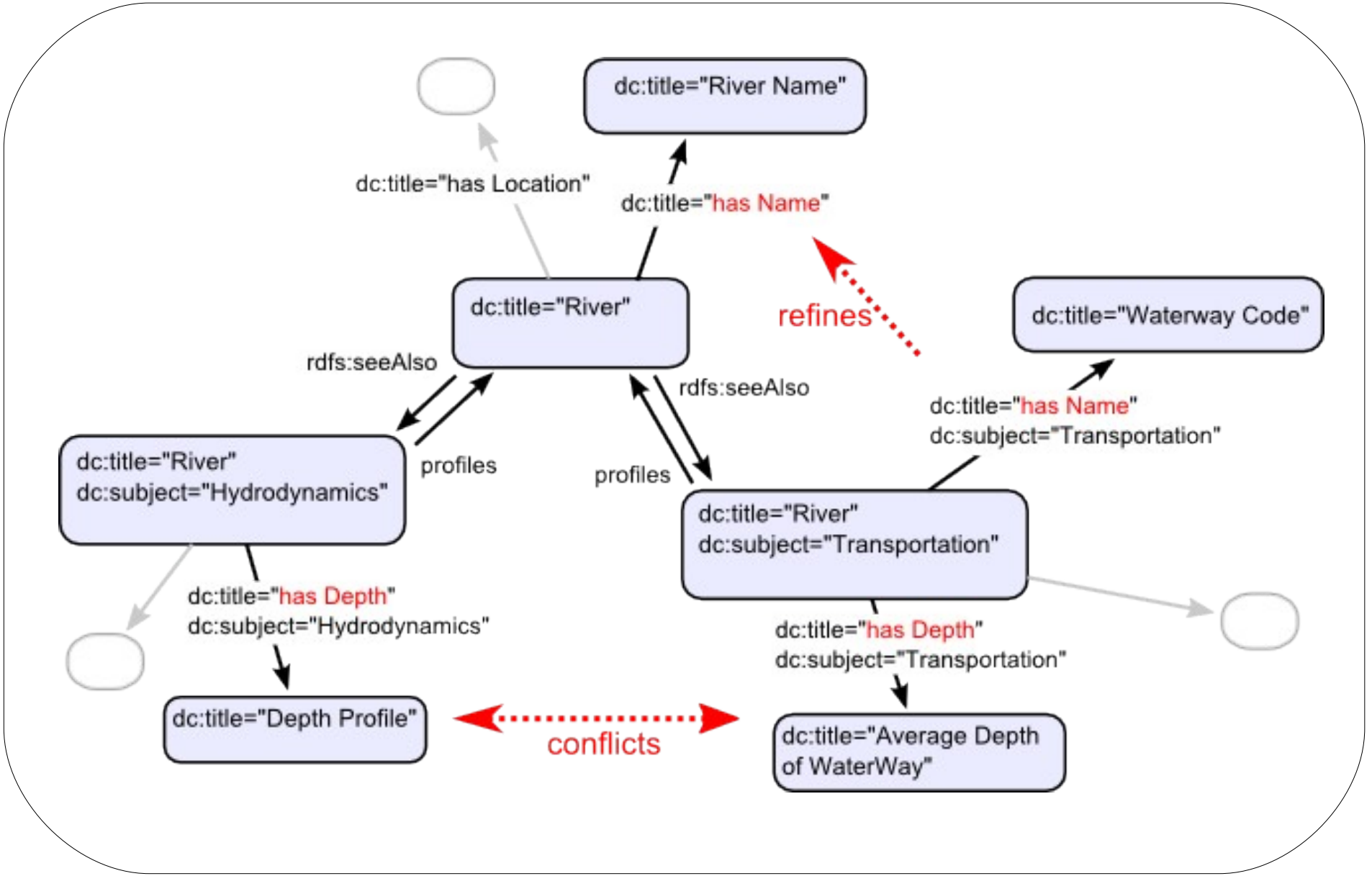
Addressing concepts

URL representing the concept identifier

- http://.../rdf/River
- http://.../rdf/subject/Transportation/River
- http://.../rdf/River&subject=Transportation
- http://.../rdf/River_Transportation

→ last three are equivalent and return a (domain dependent) description of the concept River

Concept Profiling



Query Actions (supporting selection)

- http://…/rdf/describe?title=River
  - see also: http://.../rdf/River
- http://…/rdf/all?subject=Transportation
  - see also: http://.../rdf/subject/Transportation/
- http://…/rdf/neighbors?title=River&subject=Transportation

→ keywords trigger SPARQL queries
(support for custom actions)

SWRLSpatial

52north
exploring horizons

Spatial Built-ins for SWRL

Supporting spatial analysis and topology tests within the ontologies.

Set of boolean and arithmetic topological operators

name(?param,...)

- equals
- disjoint
- contains
- overlaps

Derived operations (touches, within, crosses,...)

name(?param,.., ?result)

- Boundary
- Distance
- Area,Length,Perimeter
- MBR,ConvexHull
- Buffer
- Intersection,Union,Difference

...

Inspired by OGC Simple Features Access (06-103r3), GeoSPARQL (09-157),

SWRL Spatial examples

- Set a river as navigable for my ship if its depth exceeds 5 meter.

  River(?river) ^ hasDepth(?river,?depth) ^ swrlb:greaterThan(?depth,5)
  → isNavigable(?river,true)

- Select all roads which cross the river Rhine

  hasGeometry(Rhine,?river_geom) ^ Road(?road) ^ hasGeometry(?road,?road_geom) ^ swrls:crosses(?river_geom,?road_geom)
  → sqwrl:select(??road)

SWRL Spatial examples (con'd)

- Identify potential flood risk for buildings.

  River(?river) ^ Building(?building) … swrls:buffer(?river_geom, ?result, 100) ^ swrls:contains(?result,?building_geom)
  → RiskedBuilding(?building)

- Query OpenStreetMap to populate ontology with river features.

  osm:query("http:..") ^ OSM_Feature(?x) ^ hasTag(?x, "River")
  → River(?x)

- Query WFS and compute convex hull of the resulting feature collection

  ogc:query-wfs("http://...", ?filter, ?query-result) ^ swrls:convexHull(?query-result, ?hull) ^ ...

SWRL Issues

- Scalability difficult
  (work on feature collections if possible)
- Reasoner support could be improved
  (Jess, IRIS, PELLET,...)
- Source Code (querying SOS and OSM, implemented built-in for contains) available in 52°N repository
- Alignment with GeoSPARQL

Wrapping Up

- Documentation of Sapience and Core (and these slides)
  http://purl.org/net/sapience/docs
- Documentation of SWRL Spatial (in the making)
  http://52north.org/twiki/bin/view/Semantics/SwrlSpatial