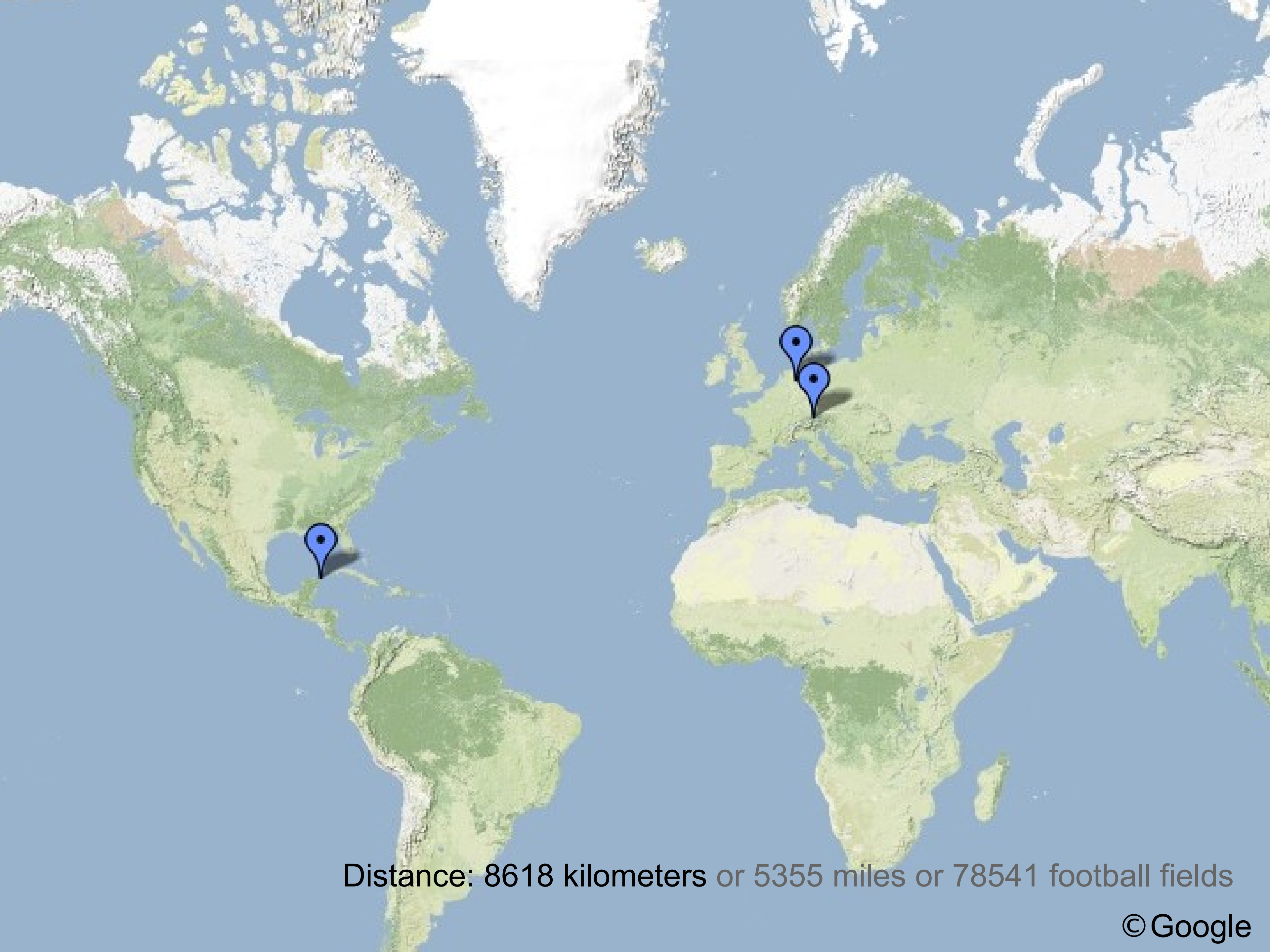
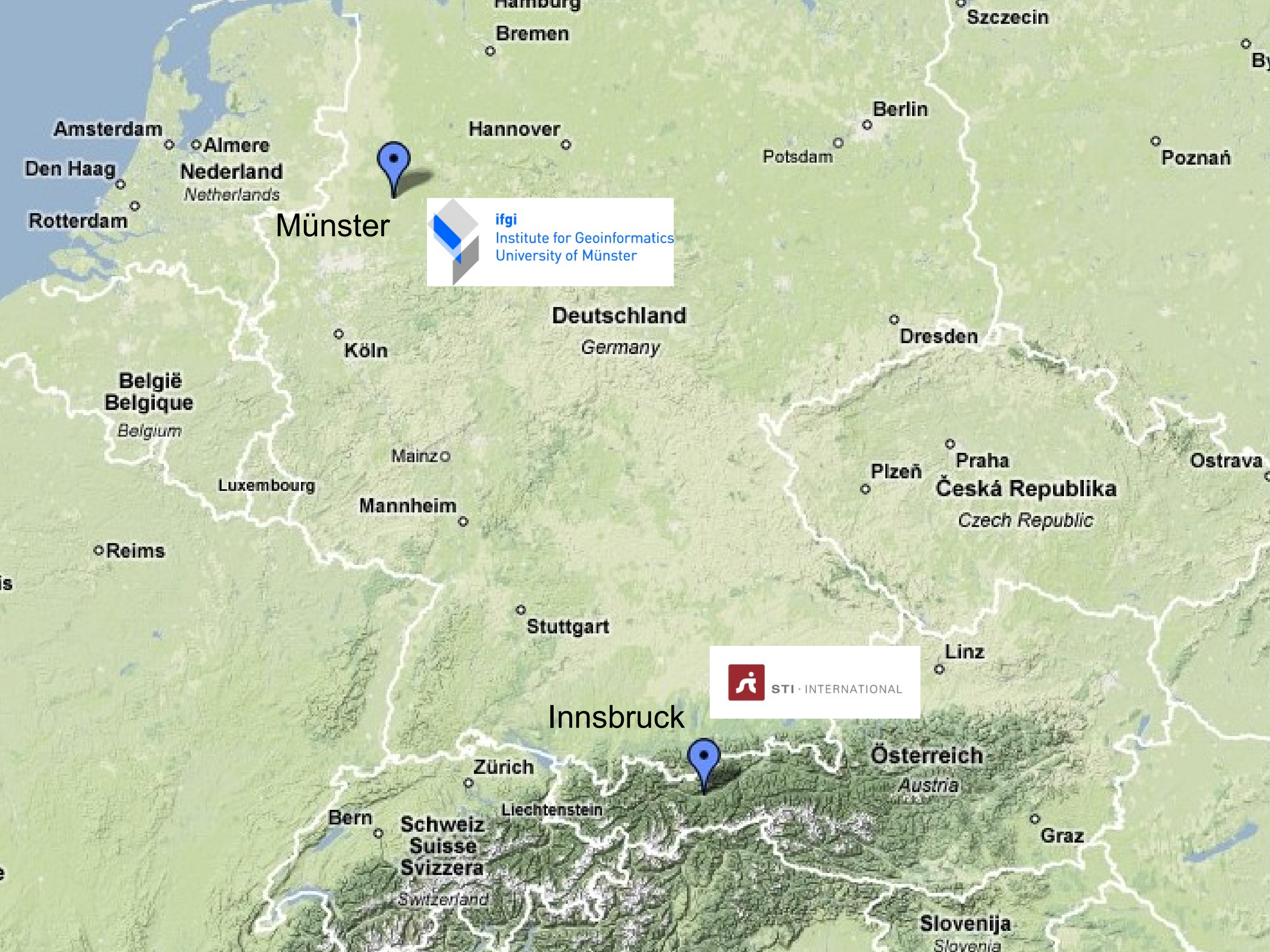
Geospatial Decision Making in the Semantic Web

Patrick Maué (IFGI)
Dumitru Roman (STI)

GEOWS 2009
February 1st, 2009 - Cancun, Mexico


Distance: 8618 kilometers or 5355 miles or 78541 football fields
© Google


Münster
ifgi
Institute for Geoinformatics
University of Münster
Innsbruck
STI · INTERNATIONAL
Bremen
Szczecin
Berlin
Hannover
Potsdam
Poznań
Amsterdam
Almere
Den Haag
Nederland
Netherlands
Rotterdam
Deutschland
Germany
Dresden
Köln
België
Belgique
Belgium
Mainz
Praha
Plzeň
Ostrava
Česká Republika
Czech Republic
Luxembourg
Mannheim
Reims
Stuttgart
Linz
Österreich
Austria
Zürich
Liechtenstein
Bern
Schweiz
Suisse
Svizzera
Switzerland
Graz
Slovenija
Slovenia

We are: STI International

- Semantic Technologies Institute International
- Research in:
  - Knowledge representation
  - Semantic Web
  - Service-Oriented Computing

STI · INTERNATIONAL

We are: IFGI

- Institute for Geoinformatics at University of Münster, Germany
- Research in
  - Semantic Interoperability
  - Spatial Assistance Systems
  - Sensor Web and Geoprocessing
  - Environmental Measurement and Spatio-temporal Modeling

ifgi
Institute for Geoinformatics
University of Münster

Background: the SWING project

Semantic Web Services Interoperability for Geospatial Decision Making

Background: the SWING project

- Presented results mostly outcome of this project
- 3 Years until 02.2009
- Seven Partners, including users (BRGM), companies (ERDAS) and research (DERI, JSI, SINTEF)
- BRGM Use Cases further discussed later
- http://www.swing-project.org (with all deliverables and publications)

And who are you?

Tutorial Objective

- Demonstrating the SWING framework
- Geospatial Semantic Web
  - Discussing Potential Applications
  - Showing potential benefits
- Raise your interest

Overview

- **Spatial Data Infrastructures**
- Semantic Web Services
- Bridging the gap: Semantic Annotations
- SWING: use cases

- SWING: Developed Tools
- Demonstration (Videos)
- Hands-On Session


Spatial Data Infrastructures

Patrick Maué (IFGI)

GEOWS 2009
February 1st, 2009 - Cancun, Mexico

Overview

- Introducing Spatial Data Infrastructures
- The European INSPIRE Directive
- SDI Standards
- Examples of standard Web Services
- SDI Applications: Geospatial Decision-Making
- Web Service Compositions for Complex Tasks
- Contemporary Problems of SDIs

What is SDI?

- Transition from desktop GIS to distributed services
- Why we need SDI?
  - Keeping data up-to-date
  - Discovery and Evaluation (cross-country)
  - Resource-intensive
  - Billing and Security

What is SDI?

- Transition from desktop GIS to distributed services
- Why we need SDI?
  - Keeping data up-to-date
  - Discovery and Evaluation (cross-country)
  - Resource-intensive
  - Billing and Security
- SDI rebuild all functionality of GIS
- SDI framework provides basis for
  - Finding and Accessing spatial data
  - Describing and Evaluating spatial data
- Applications built on top of SDIs

INSPIRE in Europe

- Different data formats and quality standards across national borders
- Some Principles (Excerpt)
  - It must be possible to combine spatial data from different sources across the EU seamlessly and share them between several users and applications.
  - It should be easy to discover available spatial data, evaluate their fitness for purpose and know the conditions applicable to their use.
- Harmonisation (Standards!) required

Standards in SDIs

- Open Geospatial Consortium (OGC)
- Currently 372 Members (Companies, Universities, ...)
- Provides open standards
- Implementation guidelines for all SDI components
  - Geospatial Web Services
  - Geospatial Data
- OGC conformal Web Services can interoperate, regardless the intended applications and the served data.

OGC Standards: OWS Common

- OGC Conformal Web Service specifies
  - Access over HTTP (with KVP-encoding)
  - Minimum set of metadata
  - The getCapabilities operation
  - Exceptions handling
  - ...
- Normative reference for all OGC standards

OGC Standards: WFS

- Web Feature Service Interface
  - Specifies interface to retrieve Geographic Features
  - Data coming, for example, as Points, Lines, and Polygons
  - Features have Geometries and additional attributes
  - Data model specified in Feature Type Schema

WFS output example

- Feature Type Zico_region

```
<element name="Zico_region"
         type="con:Zico_regionType"
         substitutionGroup="gml:_Feature" />
<complexType name="Zico_regionType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="msGeometry" type="gml:GeometryPropertyType"/>
        <element name="REGIONAL" type="string"/>
        <element name="NATIONAL" type="string"/>
        <element name="LIBELLE" type="string"/>
        <element name="TYPE" type="string"/>
        <element name="LA_MESURE" type="string"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

OGC Standards: WFS

- Web Feature Service Interface
  - Specifies interface to retrieve Geographic Features
  - Data coming, for example, as Points, Lines, and Polygons
  - Features have Geometries and additional attributes
  - Data model specified in Feature Type Schema
  - Geographic Features with Geometry and arbitrary attributes
- WFS data encoded in OGC GML
  - XML-dialect used to encode feature-based geospatial data
  - Adapted ISO standard
- + WCS, WCTS, …
  - The storage layer of traditional GIS

OGC Standards: WPS and WMS

- Web Processing Service (WPS) Interface
  - Provides executable processes
  - No restrictions on input and output data
  - Takes role of GIS processing component

OGC Standards: WPS and WMS

- Web Processing Service (WPS) Interface
  - Provides executable processes
  - No restrictions on input and output data
  - Takes role of GIS processing component
- Web Mapping Service (WMS)
  - Visualisation of geospatial data
  - Render images in common formats
  - Can be directly integrated into websites

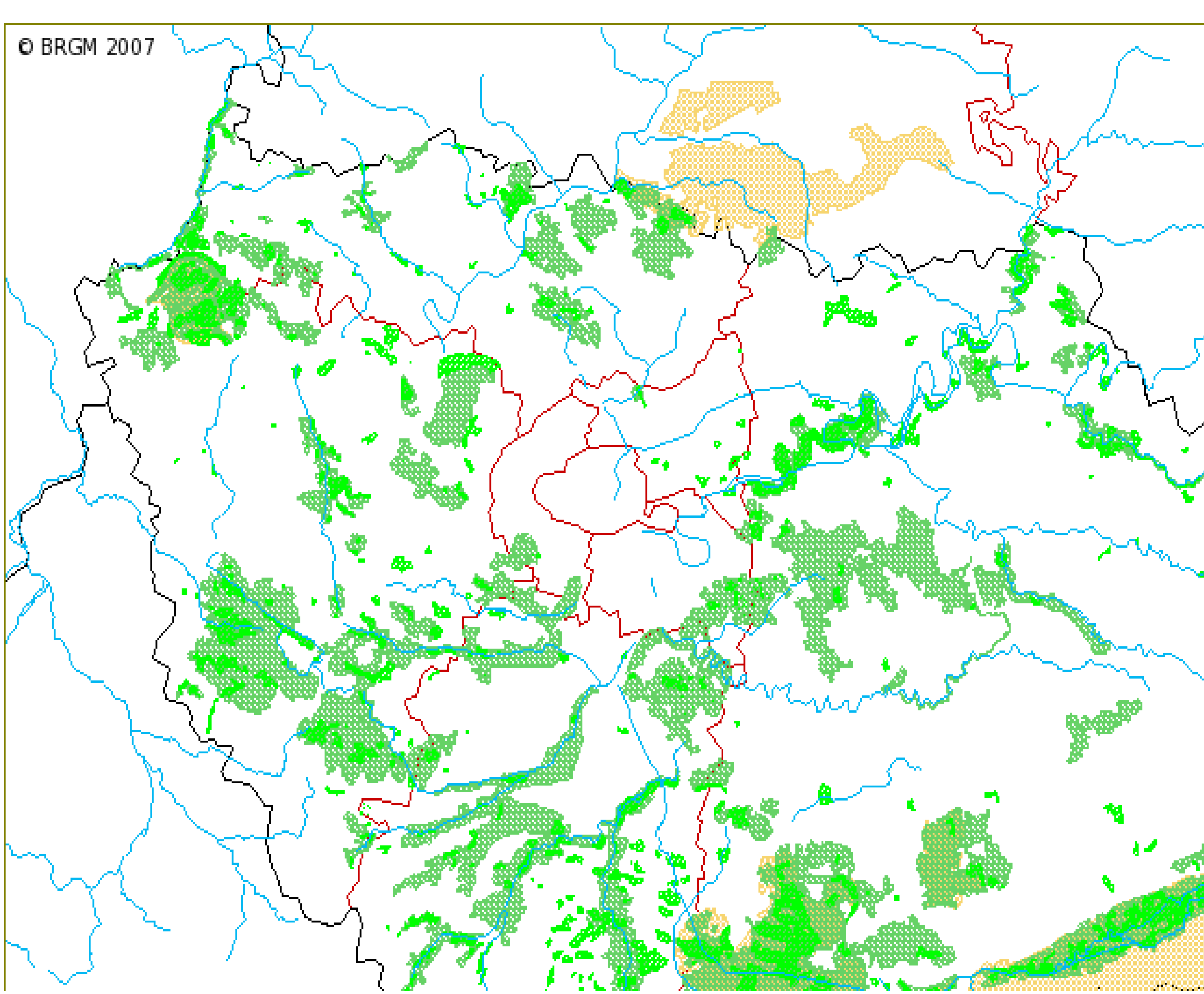
Examples

- Requesting a WFS from the Browser
- Requesting a WFS from generic GIS client

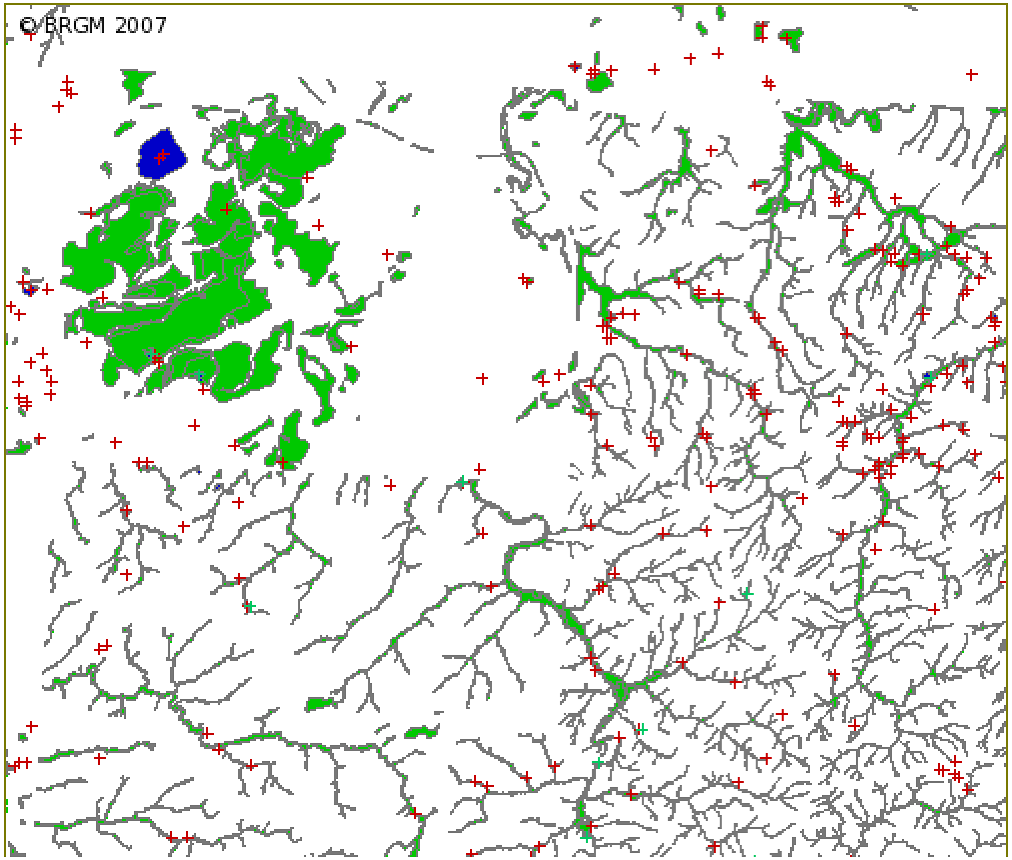
Geospatial Decision-Making

- Space as factor for decision making
- Examples: landfills, power plants, route planning
- Is a multi-criteria analysis looking at
  - Constraints (e.g. Important Bird Areas)
  - Requirements (e.g. water access)
- Requires
  - Acquisition of needed data
  - Preparation of data
  - Running the analysis
  - Rendering results for domain experts

Constraints



Requirements



Quarries With production

+ Exploitation with production

Quarries Without production

+ Exploitation

Quarries boundaries

Exploitation boundaries

Beds

Beds

Basins

Basins

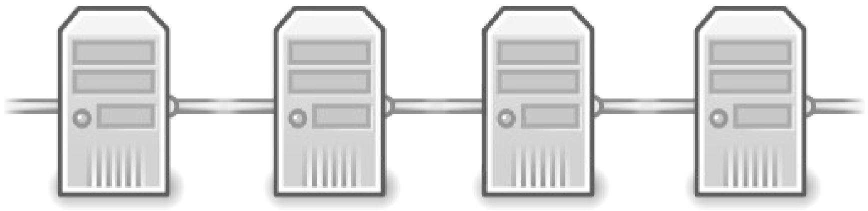
Sites

Sites

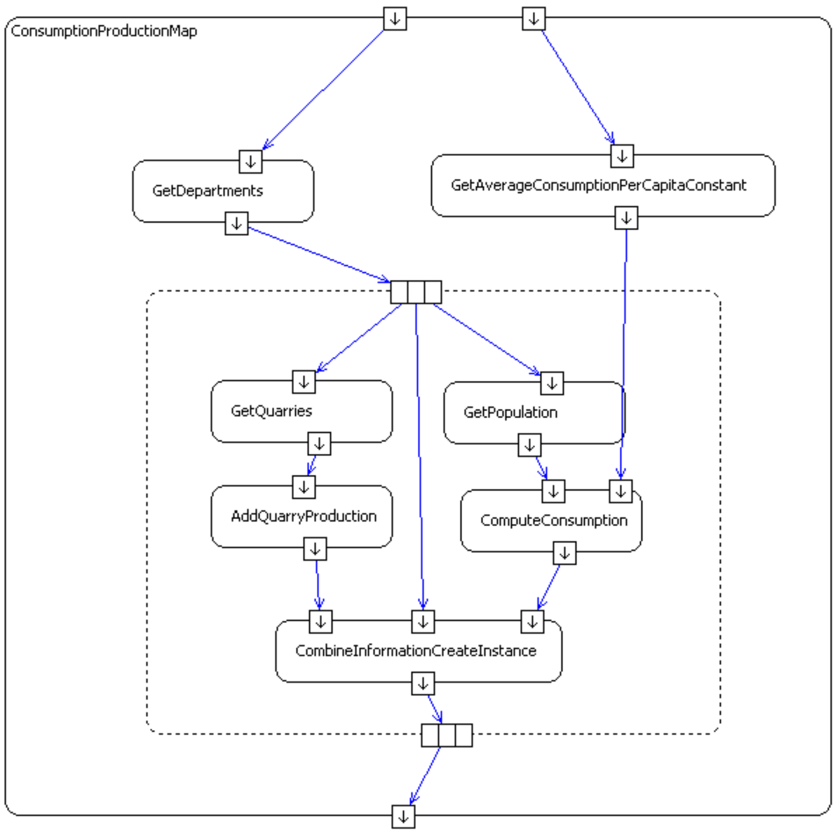
Workflows

GEOWS.2009

- Composing atomic Web Services
- Result itself a complex Web Service
- Workflow Engines
- Workflow Modeling Approaches
  - XML-based (BPEL, Wf-XML)
  - UML-based (Activity Diagrams)
  - ASM or Petri-Nets



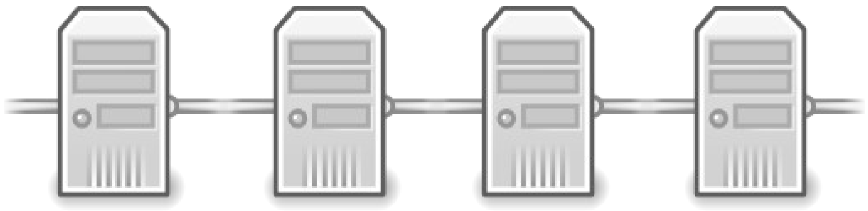
Workflow example




```
    in soa#AggregateRequest withGrounding _"http://set.sintef.no:8080/
(Aggregation/aggregate/aggregateRequest)"

      in wfs#Query
      in ogc#Intersects
      in swi#depproductionconsumption
      in gml#GeometryPropertyType

      out wfs#FeatureCollection
      //out sso#MultiplyResponse withGrounding _"http://localhost:8081
(Support/multiply/multiplyResponse)"
      out sso#MultiplyResponse withGrounding _"http://set.sintef.no:80
(Support/multiply/multiplyResponse)"
      out brgm#SocioEconomicConstantsResponse withGrounding _"http://s
wsdl#wsdl.interfaceMessageReference(SocioEconomicConstants/getValueByK
    out ins#INSEEgetPopulationByDepartmentResponse withGrounding _"htt
(INSEE/getPopulationFromRegion/getPopulationFromRegionResponse)"
    out soa#AggregateResponse

    controlled oasm#ControlState

    transitionRules _"http://www.example.org/TestCreateBoth_UoM#transi

    forall (?controlstate) with (
           ?controlstate[oasm#value hasValue oasm#InitialState] member

            forall (?constRequest,?coord) with (
                  ?constRequest memberOf brgm#SocioEconomicConstantsRe
                  ?coord memberOf gml#CoordinatesType)
            do
                  add(_#1[gml#coordinates hasValue ?coord, gml#srsName
                  add(_#2[ogc#propertyName hasValue "qua:msGeometry"] 
                  add(_#3[ogc#arguments hasValue _#1, ogc#refersTo has
                  add(_#4[ogc#encodes hasValue _#3] memberOf ogc#Filte
                  add(_#5[wfs#filter hasValue _#4,wfs#typeName hasValu
                  add(_#6[wfs#query hasValue _#5,wfs#service hasValue
virtual#GetDepartmentFeature)
            endForall

        delete(?controlstate[oasm#value hasValue oasm#InitialState])
            add(?controlstate[oasm#value hasValue oasm#DummyState])
      endForall
```

Workflows

- Composing atomic Web Services
- Result itself a complex Web Service
- Workflow Engines
- Workflow Standards
  - XML based (BPEL, Wf-XML)
  - ASM or Petri-Nets



Open Issues of SDIs

- Harmonization in between Standards
- Security and licensing
- Complexity & Performance
- **Semantic Interoperability**

Semantic Interoperability


```xml
<element name="Zico_region"
         type="con:Zico_regionType"
         substitutionGroup="gml:_Feature" />
<complexType name="Zico_regionType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="msGeometry" type="gml:GeometryPropertyType"/>
        <element name="REGIONAL" type="string"/>
        <element name="NATIONAL" type="string"/>
        <element name="LIBELLE" type="string"/>
        <element name="TYPE" type="string"/>
        <element name="LA_MESURE" type="string"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Overview

- Spatial Data Infrastructures
- **Semantic Web Services**
- Bridging the gap: Semantic Annotations
- SWING: use cases

- SWING: Developed Tools
- Demonstration (Videos)
- Hands-On Session

Semantic Web Services (SWS)

GEOWS 2009
February 1st, 2009 - Cancun, Mexico

Outline

- Tasks to be automated in SWS
- Ontologies and Web services: the WSMO approach
- WSML – the language for formalizing WSMO
- Web Service Discovery

Semantic Web and Web Services - SWS

Dynamic

Web Services
UDDI, WSDL, SOAP

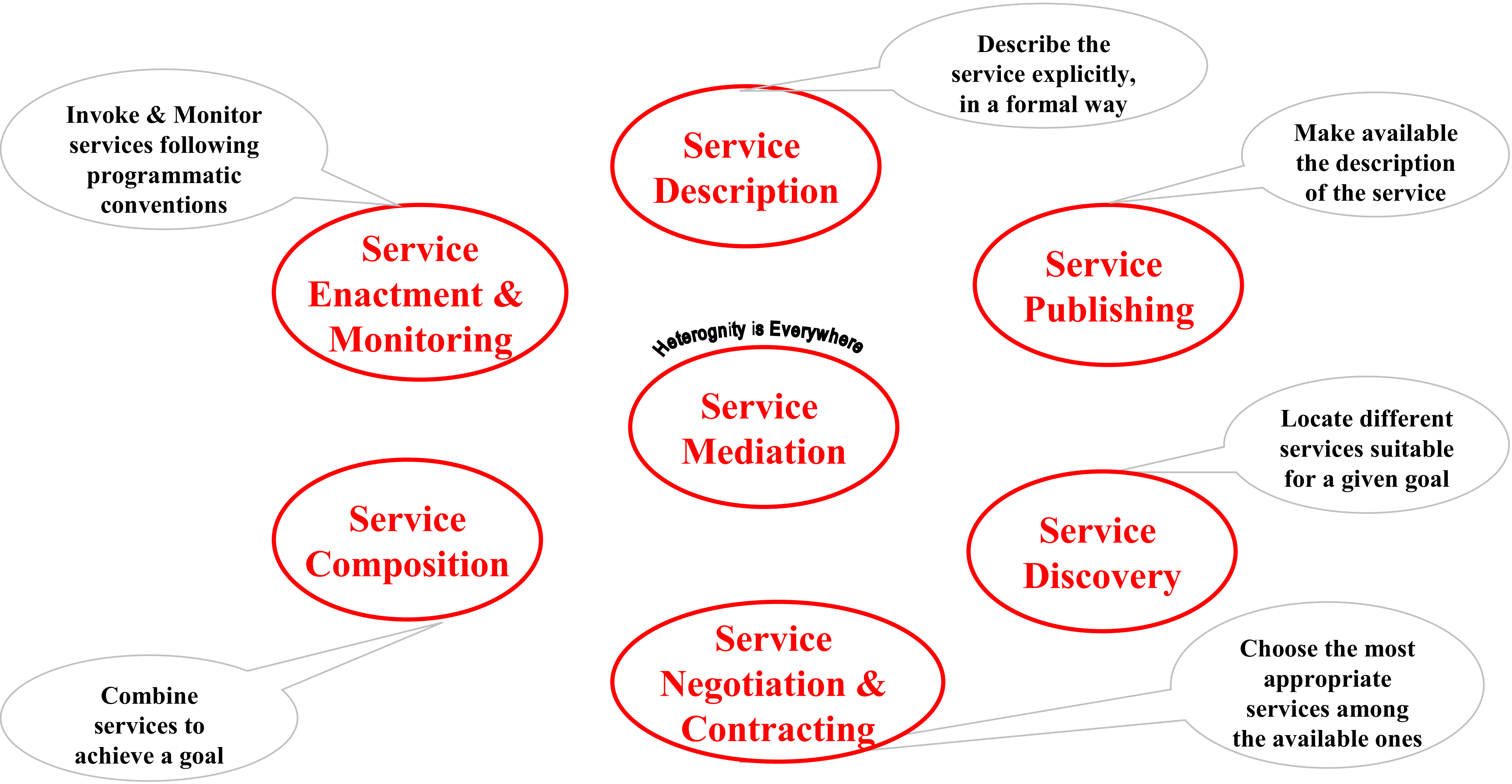
Semantic Web Services

Static

WWW
URI, HTML, HTTP

Semantic Web
RDF, RDF(S), OWL, etc.

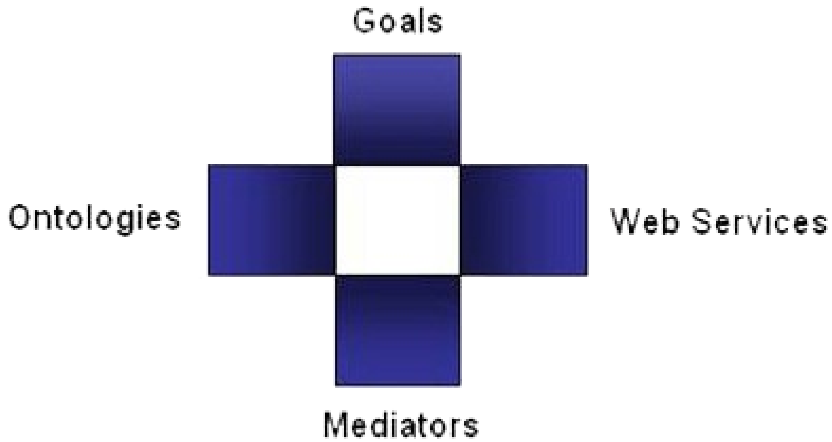
SWS – Tasks to be Automated



The WSMO Approach to SWS

Objectives that a client may
have when consulting a Web Service

Provide the formally specified terminology of the information used by all other components



Semantic description of Web Services:
- **Capability** (*functional*)
- **Interfaces** (*usage*)

Connectors between components with mediation facilities for handling heterogeneities

What is an ontology?

- Formal,
- explicit specification of
- a shared conceptualization of a domain.

- Meaning of ontology is unambiguous
- Avoids misunderstanding
- Specification using formal language
- Enables reasoning: making implicit information explicit
- Hampers consensus

- Make domain assumptions explicit
  - For reasoning
  - For clarifying understanding of domain
- Minimal ontological commitment
  - Too much explicit => no consensus
  - Too little explicit => ontology unusable
  - Minimal ontological commitment = "make as little as explicit as possible, while keeping ontology useful"

- Domain: specific part of the world
- Conceptualization
  - Forming idea of domain in the minds of people
- Shared among its users
  - Facilitates accepting the ontology

Elements of Ontologies

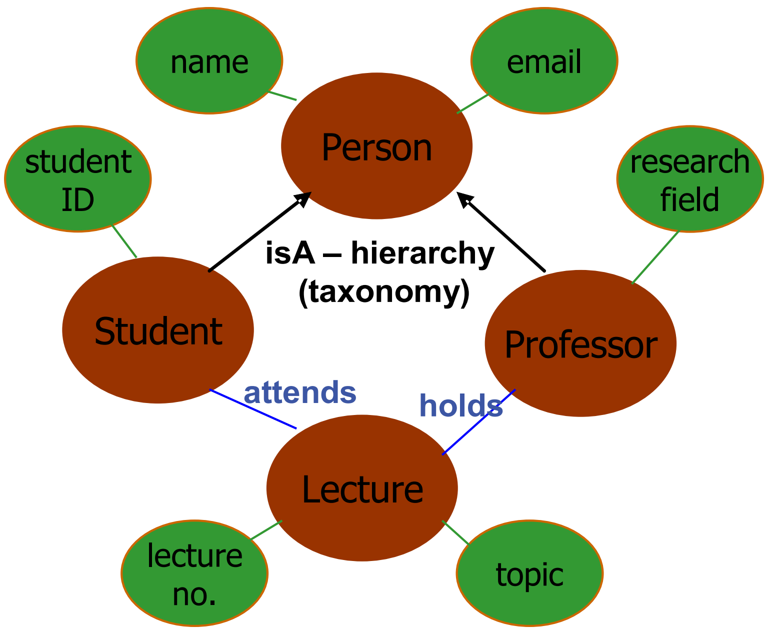
Concept
conceptual entity of the domain

Property
attribute describing a concept

Relation
relationship between concepts or properties

Axiom
coherency description between Concepts / Properties / Relations via logical expressions

Instance
individual in the domain

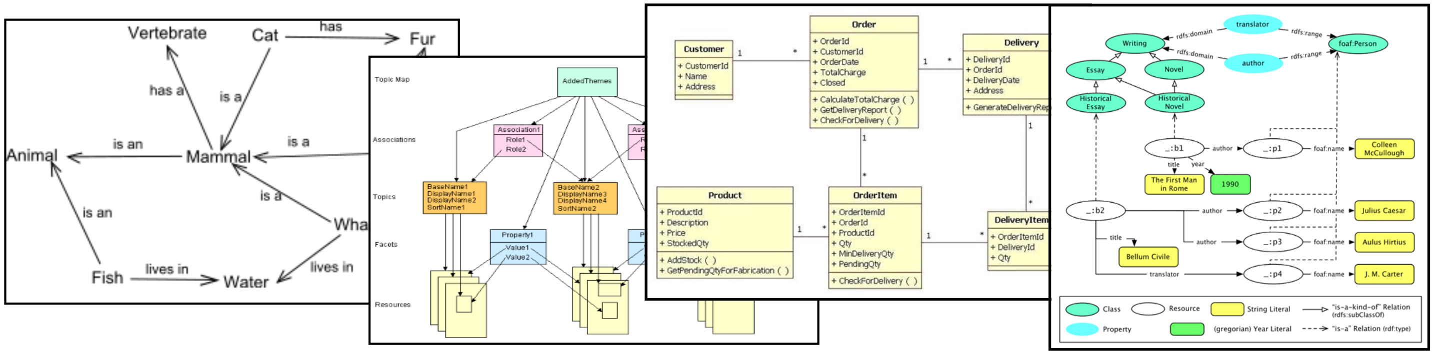


holds(Professor, Lecture) =>
Lecture.topic = Professor.researchField

Ann memberOf student
name = Ann Lee
studentID = 12345

Wide Variety of Languages for Specifying Ontologies

- **Graphical**: Semantic Networks, Topic Maps, UML, RDF



- **Logical**: Description Logics, First Order Logic, Rules, Conceptual Graphs

DL Syntax	Example
$C_1 \sqcap \ldots \sqcap C_n$	Human ⊓ Male
$C_1 \sqcup \ldots \sqcup C_n$	Doctor ⊔ Lawyer
$\neg C$	¬Male
$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} ⊔ {mary}
$\forall P.C$	∀hasChild.Doctor
$\exists P.C$	∃hasChild.Lawyer
$\leqslant nP$	⩽1hasChild
$\geqslant nP$	⩾2hasChild

Brothers are siblings

$\forall x, y \;\; Brother(x,y) \;\Rightarrow\; Sibling(x,y).$

"Sibling" is symmetric

$\forall x, y \;\; Sibling(x,y) \;\Leftrightarrow\; Sibling(y,x).$

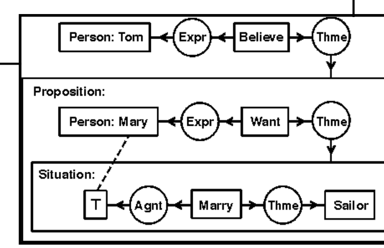
One's mother is one's female parent

$\forall x, y \;\; Mother(x,y) \;\Leftrightarrow\; (Female(x) \land Parent(x,y)).$

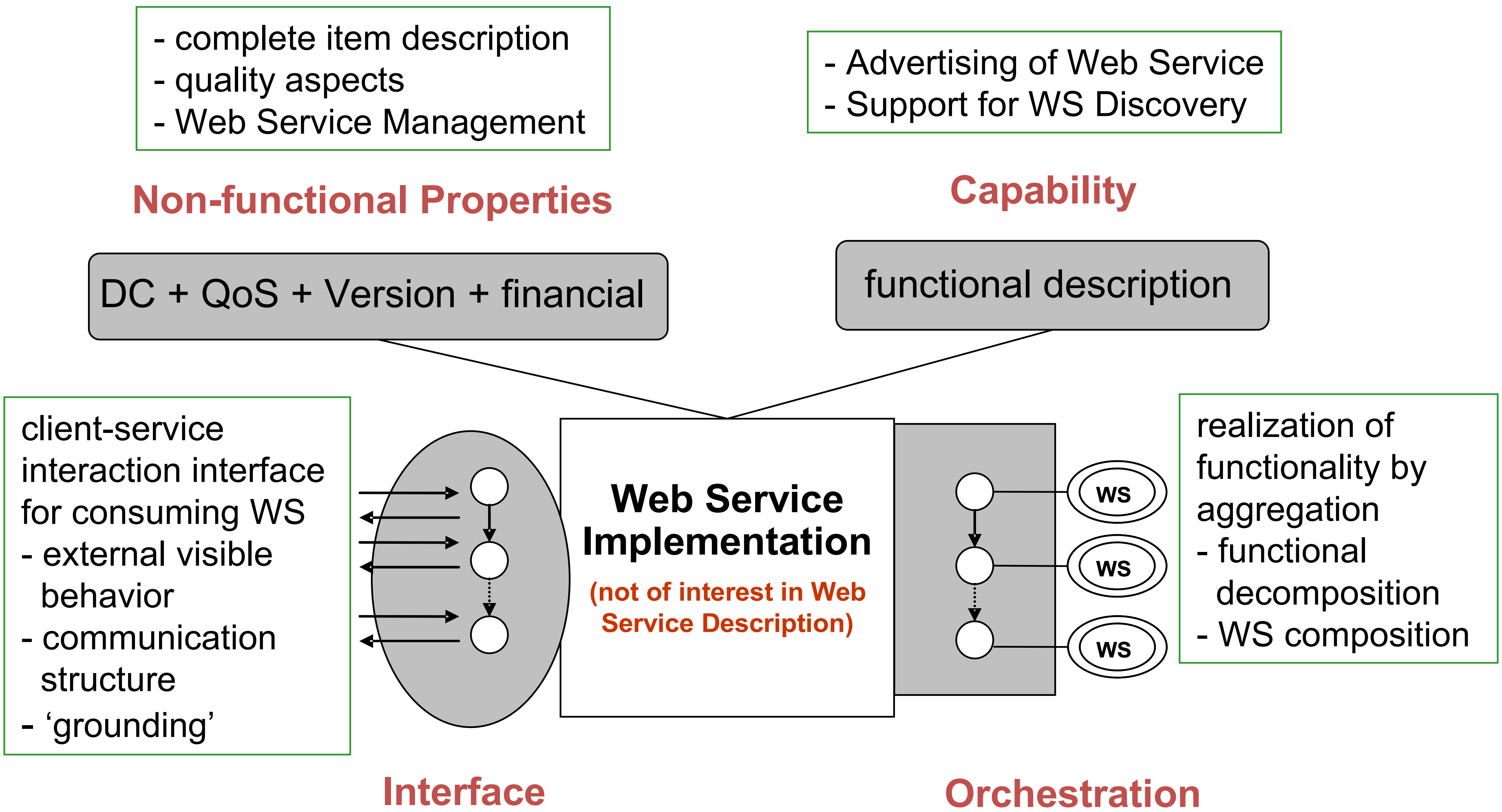
A first cousin is a child of a parent's sibling

$\forall x, y \;\; FirstCousin(x,y) \;\Leftrightarrow\; \exists p, ps \;\; Parent(p,x) \land Sibling(ps,p) \land Parent(ps,y)$

```
sibling(X, Y)      :- parent_child(Z, X), parent_child(Z, Y).
parent_child(X, Y) :- father_child(X, Y).
parent_child(X, Y) :- mother_child(X, Y).
mother_child(trude, sally).
father_child(tom, sally).
father_child(tom, erica).
father_child(mike, tom).
```



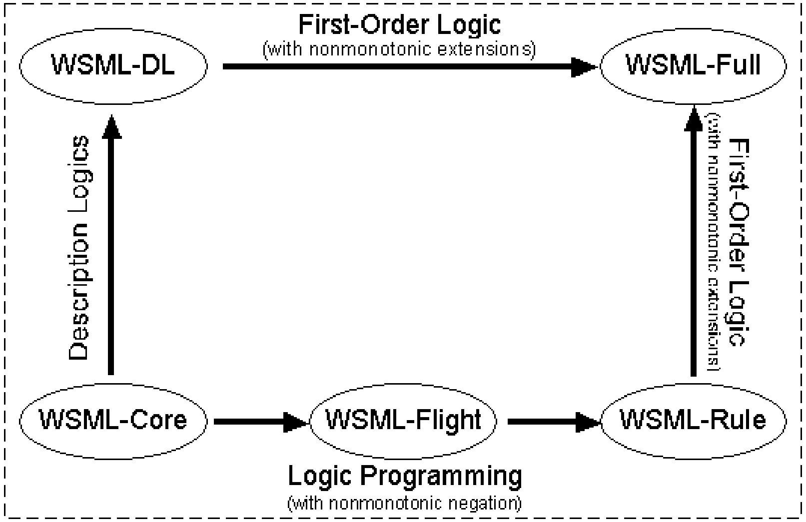
A Conceptual Model for Web Services



Web Service Modeling Language (WSML)

- Aim – to provide a language (or a set of interoperable languages) for representing the elements of WSMO: Ontologies, Web services, Goals, Mediators

- For ontologies, WSML provides a formal language based on:
  - *Description Logics*
  - *Logic Programming*
  - *First-Order Logic*
  - *F-Logic*



- WSML is a family of languages layered on top of XML and RDF

WSML in SWING

- Ontologies
  - GeographicDatatypes, GeospatialOperations, QuarriesOntology, MeasurementOntology, WFS, Annotation, etc.
- Web Services
  - Define functionalities of WFS and WPS Web Services
- Goals
  - Define WFS and WPS Goals
- Annotations
  - Encode annotations coming from the annotation tool

WSML in SWING - Example

GeographicDatatypes.wsml

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { _"http://swing-project.org#GeographicDatatypes"
 }

ontology GeographicDatatypes

concept GM_Object
hasSRS impliesType SRS

concept Polygon subConceptOf GM_Object

concept SRS

concept projSRS subConceptOf SRS

instance gk memberOf projSRS

axiom gm_objectDefinition definedBy
?x[hasSRS hasValue ?srs] implies ?x memberOf GM_Object.
```

UnionWPS.wsml

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { _"http://swing-project.org#UnionWPS",

        GeoTypes _"http://swing-project.org#GeographicDatatypes",
        GeoOp    _"http://swing-project.org#GeospatialOperations"
 }

webService UnionWPS

importsOntology {GeoOp#GeospatialOperations, GeoTypes#GeographicDatatypes}

capability UnionWPSCapability

sharedVariables { ?a, ?b, ?refsys }

precondition UnionWPSPrecondition definedBy
    ?a[GeoTypes#hasSRS hasValue ?refsys] memberOf GeoTypes#GM_Object and
    ?b[GeoTypes#hasSRS hasValue ?refsys] memberOf GeoTypes#GM_Object and
    ?refsys memberOf GeoTypes#projSRS.

postcondition UnionWPSPostcondition definedBy
    ?c memberOf GeoTypes#Polygon and
    ?c[GeoTypes#hasSRS hasValue ?refsys] and
    GeoOp#union(?a, ?b, ?c).
```

Goal2.wsml

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { _"http://swing-project.org#Goal2"
,
     GeoTypes _"http://swing-project.org#GeographicDatatypes",
     GeoOp _"http://swing-project.org#GeospatialOperations" }

goal Goal2

importsOntology {GeoOp#GeospatialOperations, GeoTypes#GeographicDatatypes}

capability Goal2Capability

sharedVariables { ?x, ?y }

precondition Goal2Precondition definedBy
    ?x[GeoTypes#hasSRS hasValue ?srs] memberOf GeoTypes#Polygon and
    ?y[GeoTypes#hasSRS hasValue ?srs] memberOf GeoTypes#Polygon and
    ?srs memberOf GeoTypes#projSRS.

postcondition Goal2Postcondition definedBy
    ?z memberOf ?outputType and
    GeoOp#overlay(?x, ?y, ?z).
```

WSML in SWING - Example

GEOWS.2009

GeographicDatatypes.wsml

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { _"http://swing-project.org#GeographicDatatypes"
 }

ontology GeographicDatatypes

concept GM_Object
hasSRS impliesType SRS

concept Polygon subConceptOf GM_Object

concept SRS

concept projSRS subConceptOf SRS

instance gk memberOf projSRS

axiom gm_objectDefinition definedBy
?x[hasSRS hasValue ?srs] implies ?x memberOf GM_Object.
```

WSML in SWING - Example

UnionWPS.wsml

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { _"http://swing-project.org#UnionWPS",

        GeoTypes _"http://swing-project.org#GeographicDatatypes",
        GeoOp    _"http://swing-project.org#GeospatialOperations"
 }

webService UnionWPS

importsOntology {GeoOp#GeospatialOperations, GeoTypes#GeographicDatatypes}

capability UnionWPSCapability

sharedVariables { ?a, ?b, ?refsys }

precondition UnionWPSPrecondition definedBy
    ?a[GeoTypes#hasSRS hasValue ?refsys] memberOf GeoTypes#GM_Object and
    ?b[GeoTypes#hasSRS hasValue ?refsys] memberOf GeoTypes#GM_Object and
    ?refsys memberOf GeoTypes#projSRS.

postcondition UnionWPSPostcondition definedBy
    ?c memberOf GeoTypes#Polygon and
    ?c[GeoTypes#hasSRS hasValue ?refsys] and
    GeoOp#union(?a, ?b, ?c).
```

WSML in SWING - Example

Goal2.wsml

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { _"http://swing-project.org#Goal2"
,
      GeoTypes _"http://swing-project.org#GeographicDatatypes",
      GeoOp _"http://swing-project.org#GeospatialOperations" }


goal Goal2

importsOntology {GeoOp#GeospatialOperations, GeoTypes#GeographicDatatypes}

capability Goal2Capability

sharedVariables { ?x, ?y }

precondition Goal2Precondition definedBy
     ?x[GeoTypes#hasSRS hasValue ?srs] memberOf GeoTypes#Polygon and
     ?y[GeoTypes#hasSRS hasValue ?srs] memberOf GeoTypes#Polygon and
     ?srs memberOf GeoTypes#projSRS.

postcondition Goal2Postcondition definedBy
     ?z memberOf ?outputType and
     GeoOp#overlay(?x, ?y, ?z).
```

Web Service Discovery

- Functionality
  - Identify possible web services W which are able to provide the requested service S for its clients
- An important issue …
  - „being able to provide a service“ has to be determined based on given descriptions only (WS, Goal, Ontos)
  - Discovery can *only be as good* as these descriptions
    - *Very detailed WS descriptions*: are precise, enable highly accurate results, are more difficult to provide; in general, requires interaction with the provider (outside the pure logics framework)
    - *Less detailed WS descriptions*: are easy to provide for humans, but usually less precise and provide less accurate results

Ease of provision

Possible Accuracy

WSML Discovery

Level of Abstraction

- Support a wide-variety of applications wrt. needed accuracy
- Basic possibilities for the description of web services:
  - Syntactic approaches
    - Keyword-based search, natural language processing techniques, Controlled vocabularies

→ WS as a set of keywords

  - Lightweight semantic approaches
    - Ontologies, What does W provide (not how)?, Coarse-grained semantic description of a service

→ WS as a set of objects

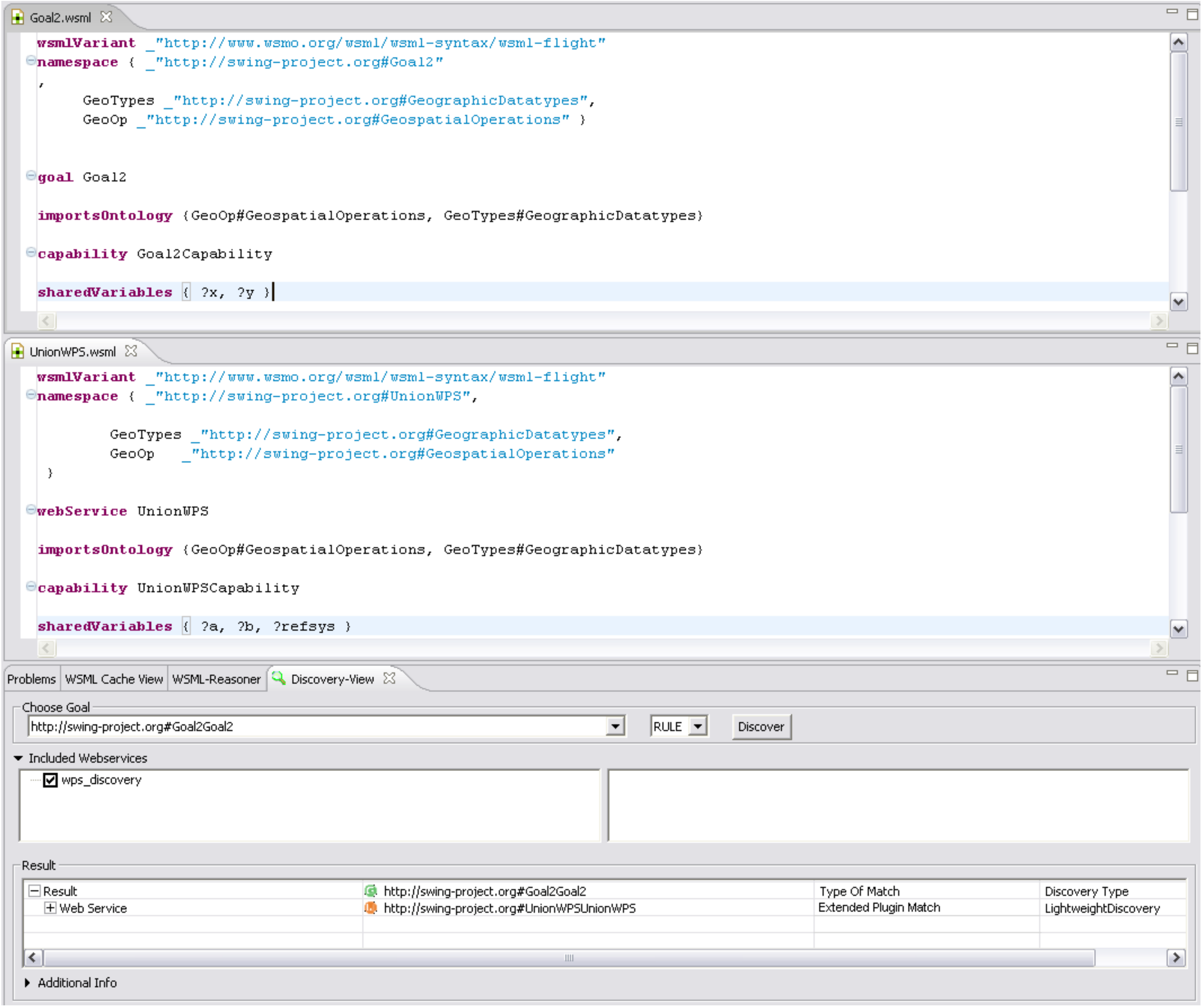
  - Heavyweight semantic approaches
    - Describes the service capability in detail, Pre/Post-Cond, takes „in-out" relationship into account, Fine-grained web service description

→ WS as a set of state-changes

WSML Discovery

- Responsible to find appropriate Web Services to achieve a goal
- Current discovery component is organized as a framework performing discovery in two steps:
  1. (optional) keyword-based matching
  2. discovery based on either simple or rich descriptions of services
     - Simple Descriptions → “lightweight” discovery
       - take into account postconditions and effects
       - WSML-DL: use concept subsumption; possible matches: exact, plugin, subsume, intersection
       - WSML-Flight / WSML-Rule: use query containment; possible matches: exact, plugin, subsume
     - Rich Descriptions → “heavyweight” discovery
       - take into account preconditions and assumptions, postconditions and effects, and the relation inbetween
       - WSML-Flight / WSML-Rule: use query containment; possible match: extended plug-in match

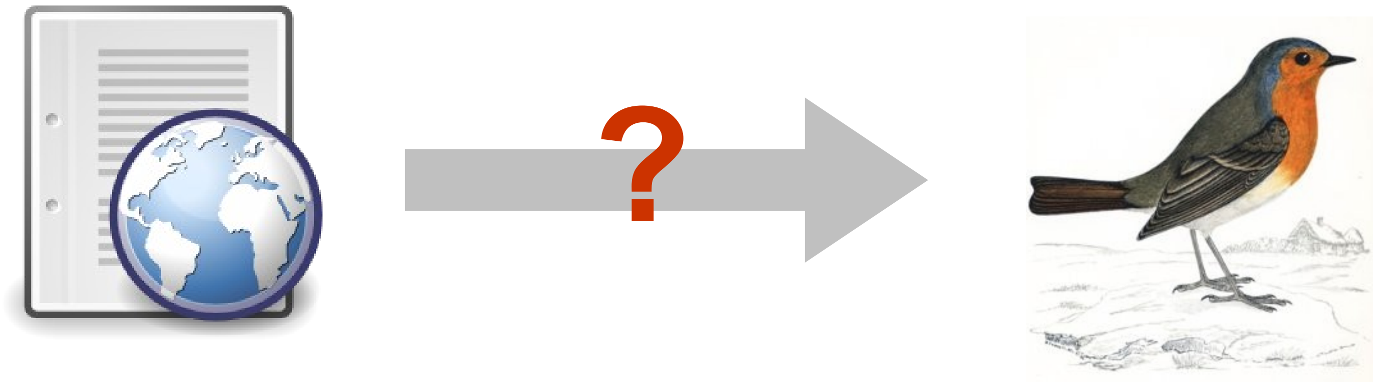
Discovery in SWING - Example



Overview

- Spatial Data Infrastructures
- Semantic Web Services
- **Bridging the gap: Semantic Annotations**
- SWING: use cases

- SWING: Developed Tools
- Demonstration (Videos)
- Hands-On Session



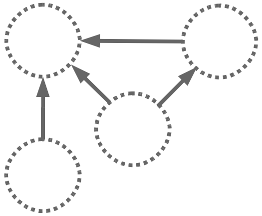
Semantic Annotations

GEOWS 2009
February 1st, 2009 - Cancun, Mexico

Looking Back

- Web Services in Spatial Data Infrastructures
  - Lack sophisticated thematic descriptions

- Semantic Web & Ontologies
  - Provide means to capture data semantics



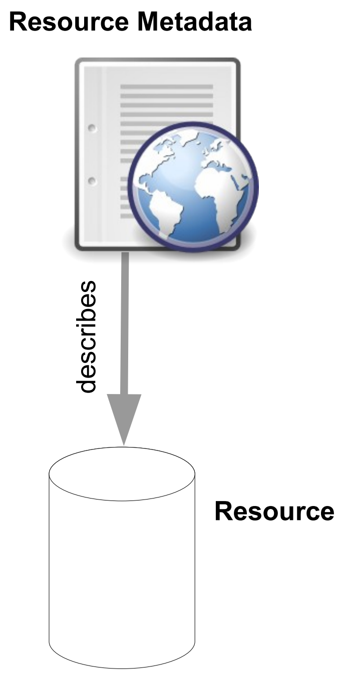
- Semantic annotations as approach to link the two worlds

Overview

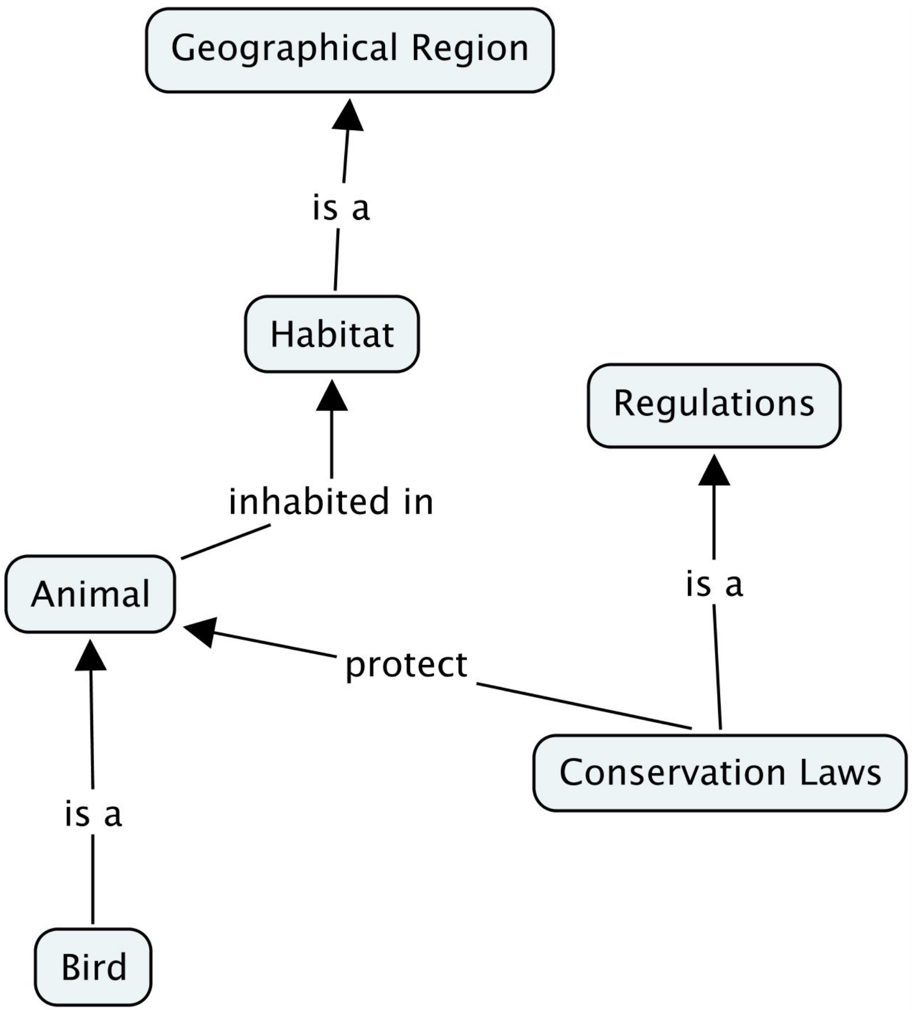
- Explaining semantic annotations
- Ways to establish the connection
- Making use of the link
  - Creating the semantic annotations
  - Querying based on semantic annotations
- Semantic Annotations in OGC Standards

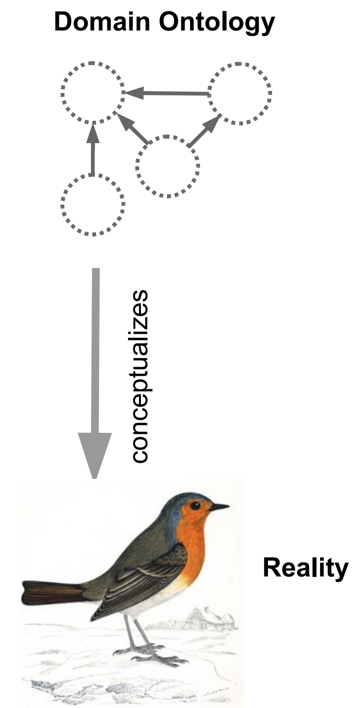
ZICO


```
<element name="Zico_region"
         type="con:Zico_regionType"
         substitutionGroup="gml:_Feature" />
<complexType name="Zico_regionType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="msGeometry" type="gml:GeometryPropertyType"/>
        <element name="REGIONAL" type="string"/>
        <element name="NATIONAL" type="string"/>
        <element name="LIBELLE" type="string"/>
        <element name="TYPE" type="string"/>
        <element name="LA_MESURE" type="string"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```



Example of corresponding Ontology

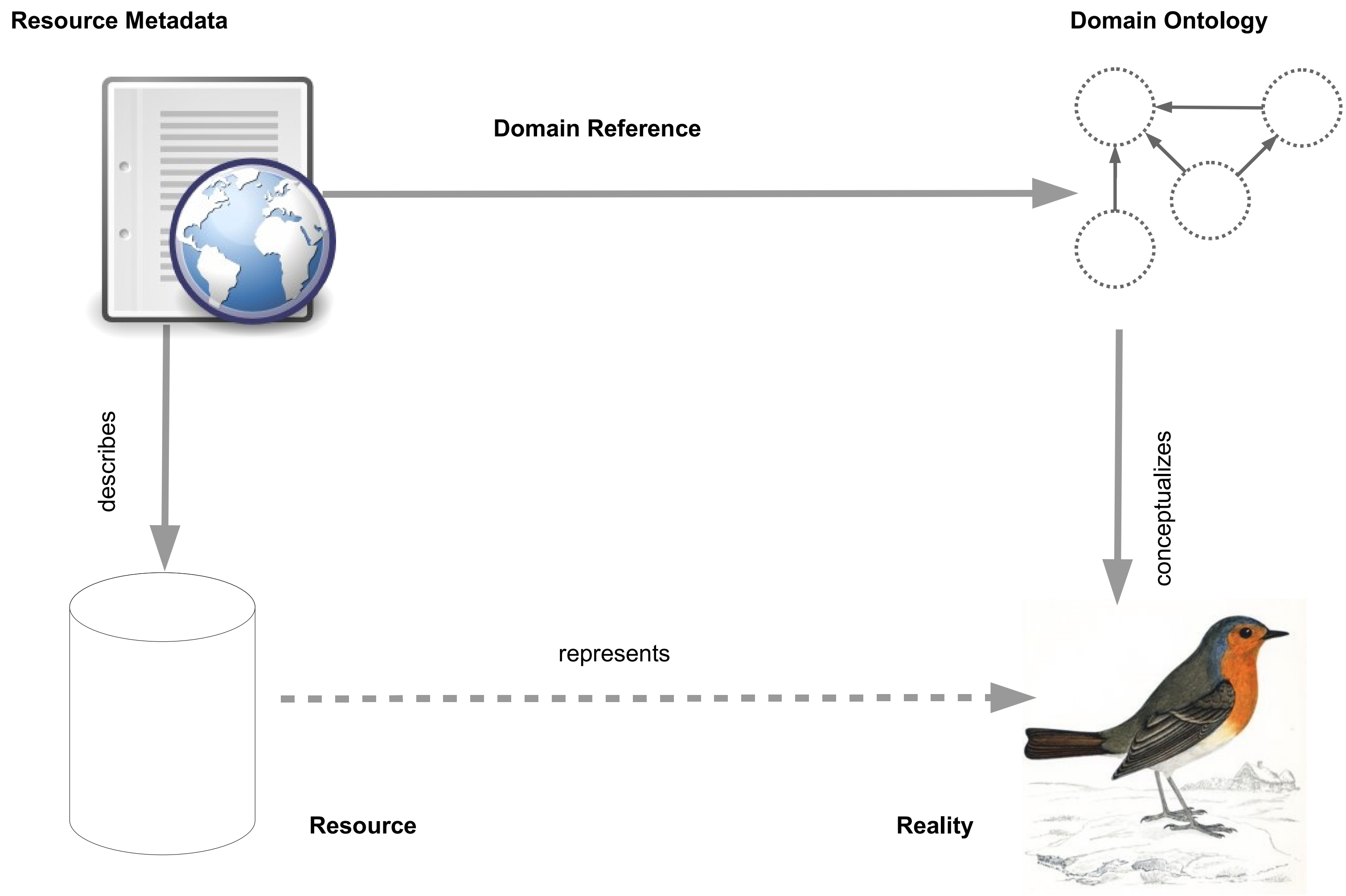




Why the difference?

- Local vs. Global
  - Describing the local and linking to global
  - Searching the global and finding the local
- Data Models vs. Real world
  - Domain Ontologies capture real world semantics
  - Data Models represent application specific knowledge
  - Semantic Annotations keep it separated

Simple solution: concept to concept



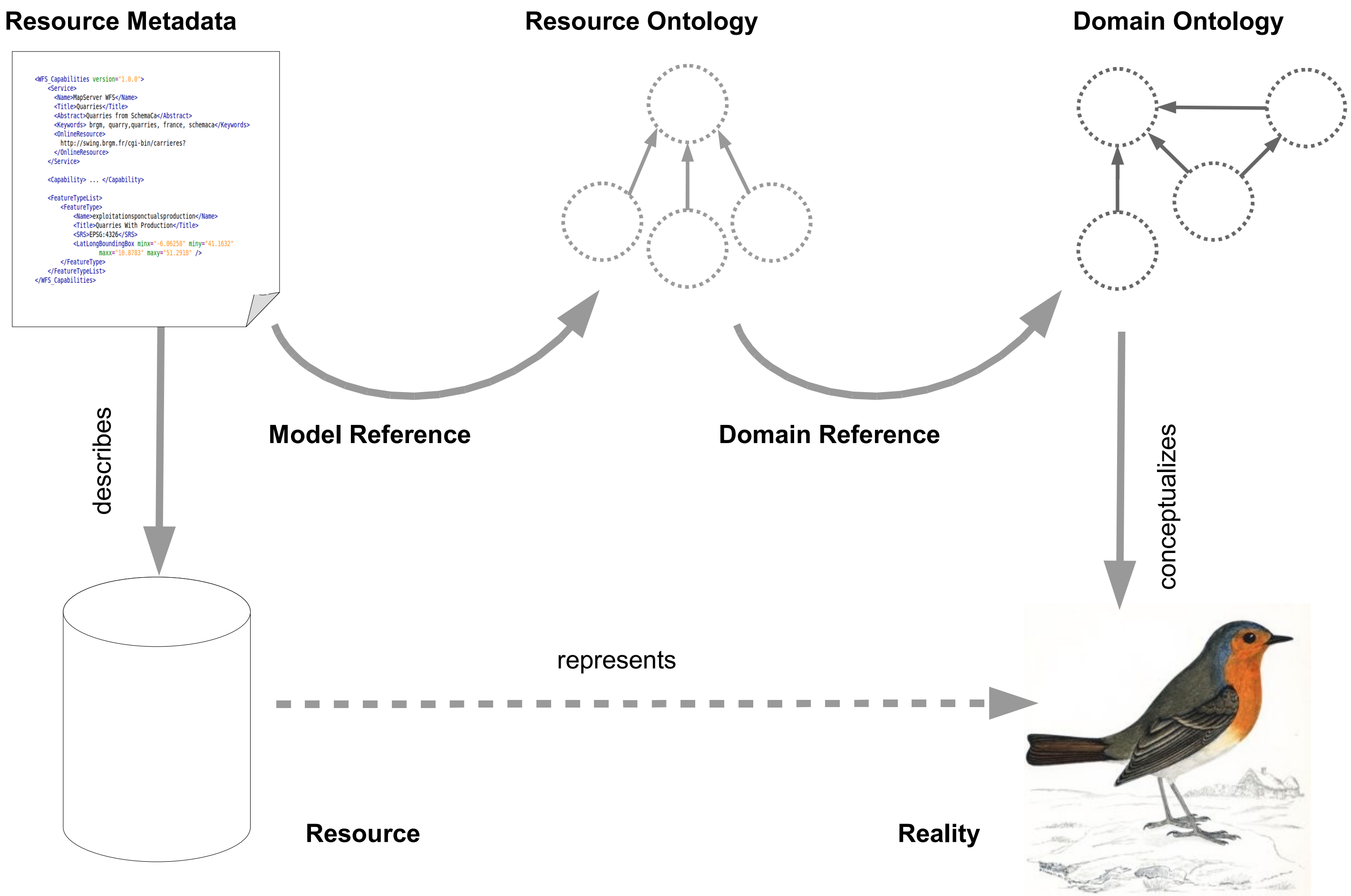
Doesn't work


```
<complexType name="Zico_regionType">
    <element name="REGIONAL" type="string" reference="dom:Identifier"/>
    <element name="LIBELLE" type="string" reference="dom:Identifier"/>
</complexType>
```

```
<complexType name="Zico_regionType">
    <element name="REGIONAL" type="string"
            reference="dom:GeographicalRegionName"/>
    <element name="LIBELLE" type="string"
            reference=="dom:ProtectedBirdRegionName"/>
</complexType>
```

- Too generic → loose benefits
- Too specific → cluttered domain vocabulary

Rule-based annotations



How to create rule based annotations


```
axiom defineZICO
 definedBy
  ?feature[LIBELLE ofType ?attrLibelle] memberOf ZicoFT and
  ?domBirdArea memberOf dom#ImportantBirdArea and
  ?domIdentifier memberOf dom#Identifier and
  dom#domainReference(?feature, ?domBirdArea) and
  dom#domainReference(?attrlibelle, ?domIdentifier) and
  dom#names(?attrlibelle, ?feature).
```

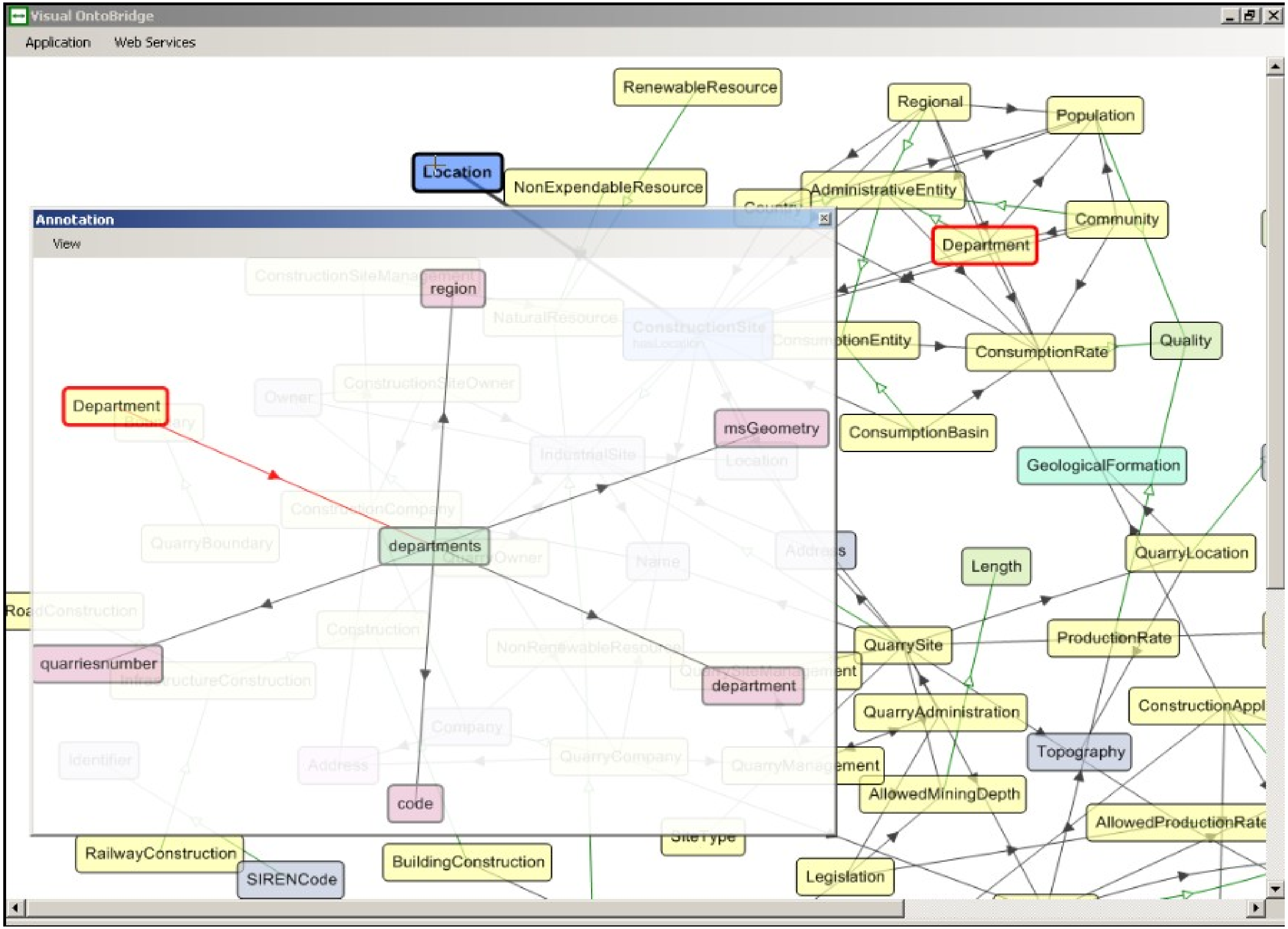
How to query rule based annotations

Goal

Post Condition

?domBirdArea memberOf dom#ImportantBirdArea and
?domIdentifier memberOf dom#Identifier and
dom#domainReference(?feature, ?domBirdArea) and
dom#domainReference(?attrlibelle, ?domIdentifier) and
dom#names(?attrlibelle, ?feature).

Visually-supported Annotations/Queries



Open Issues: Standardization

- Model References already standard (W3C SAWSDL)
- Support in OGC Standards required
  - Storing semantic annotation
  - Querying semantic annotations
  - End-user tools support
- Discussion Paper with different approaches prepared

Open Issues: Processes

- Model References already standard (W3C SAWSDL)
- Support in OGC Standards required
  - Storing semantic annotation
  - Querying semantic annotations
  - End-user tools support
- Discussion Paper with different approaches prepared
- How can we annotate Geoprocesses
  - Domain vocabulary of Geo-operations required? All?
  - Or just describing relation between input and output?

Overview

- Spatial Data Infrastructures
- Semantic Web Services
- Bridging the gap: Semantic Annotations
- **SWING: use cases**

- SWING: Developed Tools
- Demonstration (Videos)
- Hands-On Session

Semantic **W**eb Services **In**teroperability in **G**eospatial decision making

Use Cases

GEOWS 2009
February 1st, 2009 - Cancun, Mexico

SWING application theme

- BRGM : Mineral resources Management
- Aggregate production and consumption
  - **EU aggregate production & consumption is the largest macro-regional market in the world**
  - Aggregates - mineral resources with average low value
    - sand, gravel, crushed stone, ….
    - produced on-shore (quarries), off-shore, and some recycling (concrete)
  - EU production 3 billion tons
  - EU employment 250,000 jobs
  - EU value 35 billion €
  - consumption 5-15 tonnes per capita per year
  - about 25,000 production sites in Europe

SWING application theme

- Quarries and Aggregates

Aggregates = crushed hard rock (limestone, volcanic rock, sandstone, recycled concrete, ….), or on- & off-shore sediments (sand & gravel)


Vosges


Puy-de-Dome


Loire-Atlantique


Sarthe


Loiret


la Manche


Amsterdam


Essonne

SWING

- Decision Making Support :
  - A new way of doing things in the Inspire Context
    - Reduce time to deliver
    - Brings interactivity

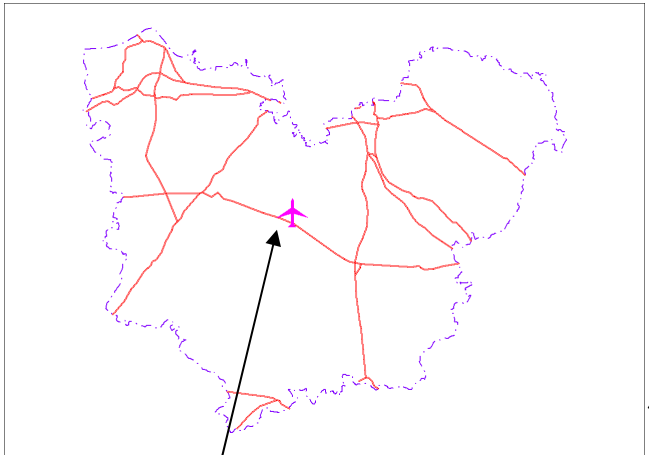
- Objectives in SWING
  - **develop a geospatial decision-making application that can dynamically find and integrate interoperable semantic web services.** (…. *with the potential of being further developed and turned into a management and assessment system for natural resources)*
  - **evaluate the appropriateness of the technical framework**

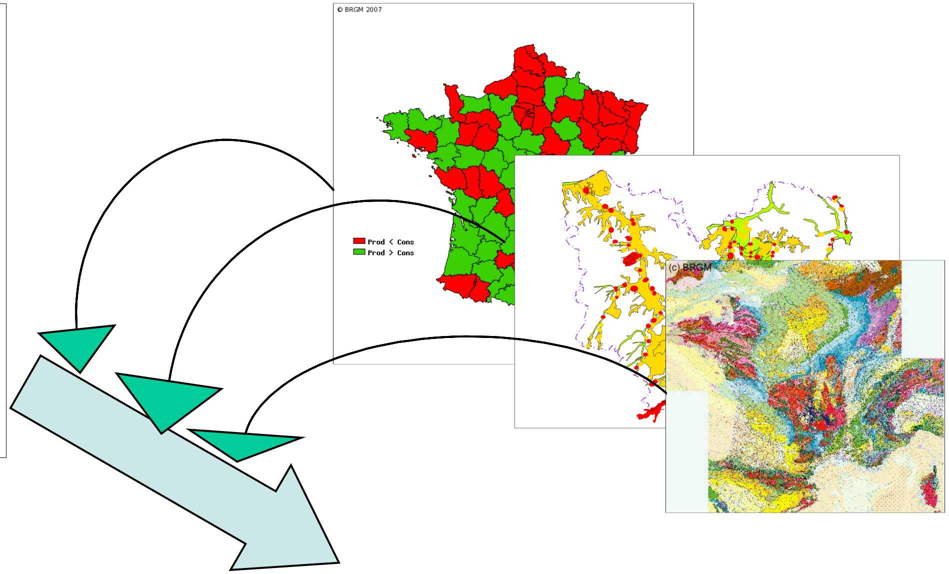
Use Cases overall objective

- Given
  - Inputs for a new infrastructure project
  - Production/Consumption of actual quarries
  - Known Land-uses constraints
  - Geology

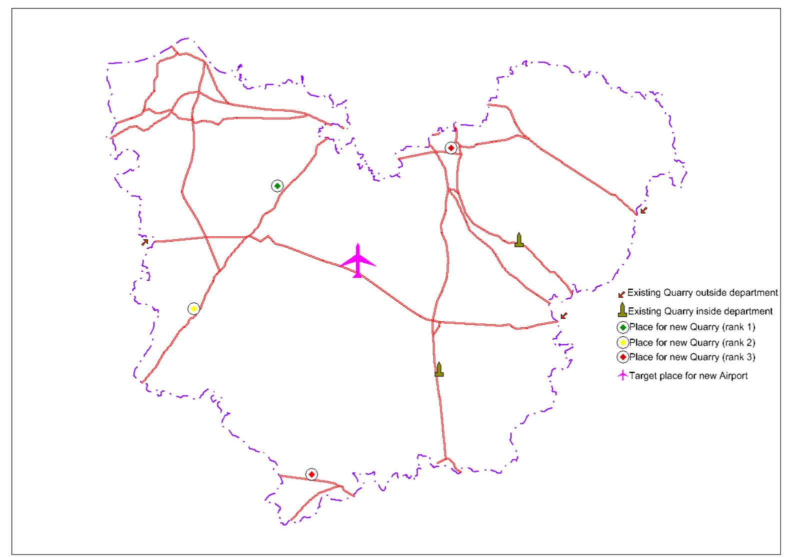
- Find places where to get aggregates
  - From existing quarries
  - By opening new ones (Land-use constraints + Geology)

Use Cases





Substances	Quantity(*)
Sand	**110 000 tons**
Granite	**50 000 tons**
Concrete	**200 000 tons**



(*) All Quantity and Substances are given for example and are NOT real

The differents steps

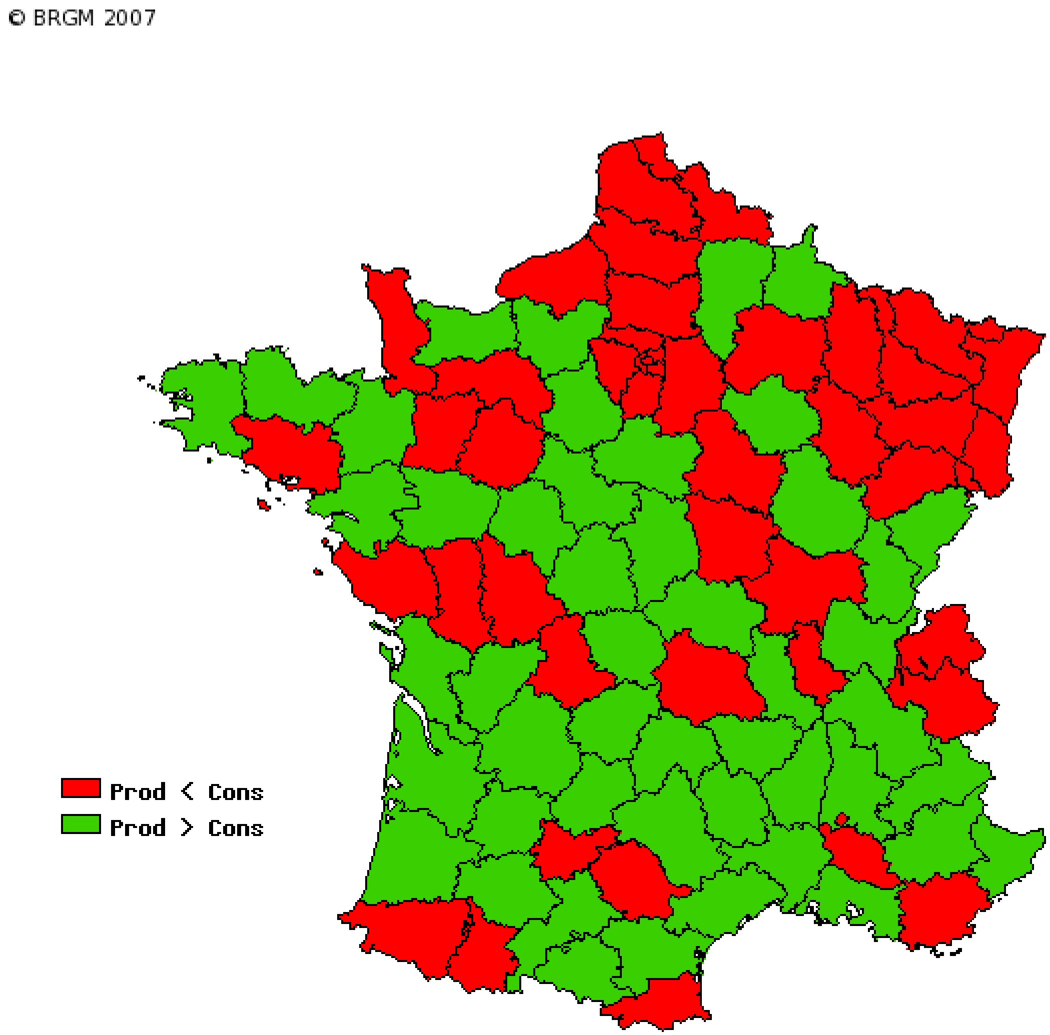
- Use Case 1 : Production/Consomption Map
- Use Case 2 : Land-Use constraints integration
- Use Case 3 : Find the best place

Use Case 1 - Create a simple map

- Thematic Objective: Create a consumption-production map of aggregates

- Technical challenges:
  - Set up needed DATA and Web Services (OGC and WSDL)
  - Build a WSML Domain Ontology
  - Annotate available WS with the Domain Ontology
  - Register WS in CAT and Store WS annotations
  - Setup simple WS composition, annotate and store into CAT, execute it with WSMX.

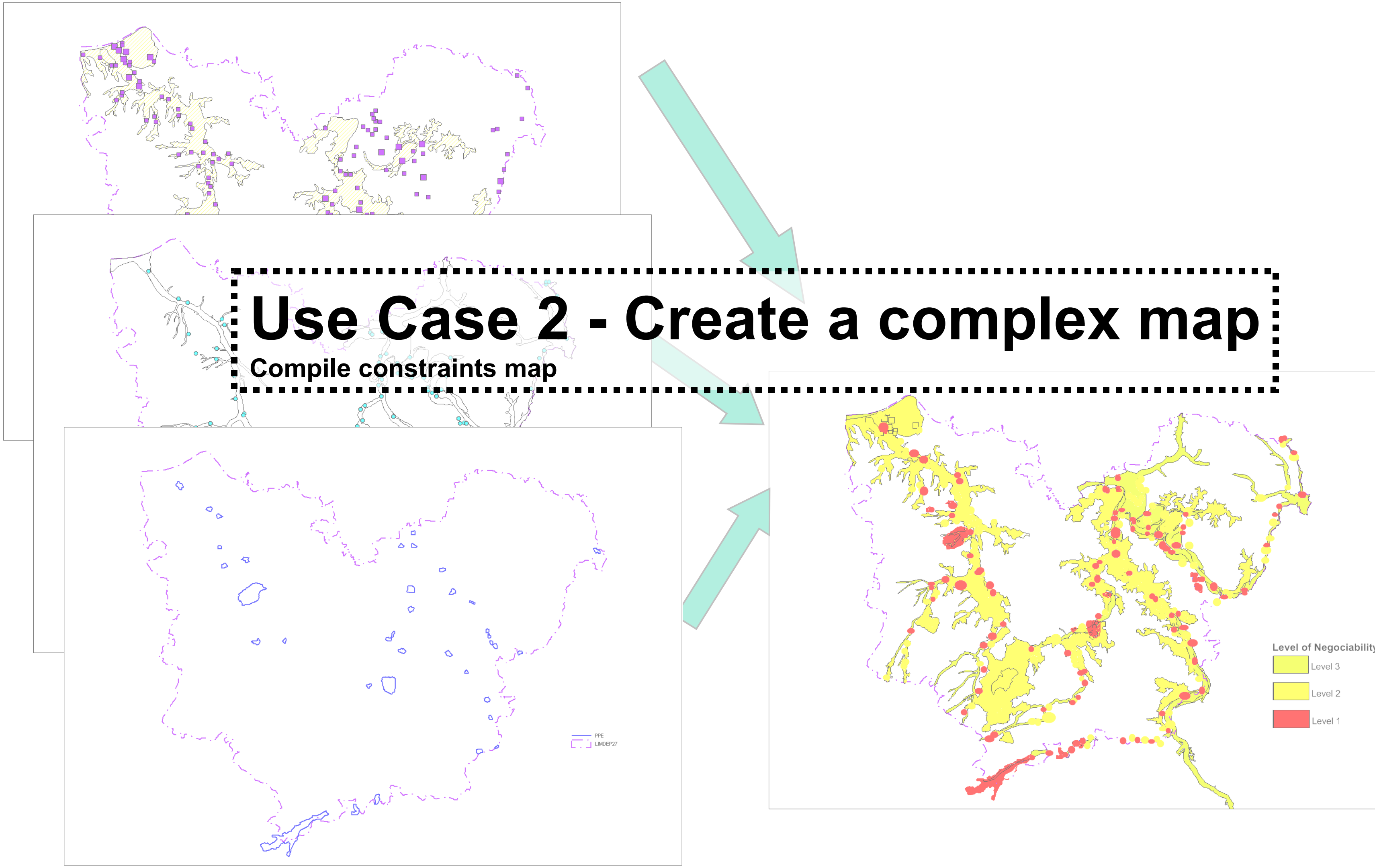
Use Case 1 - Create a simple map



Fake Consumption/Production Map (based on population of departments)

Use Case 2 - Create a complex map

Compile constraints map

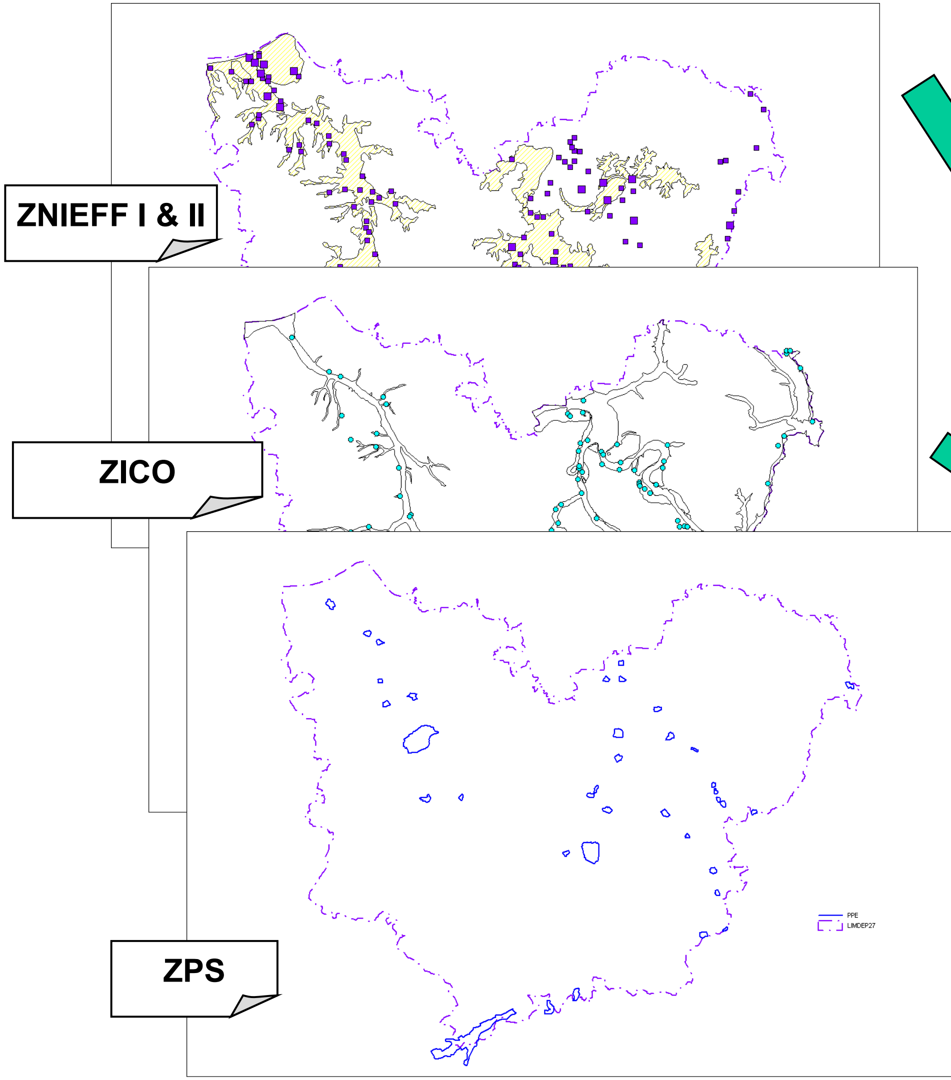


Use Case 2 - Create a complex map

- Thematic Objective: Create a map of land-use constraints and publish it as a decision making support document.
- Data Sources for Use Case 2
- Technical challenges
  - Implement WPS to combine multiple constraints
  - Extend the Domain Ontology to take land-use constraints into account; Improve the Ontology engineering process
  - Improve the annotation process (towards semi-automatic annotation)
  - Improve technical architecture of the end-user interface

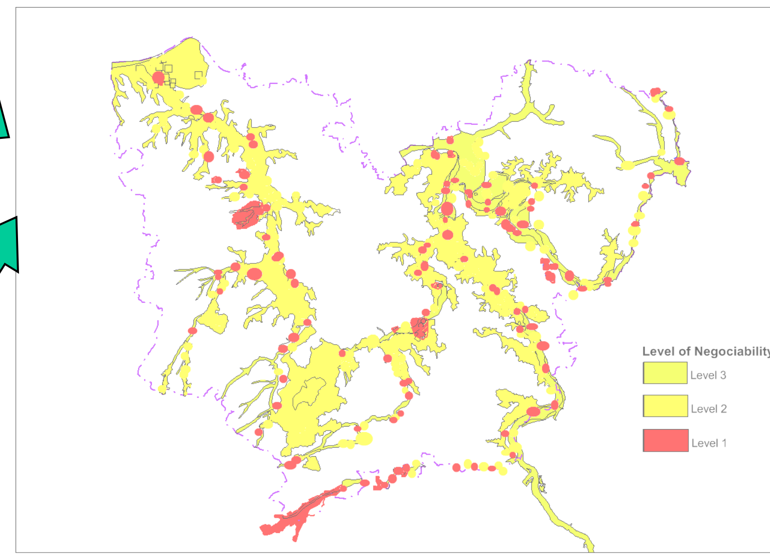
Use Case 2 - Create a complex map

GEOWS.2009

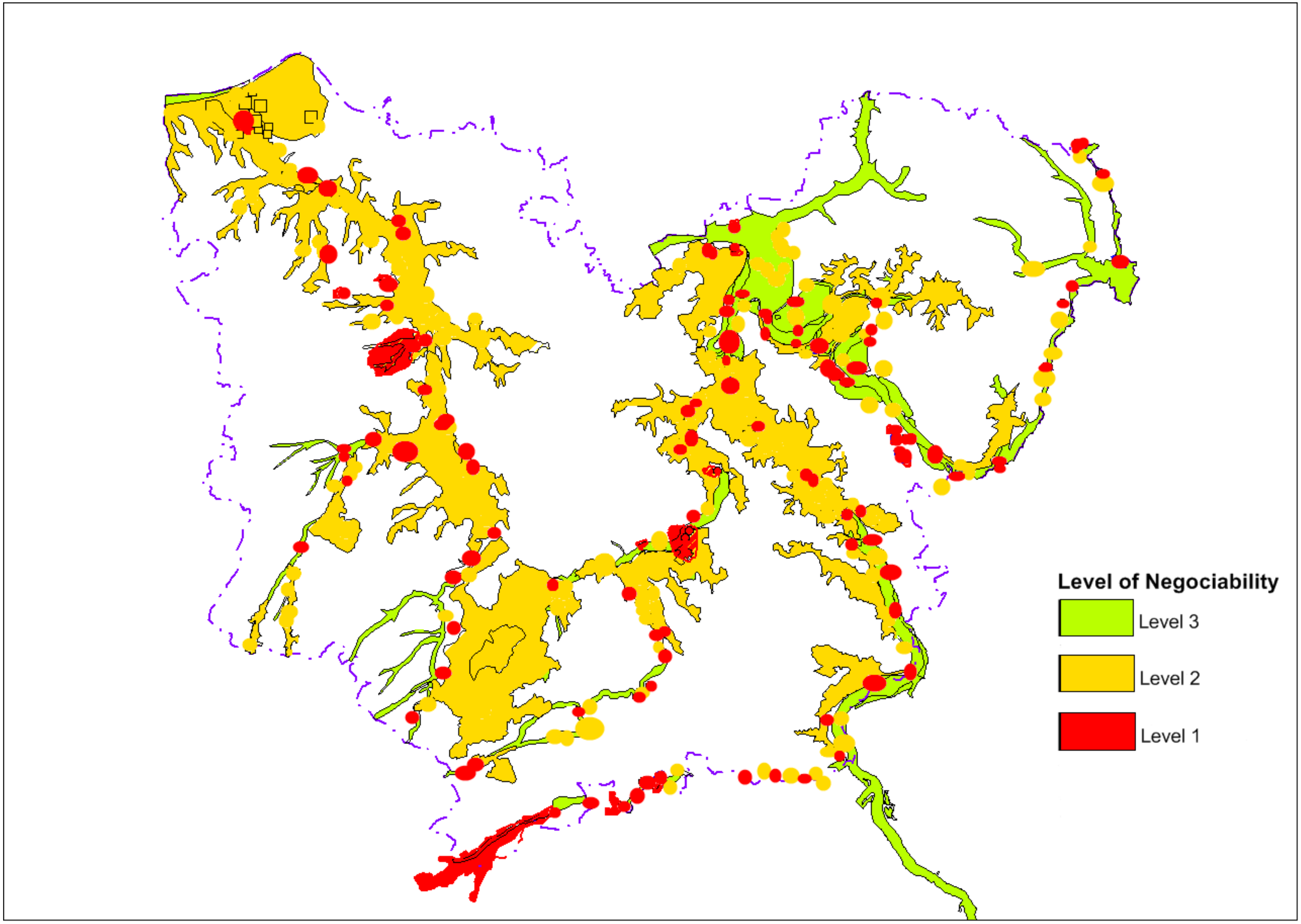


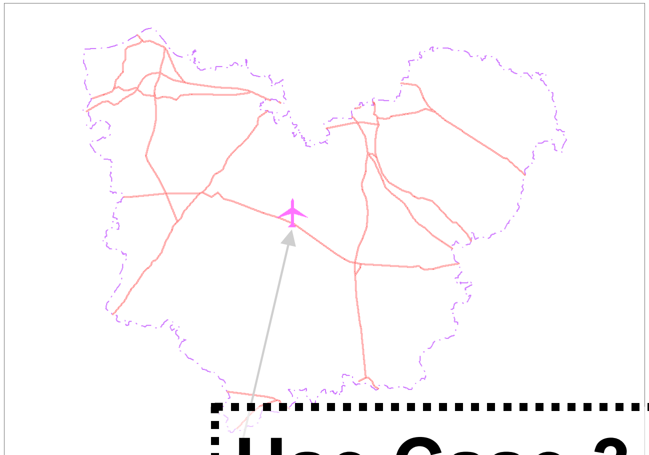
Land-Use Constraints

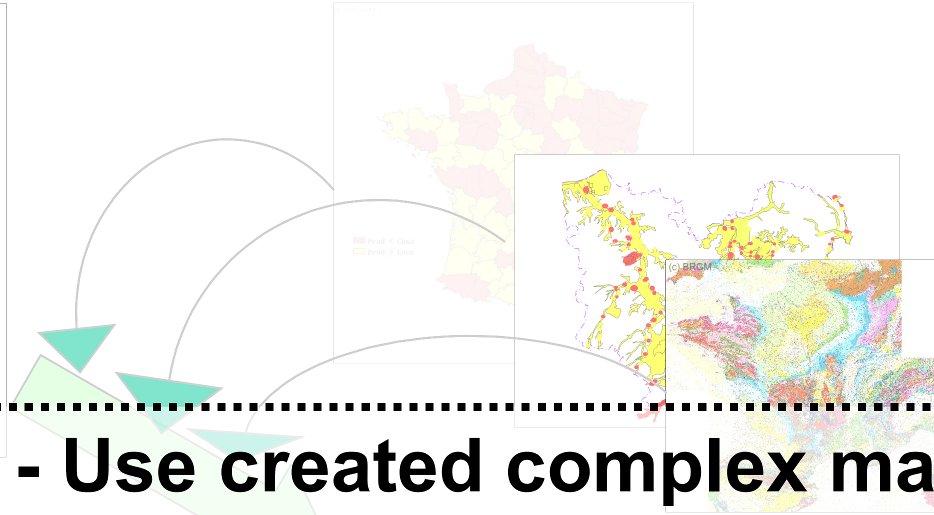
Negotiability level



Use Case 2 - Create a complex map



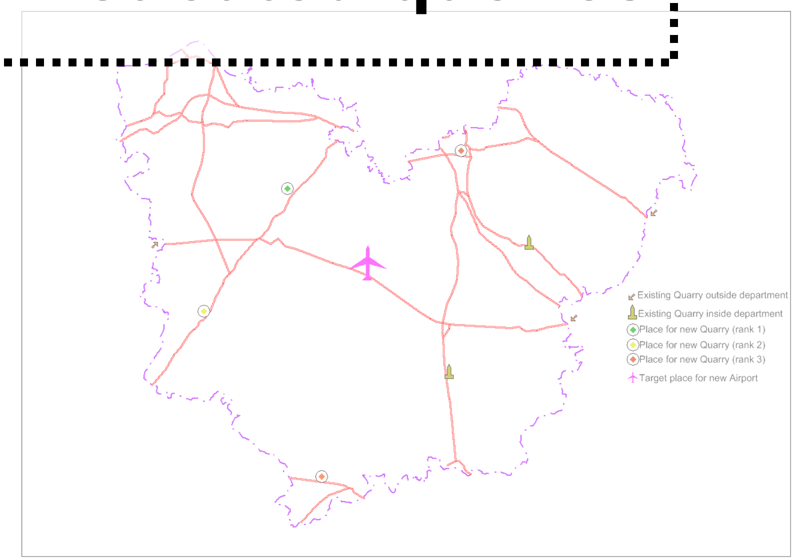




Use Case 3 - Use created complex map to make sophisticated queries

Get quarry best location

Substances	Quantity(*)
Sand	110 000 tons
Granite	50 000 tons
Concrete	200 000 tons



(*) All Quantity and Substances are given for example and are NOT real

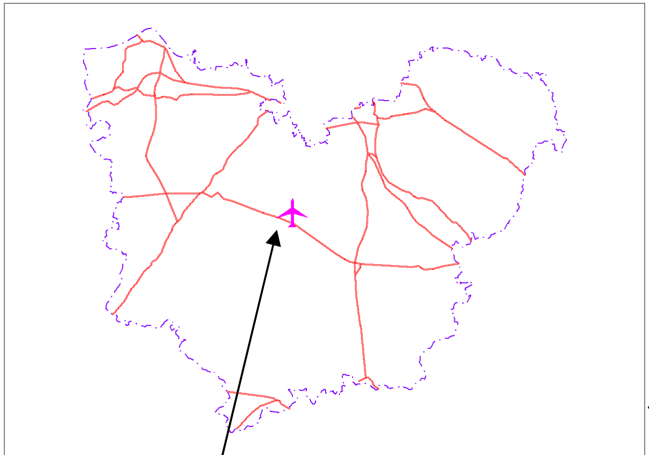
Use Case 3 – Integrate Multiple Criteria

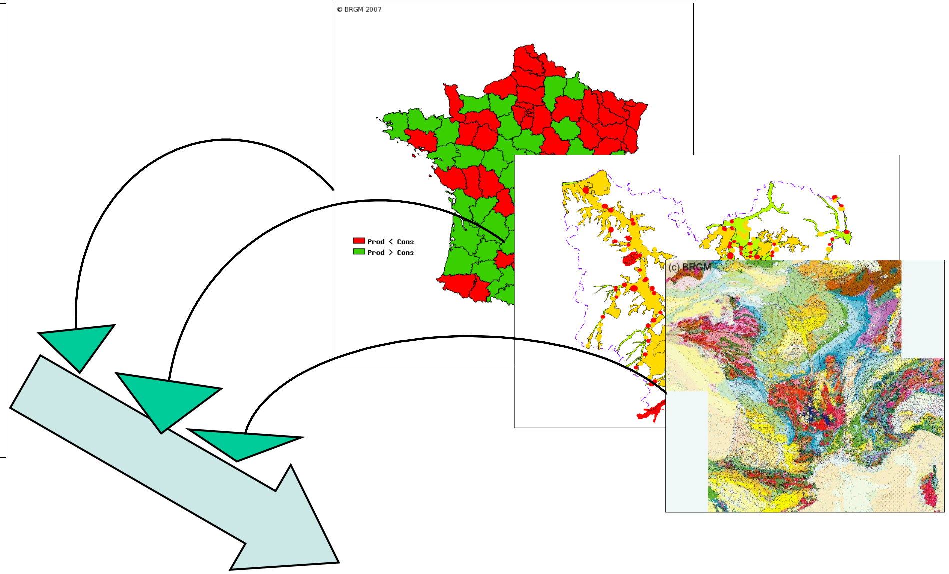
- Thematic Objective: Create an interactive map of the ranking according to combined criteria

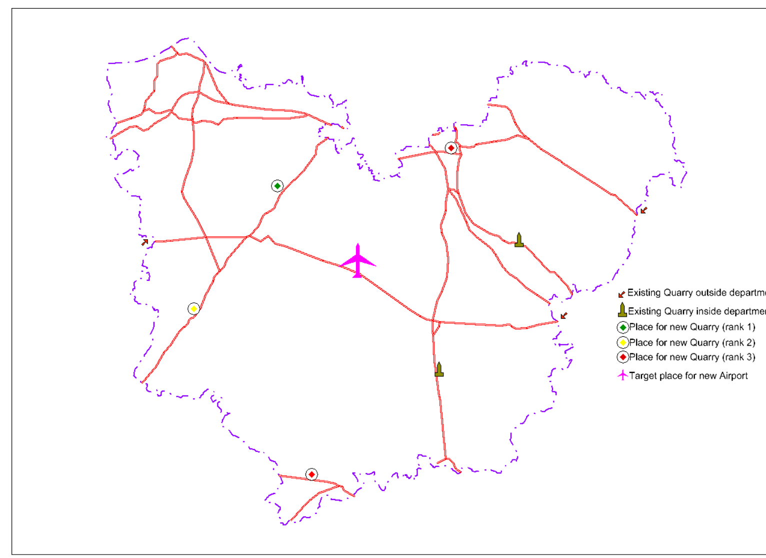
- Data Sources for UC3

- Technical challenges:
  - Extended the domain ontology to catch domain experts knowledge
  - Use geoprocessing facilities to compute spatial data needed for answering more sophisticated queries
  - Improved the annotation process towards semi-automatic annotation
  - Use of mediation for interoperability (service request parameters mediation)
  - Improve technical architecture of the end-user interface

Use Case 3 - Make sophisticated queries



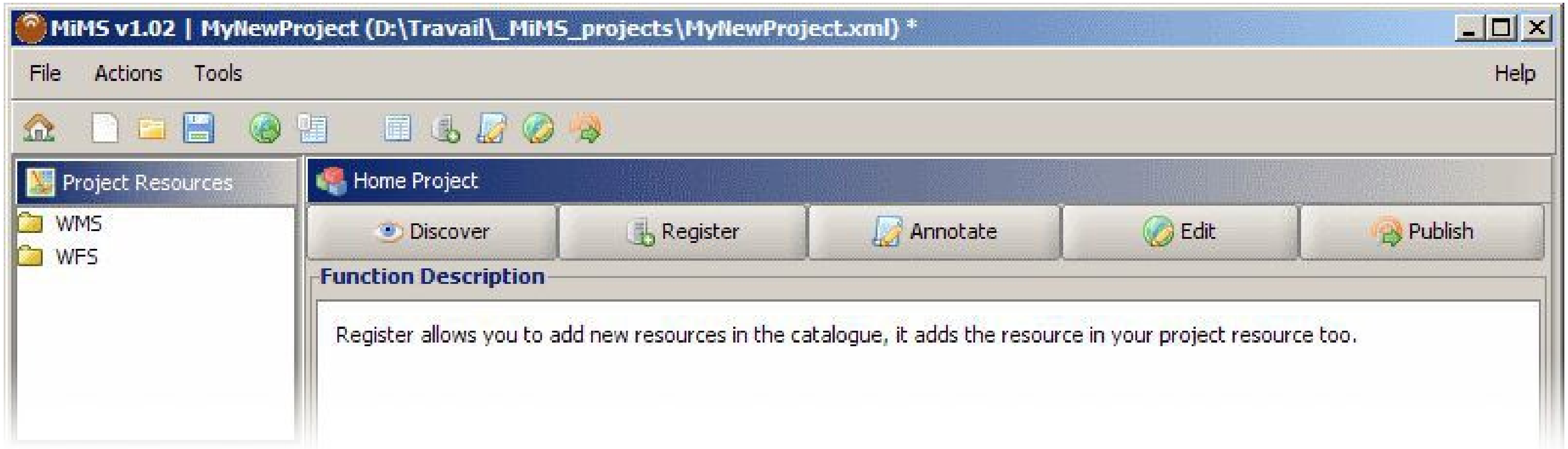




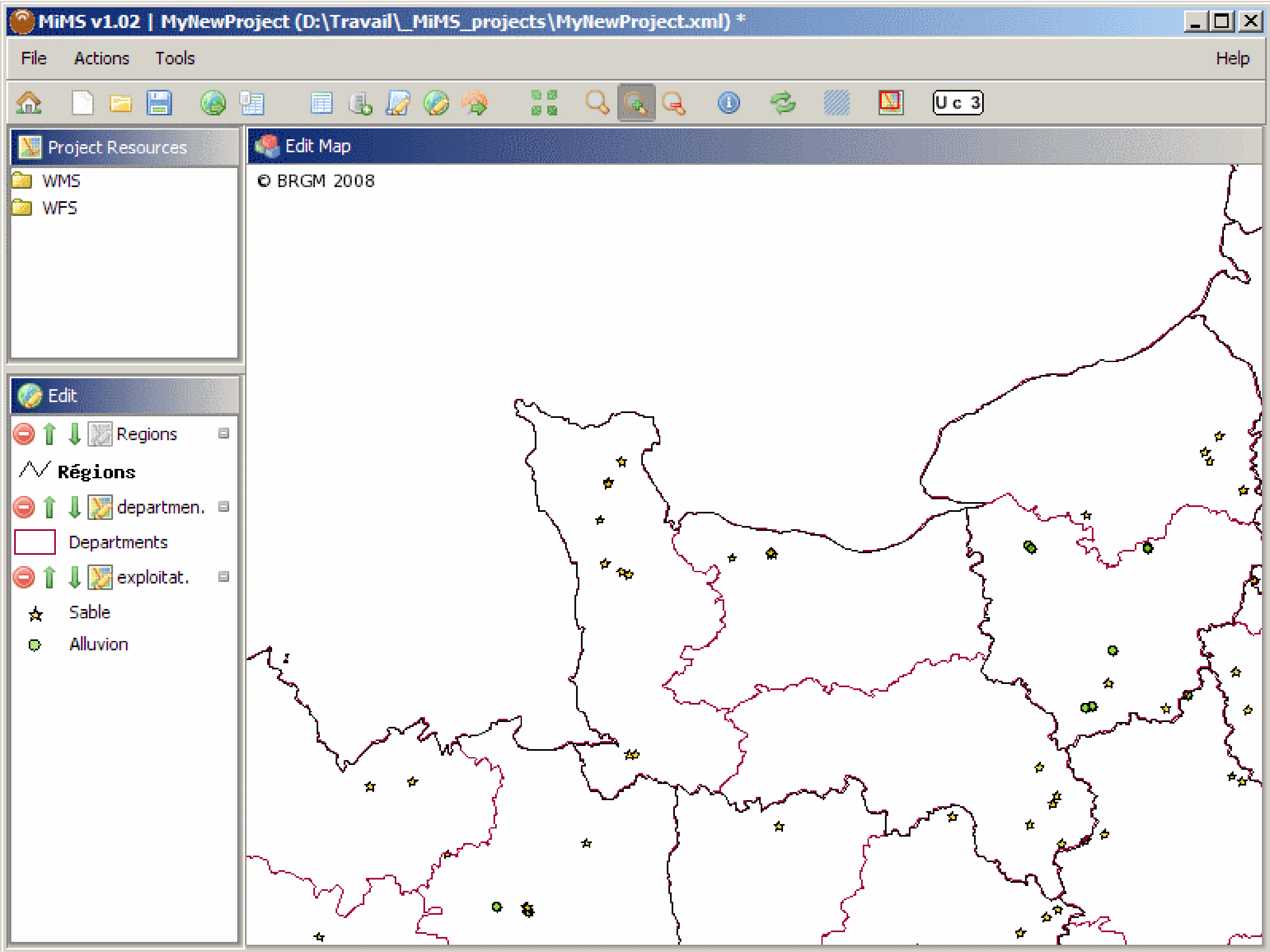
Substances	Quantity(*)
Sand	**110 000 tons**
Granite	**50 000 tons**
Concrete	**200 000 tons**

(*) All Quantity and Substances are given for example and are NOT real

Generating and Publishing the Website using MiMS



Generating and Publishing the Website using MiMS (cont')



Break

Overview

- Spatial Data Infrastructures
- Semantic Web Services
- Bridging the gap: Semantic Annotations
- SWING: use cases

- **SWING: Developed Tools**
- Demonstration (Videos)
- Hands-On Session

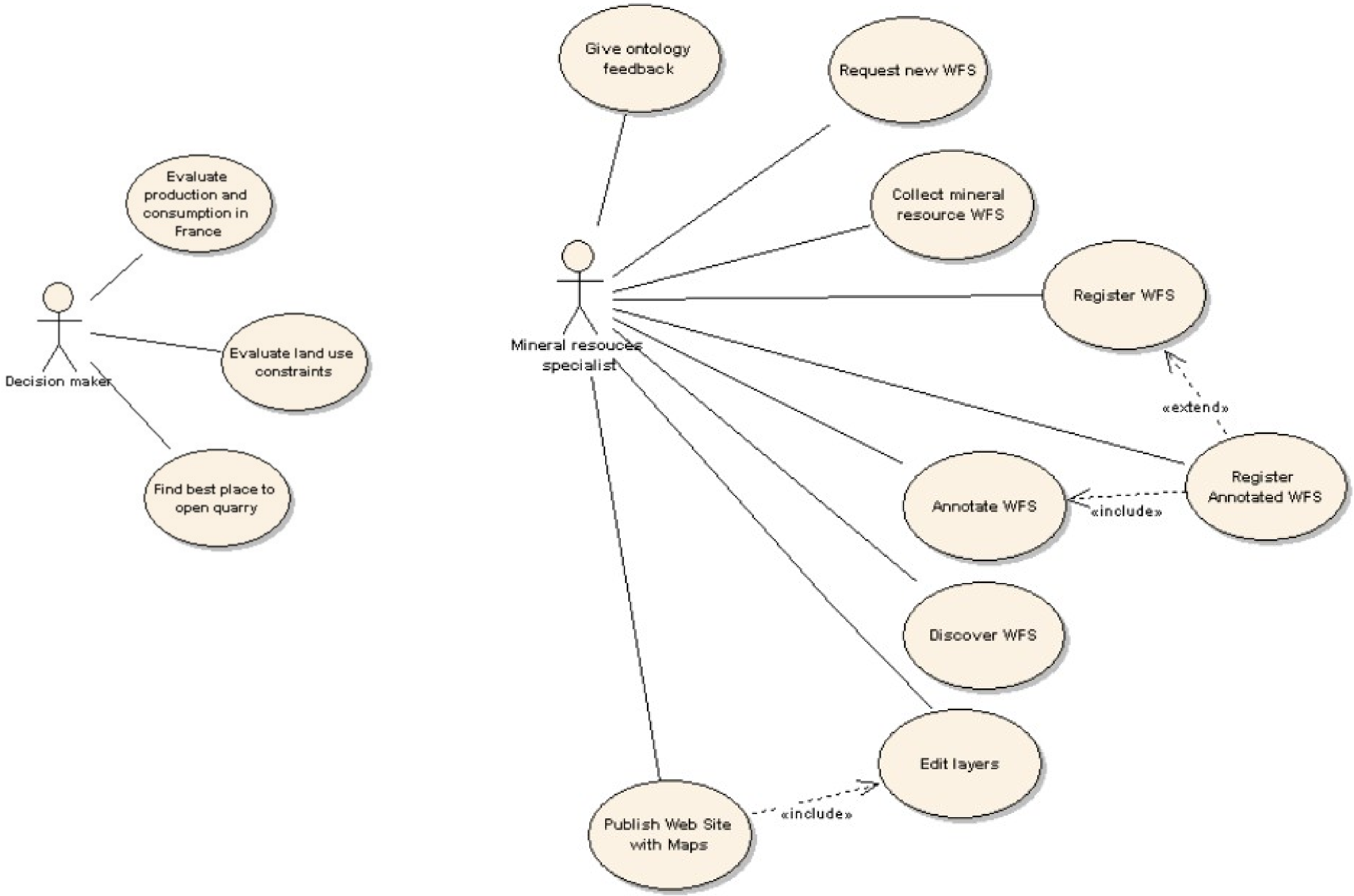
SWING Architecture Overview

GEOWS 2009
February 1st, 2009 - Cancun, Mexico

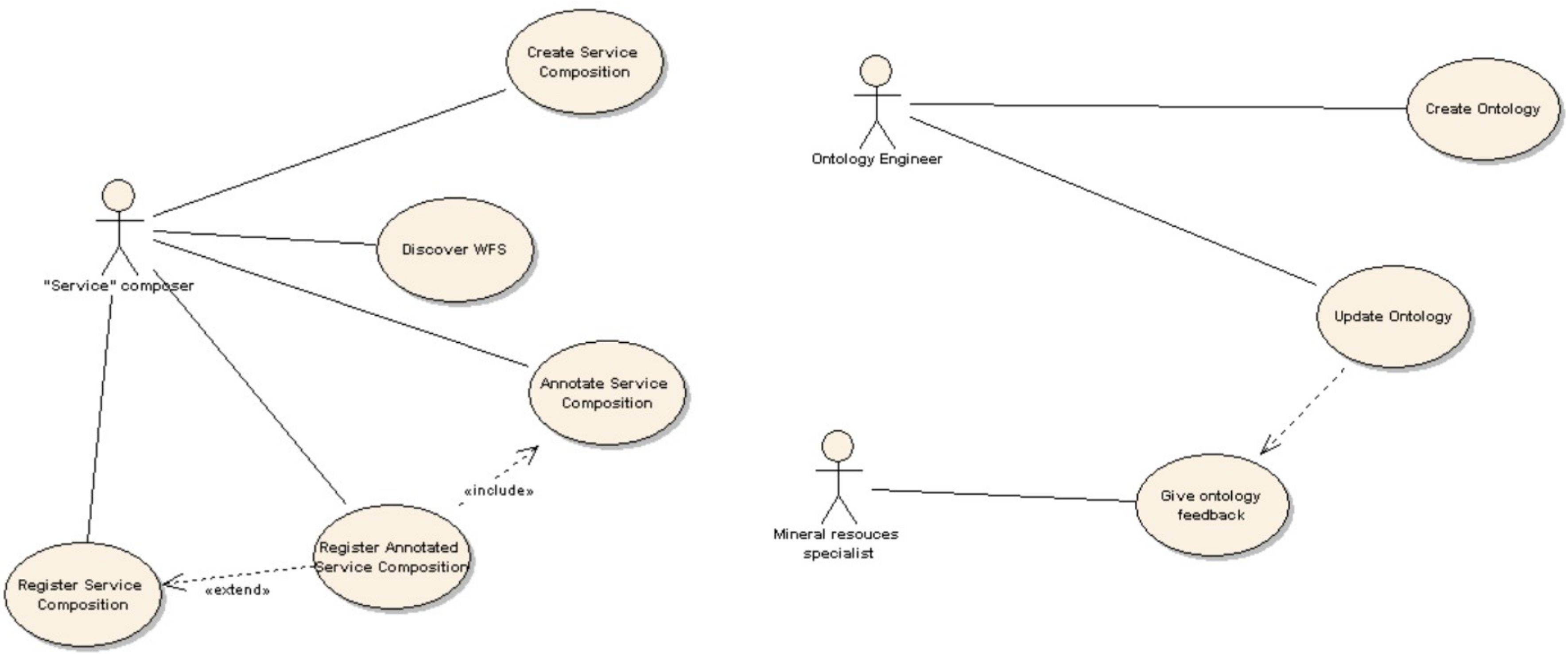
Outline

- Roles
- Components and Interactions

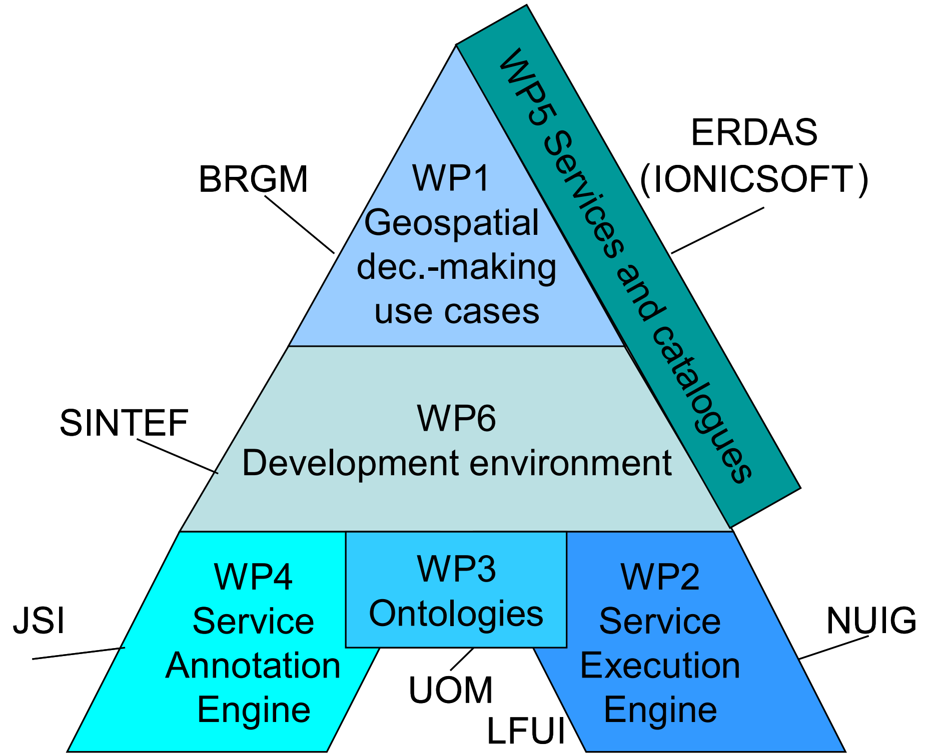
Roles – Decision Maker and Mineral Resource Specialist



Roles – Service Composer and Ontology Engineer



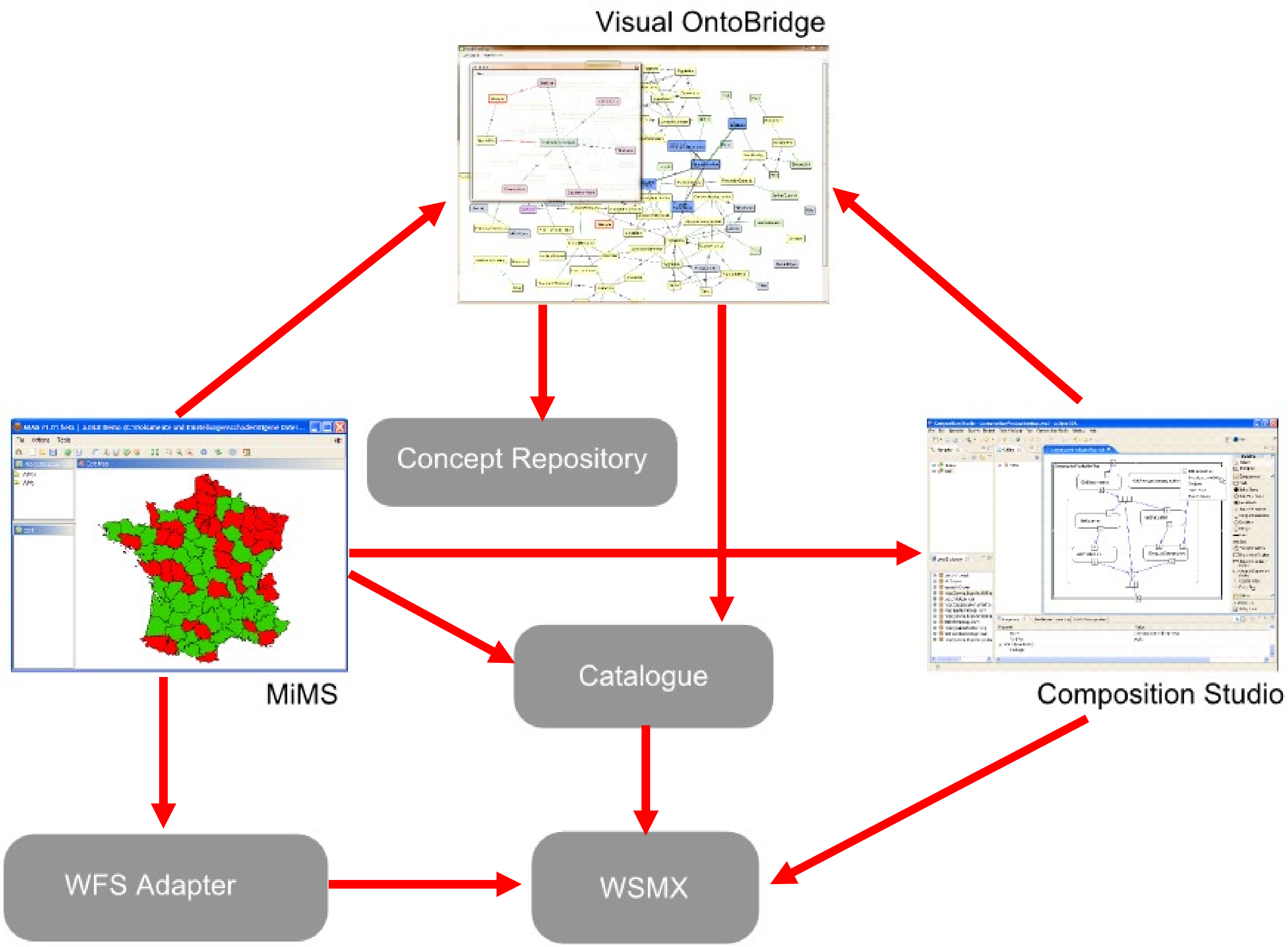
SWING Work Packages and Main Responsibilities



SWING Components

- **MiMS** (WP1): Environment for domain expert. Convenient semantic annotation & discovery; use composed services like standard OGC services
- **WSMX** (WP2): Semantic web services platform. Geospatial semantic discovery; execution of composed services (as ASMs)
- **Concept Repository** (WP3): Ontologies for semantic annotation. Used throughout components
- **Visual OntoBridge** (WP4): Annotation tool. Semi-automatic annotation of services and queries; provides user with most plausible annotations
- **Catalogue** (WP5): OGC Catalogue. Semantic discovery in interaction with WSMX; also provides adapter OGC $\leftrightarrow$ WSMX ASM execution
- **Composition Studio** (WP6): Environment for IT expert. Convenient semantic annotation & discovery; graphically compose services; automatic export into WSMX ASMs

High-level Architecture and Interactions



Overview

- Spatial Data Infrastructures
- Semantic Web Services
- Bridging the gap: Semantic Annotations
- SWING: use cases

- SWING: Developed Tools
- **Demonstration (Videos)**
- **Hands-On Session**

Hands-On Session

- **Exercise**: Building a workflow with Composition Studio
- **Task**: *Create a composition as described in the File CreatingWorkflowsSlides.pdf*

- **Exercise**: Creating a decision-support map with MiMS: Extract MiMS.zip in folder c:\MiMS and start launcher.bat
- **Tasks**:
  - *Try to follow the Steps in the mentioned Video*
  - *Create a „background" map (e.g. search for departments)*
  - *Find data about protected bird areas, quarries, mineral resources, ...*
  - *Create a legend and publish it as a website*
  - *Annotate an existing Web Feature Service about Birds*