Integrating Environmental Models on the Web

~ Challenges and Opportunities ~

Dumitru Roman

SINTEF, Norway

dumitru.roman@sintef.no

September 10, 2010

Outline

- Environmental Models
  - Examples in Hazard Risk Assessment: Oil Spill and Landslide
  - Models as (Web) Services and ENVISION
- Types of Web services
  - WSDL and SAWSDL
  - OGC Web services
  - RESTful Web services
- Lightweight semantic annotations for Web services: WSMO-Lite
  - Types of service semantics and WSMO-Lite service description model
  - Attaching semantic annotations to WSDL and RESTful Web services
- Summary and conclusions

Outline

- Environmental Models
  - Examples in Hazard Risk Assessment: Oil Spill and Landslide
  - Models as (Web) Services and ENVISION
- Types of Web services
  - WSDL and SAWSDL
  - OGC Web services
  - RESTful Web services
- Lightweight semantic annotations for Web services: WSMO-Lite
  - Types of service semantics and WSMO-Lite service description model
  - Attaching semantic annotations to WSDL and RESTful Web services
- Summary and conclusions

ICT for Environmental Services and Climate Change
EC Funding

- Overall aim: provide the foundations for an infrastructure for monitoring, predicting and managing the environment and its natural resources
  - Supporting flexible discovery and chaining of distributed environmental services
- EC funded projects:
  - **ENVISION**: Environmental Services Infrastructure with Ontologies
    www.envision-project.eu
  - **NETMAR**: Open service network for marine environmental data
    netmar.nersc.no/
  - **OEPI**: Exploring and Monitoring Any Organisation's Environmental Performance Indicators
    www.oepi-project.eu
  - **PESCADO**: Personalized Environmental Service Configuration and Delivery Orchestration
    www.pescado-project.eu
  - **SUDPLAN**: Sustainable Urban Development Planner for Climate Change Adaptation
    www.smhi.se/sudplan
  - **TATOO**: Tagging Tool based on a Semantic Discovery Framework
    www.tatoo-fp7.eu
  - **UncertWeb**: The Uncertainty Enabled Model Web
    www.uncertweb.org
  - **UrbanFlood**: UrbanFlood
    urbanflood.eu

What are Environmental Models?

- Computer models that aim to re-create what occurs during some event in nature
- Increasingly significant in decision making
  - Diagnose and examine causes and precursor conditions of events that have taken place (i.e. what happened and why it happened)
  - Forecast outcomes and future events (i.e., what will happen).
- Models are being developed by a wide variety of scientific and engineering disciplines
  - Many types of models, e.g. economic, behavioral, physical, engineering design, health, ecological, transport
  - Good models come from an assortment of disciplines
    - ***Increased interoperability between models is needed!***

Characteristics of Environmental Models

- Static, centralized, and closed systems
- Tightly coupled components
  - Integration of components requires significant work
  - Low level of reuse and sharing
- Isolated systems with limited audience (i.e. experts)
  - The growth of the system is planned
  - Limited possibilities for wider community involvement in model reuse and development

=> *There is a clear need for a dynamic, open, distributed and shared environmental modeling infrastructure that enables a high level of model reuse and is easily accessible for both experts and non-experts!*

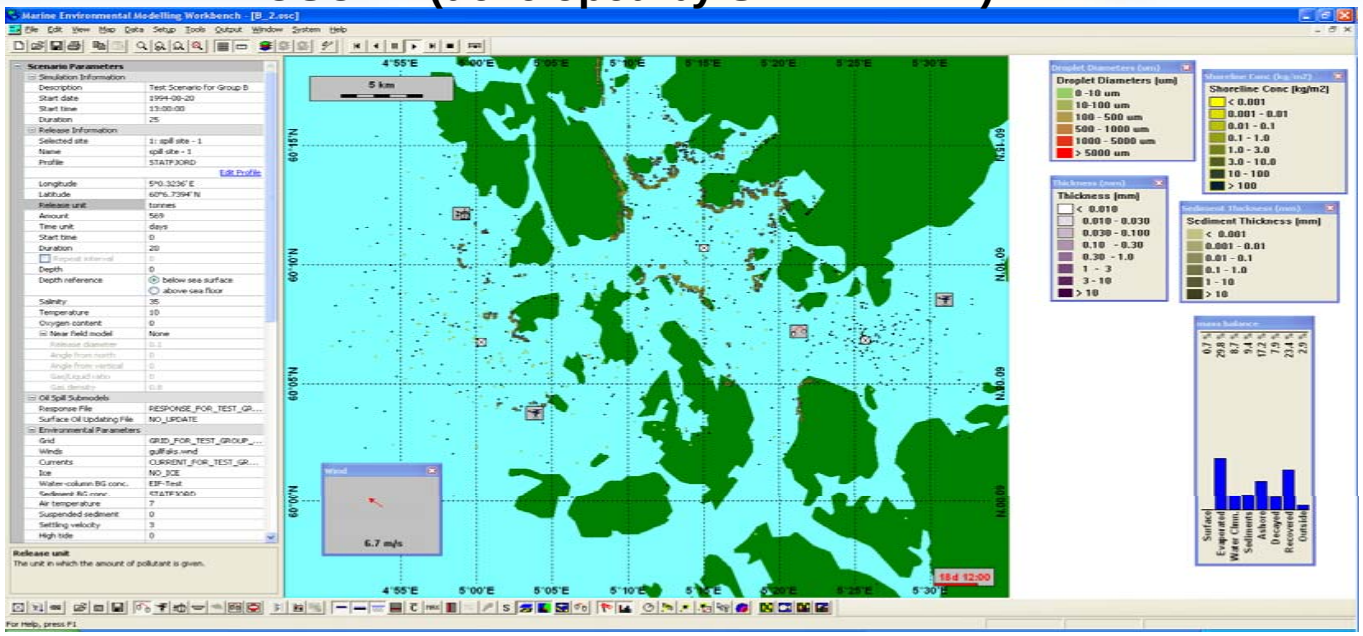
Hazard Risk Assesment in Oil&Gas Industry

- Difficult decisions ahead!
- A clear need for risk analysis tools to support:
  - strategic,
  - tactical, and
  - operational

  decision-making

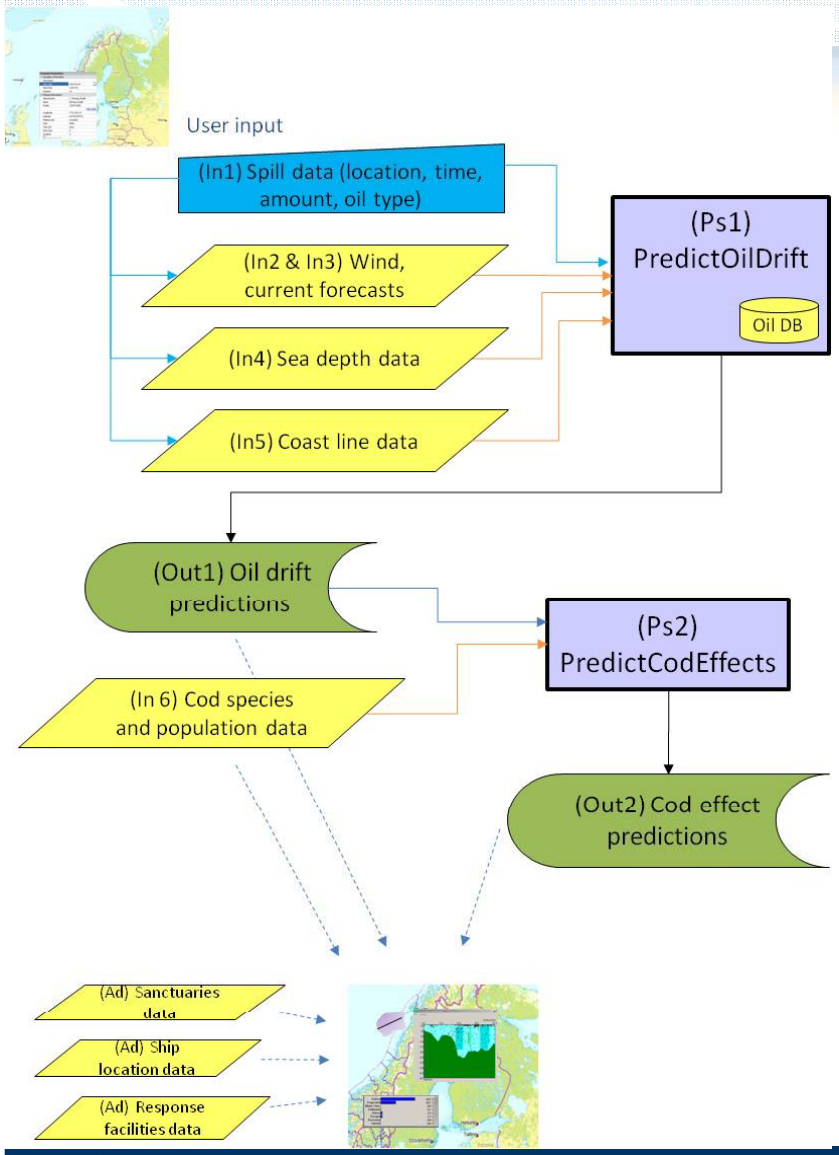
Oil Spill Contingency And Response (OSCAR)

OSCAR (developed by SINTEF MET)



- Examples Challenges:
  - *If I want to move my oil drift model to a new geographic region, how do I find appropriate data sources? And how can I trust that the data works with my model?*
  - *Given a real oil spill, where do I find up-to-date forecast data for my oil drift model? How can I utilise them by incorporating them within my workflow model and how can I trust that it works with it?*
  - *How can I look at biological effect models that are not "hardwired" into my model?*

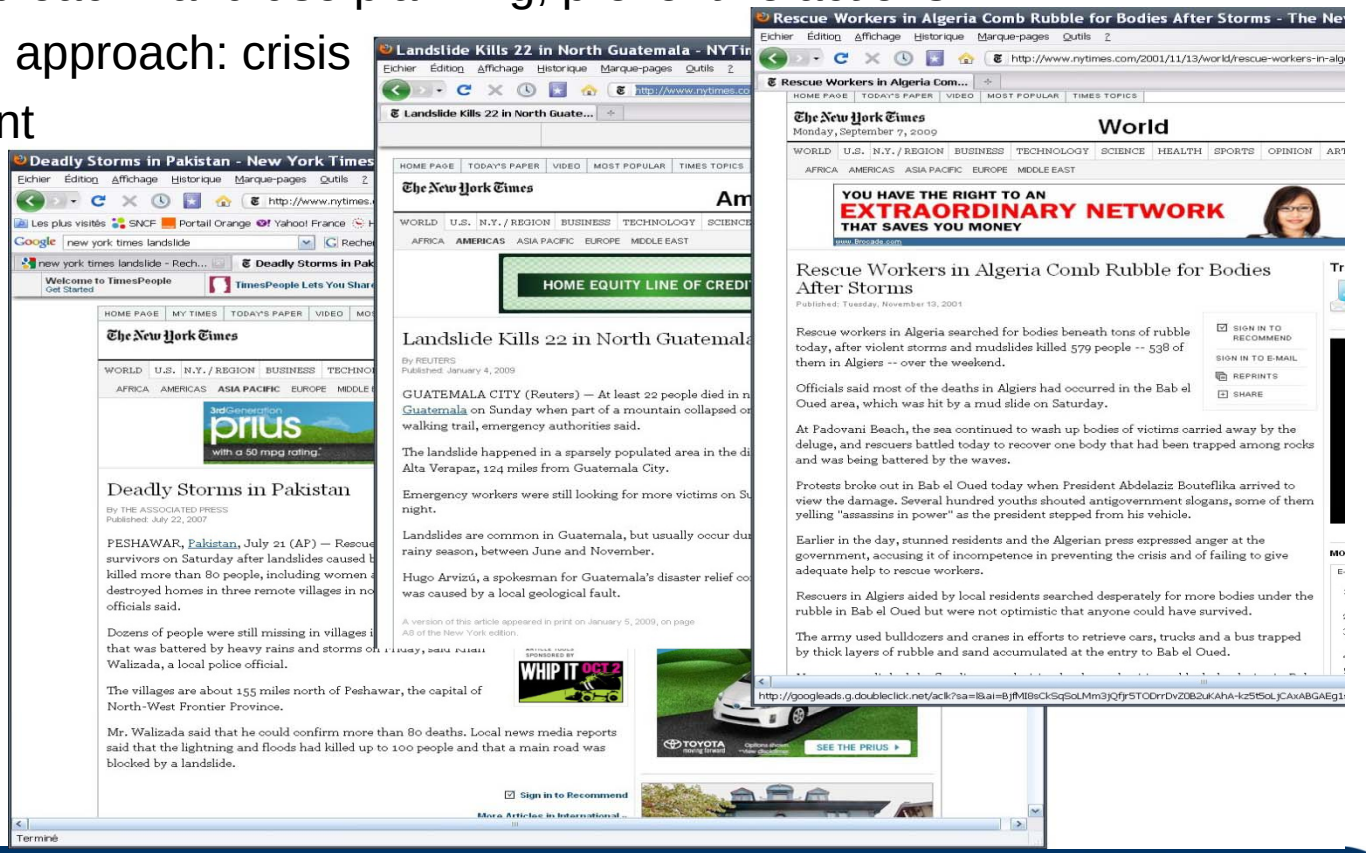
Environmental Models as (Web) Services



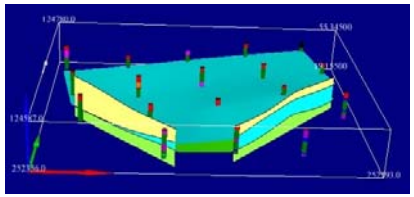
Input data			Model service	Output data		
Name		Data type		Name		Data type
In1	User input (location, time, amount, oil type)	Simple data types	Ms1 *PredictOilDrift*	Out1	Oil drift predictions: *Oil concentration in water column (3D)* *Oil slick position (2D)* *Mass balance (1D)*	Grids time series
In 2	Wind forecast	2D grid time series				
In 3	Current forecast	3Dgrid time series				
In4	Sea depth data	2D grid				
In 5	Coast line data	2D grid				
In6	Cod species and population data	Simple data types; 2D grid	Ms2 *PredictCodEffects*	Out2	Predictions of the effect on cod: *Lethality*	2D time series
Out1	Oil drift predictions	Grids time series				

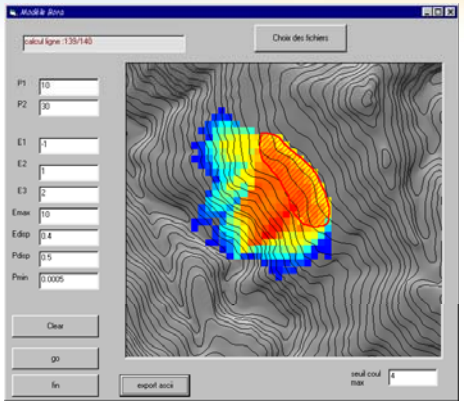
Landslide Hazard Risk Assessment

- A set of risk analysis tools for:
  - Strategic approach: global effects of climate change
  - Tactical approach: land use planning, preventive actions
  - Operational approach: crisis management



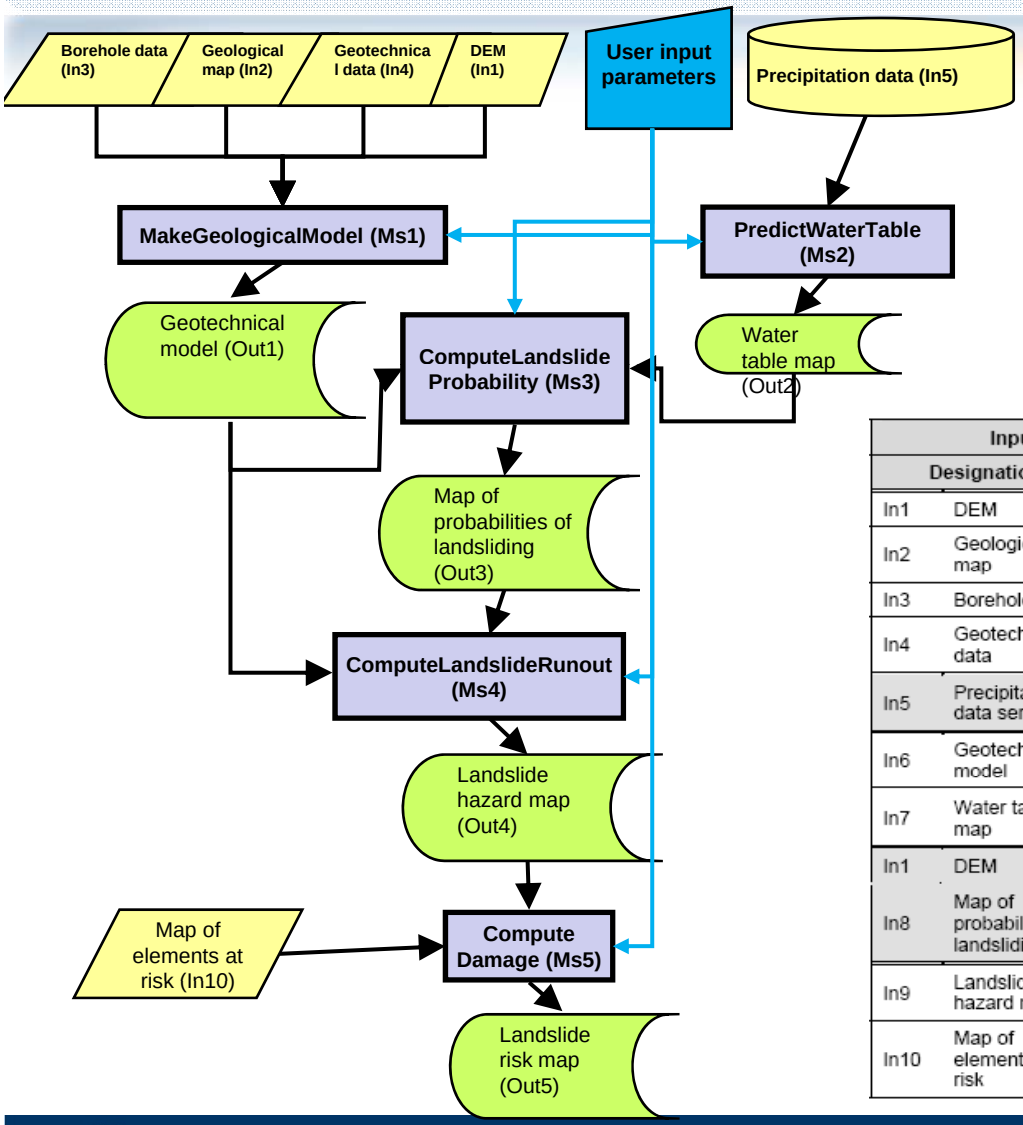
Landslide Environmental Models





- Example Challenges
  - *How can I find a landslide hazard / risk model for a given typology of landslide, using a given method, which computes a landslide hazard / risk map at a given scale?*
  - *How can I find input data such as a DEM, a geological map, geotechnical data, a landslide inventory, precipitation series, on my study area, with a given spatial or temporal resolution and which fit my landslide risk model?*
  - *How can I register a landslide hazard / risk modelling workflow and make it easy to discover?*
  - *Can a data service which fails in a landslide risk modelling workflow be replaced by another service providing data on the same study area, with the required spatial / temporal resolution and quality, during execution?*

Environmental Models as (Web) Services



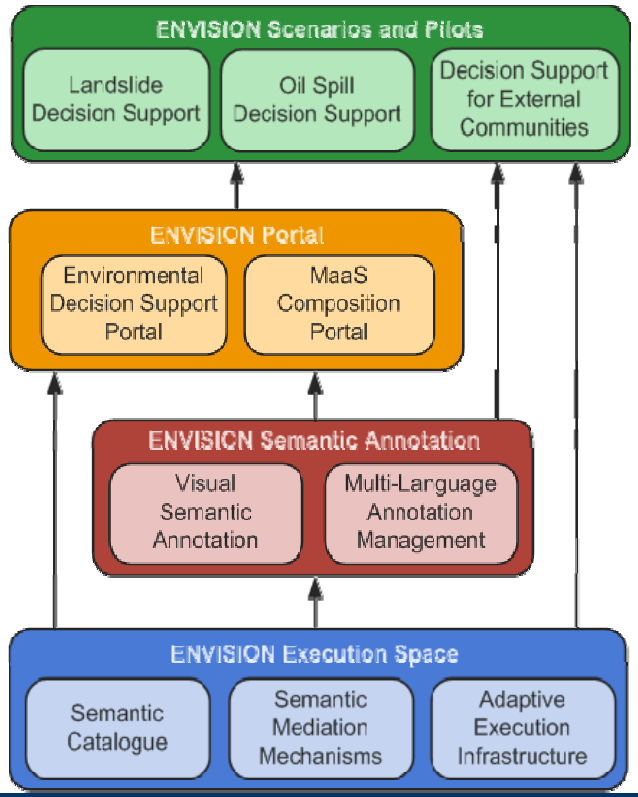
Input data			Processing service		Output data		
Designation		Format			Designation		Format
In1	DEM	Grid	Ps1	MakeGeologicalModel	Out1	Geological / geotechnical model	Grids
In2	Geological map	Vector					
In3	Borehole data	Points					
In4	Geotechnical data	Grids					
In5	Precipitation data series	Series	Ps2	PredictWaterTable	Out2	Water table map	Grid
In6	Geotechnical model	Grid	Ps3	ComputeLandslideProbability	Out3	Map of safety factors (SF) or map of probabilities of landsliding	Grid
In7	Water table map	Grid					
In1	DEM	Grid	Ps4	ComputeLandslideRunout	Out4	Landslide hazard map	Grid
In8	Map of probabilities of landsliding	Grid					
In9	Landslide hazard map	Grid	Ps5	ComputeDamage	Out5	Risk map	Vector
In10	Map of elements at risk	Vector					

ENVISION

http://www.envision-project.eu/

envision
environmental services infrastructure with ontologies

- ENVISION aims to provide a dynamic, open, distributed and shared environmental modeling infrastructure that enables a high level of model reuse and is easily accessible for both experts and non-experts
  - **Model as a Service (MaaS) = Models + SaaS + SWS**
- Portal with a pluggable decision support framework
  - Visual service chaining
  - Migration of existing models to MaaS
- Semantic annotation infrastructure
  - Visual semantic annotation mechanism
  - Multilanguage ontology management
- Execution space
  - Semantic discovery catalogue
  - Semantic service mediator
  - Adaptive service chaining execution



Generic scenario for ENVISION

Design time
(provide on-the-shelf modeling solutions)

- Discover existing resources
- Build the modeling workflow
- Register/Annotate the new Service

Set-up time
(connect the appropriate sources of information to feed the modeling service)

- Discover existing Modeling Services
- Select a region of interest
- Discover existing data sources
- Select the data sources
- Set the parameters
- Play the scenario

Execution time
(interact with the information provided by the models and monitor the system)

- Discover existing Modeling Services
- Select a region of interest
- Discover existing data sources
- Select the appropriate sensors data streams
- Select functional parameters for the alerting system

Semantic Annotations are a key enabler for discovery of services!

Outline

- Environmental Models
  - Examples in Hazard Risk Assessment: Oil Spill and Landslide
  - Models as (Web) Services and ENVISION
- Types of Web services
  - WSDL and SAWSDL
  - OGC Web services
  - RESTful Web services
- Lightweight semantic annotations for Web services: WSMO-Lite
  - Types of service semantics and WSMO-Lite service description model
  - Attaching semantic annotations to WSDL and RESTful Web services
- Summary and conclusions

Web Service Definition Language (WSDL)

http://www.w3.org/TR/wsdl20/

Schema (types)

Interface

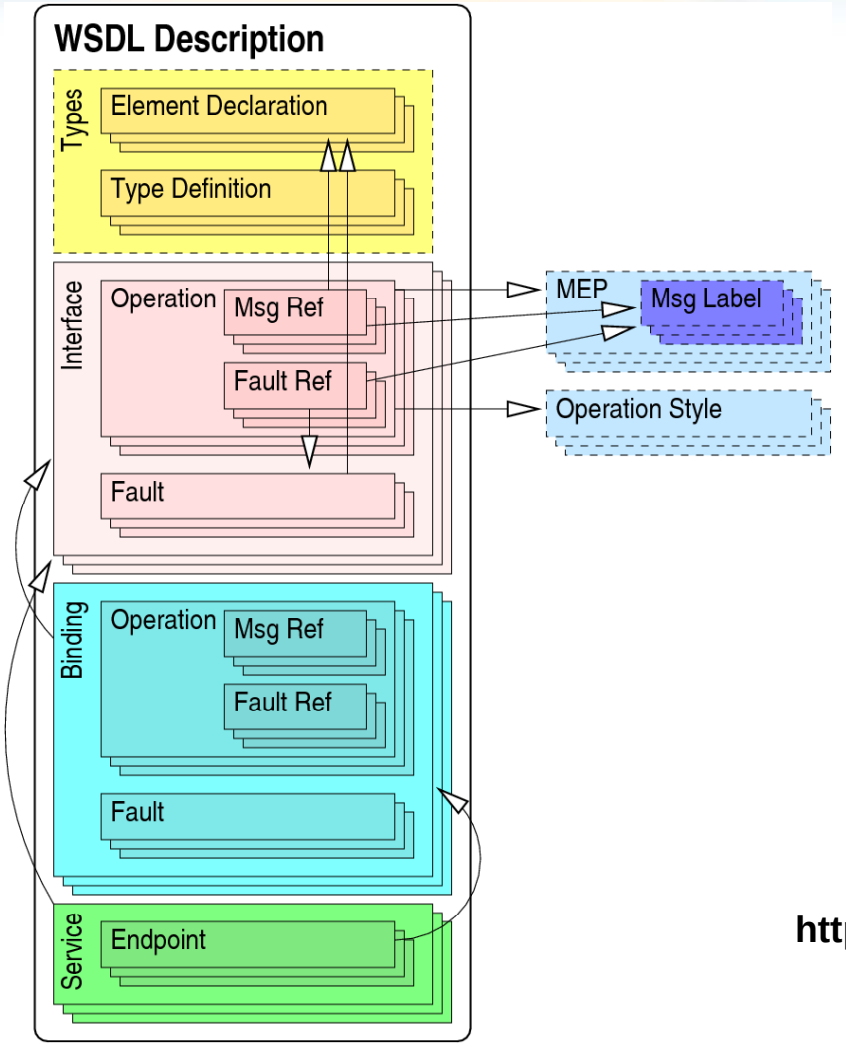
Operations

Binding

Network details

Service

Endpoints



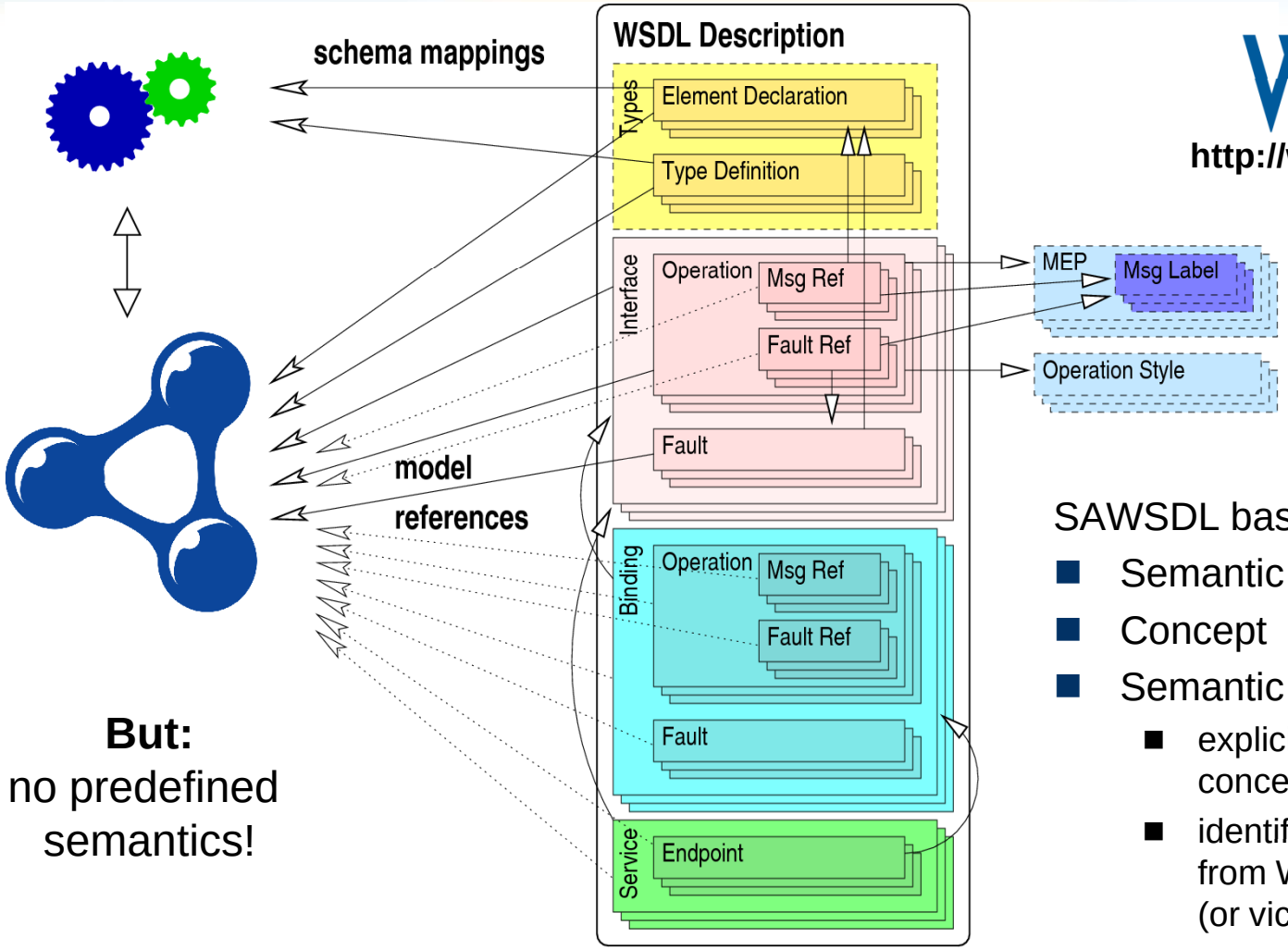
Message exchange patterns

Operation styles

W3C®

http://www.w3.org/

Semantic Annotations for WSDL and XML Schema (SAWSDL) http://www.w3.org/TR/sawsdl/



http://www.w3.org/

But:
no predefined semantics!

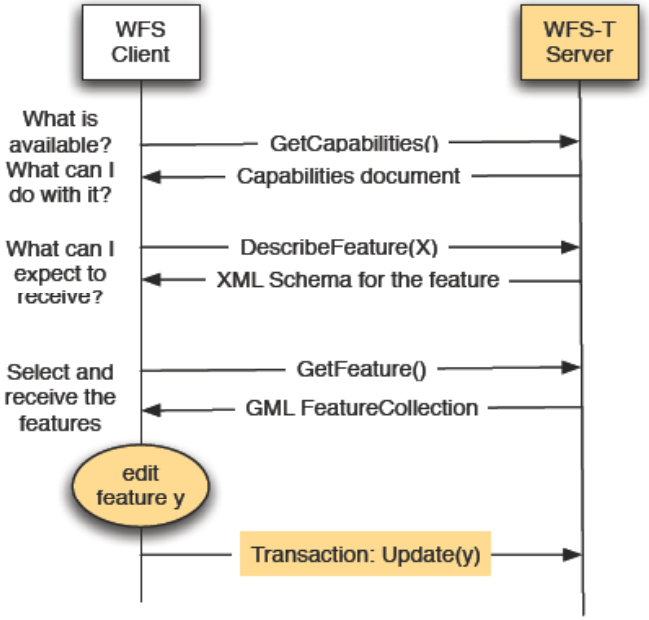
SAWSDL basic terminology:

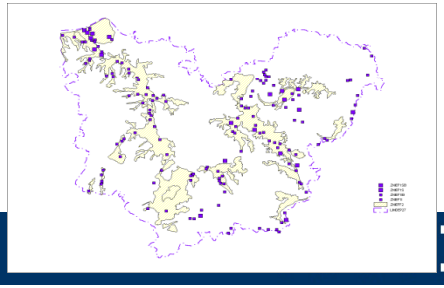
- Semantic Model
- Concept
- Semantic Annotation
  - explicit identifiers of concepts
  - identifiers of mappings from WSDL to concepts (or vice versa)

OGC Web services

- Interfaces for accessing and processing geospatial information
- Standardized by the Open Geospatial Consortium
- Web Mapping Service (WMS)
  - Visualisation of geospatial data
- Web Feature Service (WFS)
  - Specifies interface to retrieve Geographic Features
  - Features have Geometries and additional attributes
  - Data model specified in Feature Type Schema
- Web Processing Service (WPS)
  - Provides executable processes
  - No restrictions on input and output data
- Other standards
  - GML: Geography Markup Language (XML encoding of data)
  - SOS: Sensor Observation Service (real-time data)

OGC®
Open Geospatial Consortium, Inc.
http://www.opengeospatial.org/





RESTful Web services

- Naturally occurred in Web applications
  - Web applications' functionalities are increasingly used by programmatic clients, such as other Web applications
- A RESTful Web service is:
  - A set of interlinked Web resources
  - Data-centric, not functionality-centric
  - Machine-oriented (no more HTML-scraping)
- Services described in HTML documentation
- Not (yet) standardized
  - But based on HTTP, XML, and the Web architecture
- E.g.: flickr, del.icio.us, facebook APIs
  - Some more RESTful than others

Other types of services or related initiatives

- Service oriented architecture Modeling Language (SoaML)
  - An emerging OMG standard for describing services using UML

http://www.soaml.org

- SOA activities at the Organization for the Advancement of Structured Information Standards (OASIS)
  - SOA Reference Model
  - SOA Reference Architecture
  - Semantic Execution Environment

OASIS

http://www.oasis-open.org

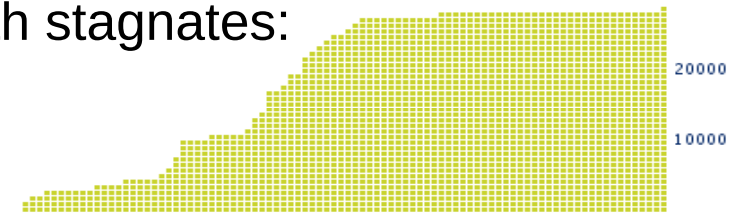
- SOA activities at The Open Group
  - SOA Ontology, Reference Model, Architecture

THE Open GROUP

http://www.opengroup.org

Web services: A reality check

- Numbers of WSDL services
  - Number of unique public WSDL-based services < 30.000
  - WSDL growth stagnates:



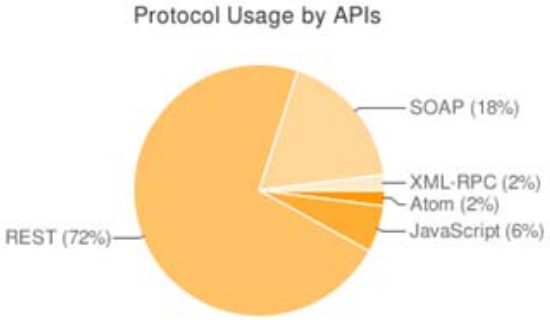
Number of Web services found during the past 45 months
[seekda.com, May 2010]

  - Large, but unknown, number of Intranet and enterprise services

- Numbers of OGC Web services
  - Number of public WMS, WFS, WPS, SOS services < 9.000 (most of them being WMS) [Google search,, May 2010]
  - Potentially large, but unknown, number of OGC services deployed inside organizations

Web services: A reality check (cont')

- Significant growth of Web APIs (RESTful Web services)
  - > 2000 Web APIs on ProgrammableWeb.com

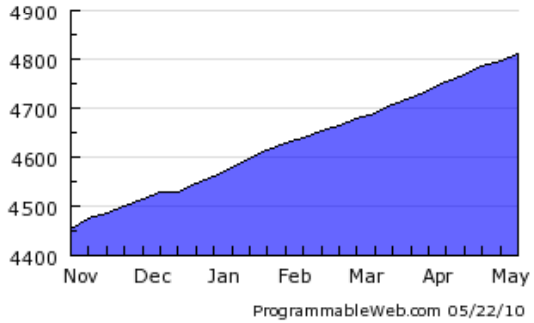


[ProgrammableWeb.com, May 2010]

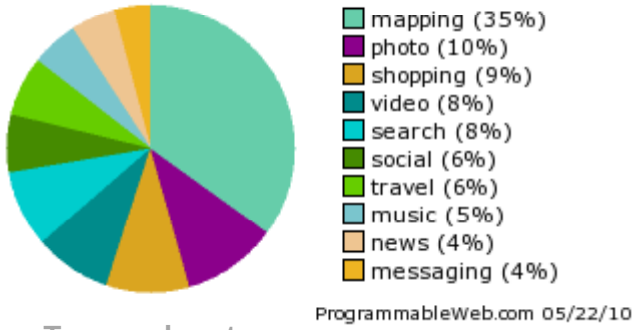
Most service interfaces are proprietary Web sites, or FTP downloads of ASCII files

Most services on the Web do not have machine-readable interface descriptions!

  - > 4.800 Mashups on ProgrammableWeb.com (combining Web APIs from one or more sources)



New mashups during the last 6 months

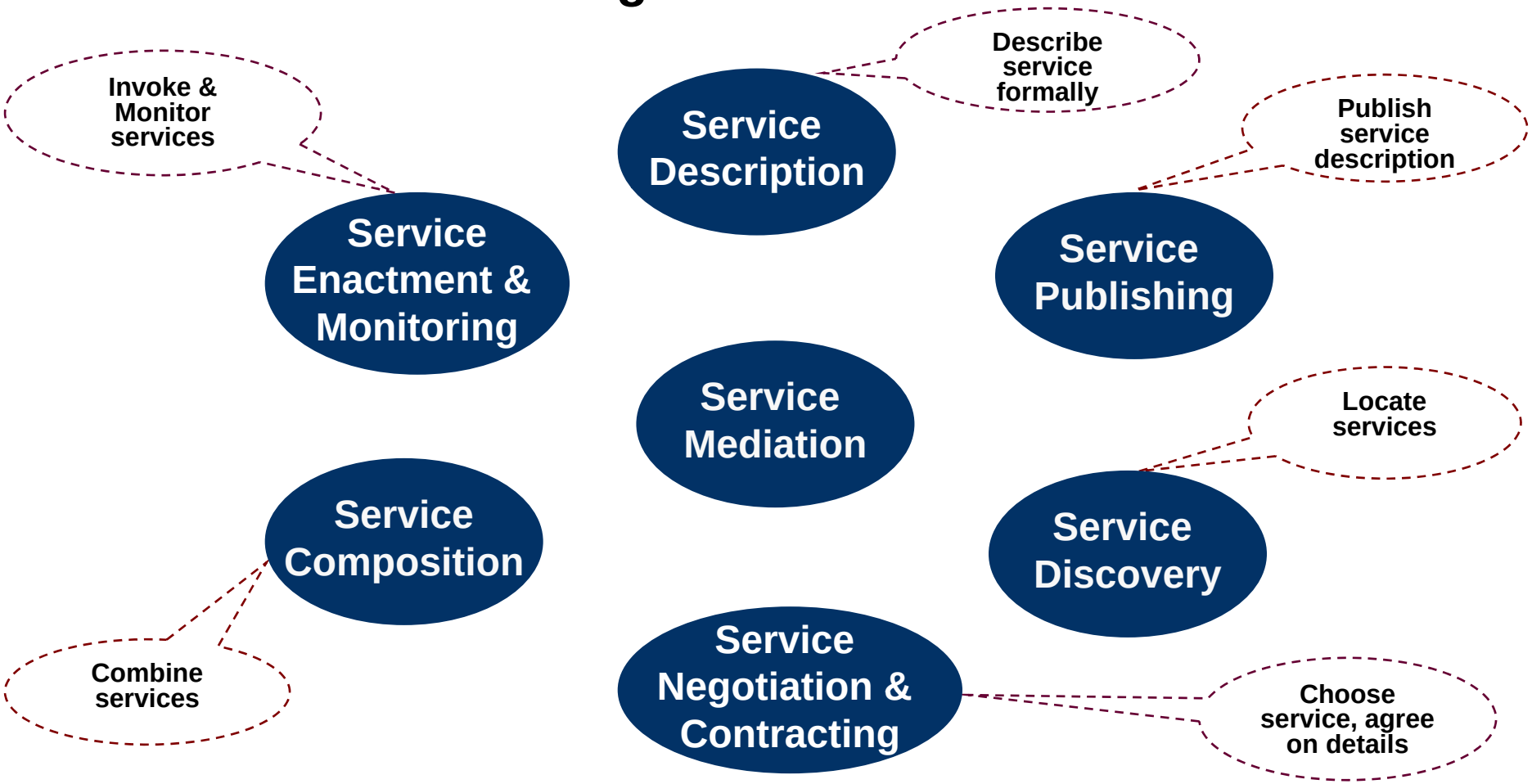


Top mashup tags

The World Wide Web is more and more used for application to application communication. The programmatic interfaces made available are referred to as *Web services*. (W3C)

Why semantics for Web services?

Automate service usage



Outline

- Environmental Models
  - Examples in Hazard Risk Assessment: Oil Spill and Landslide
  - Models as (Web) Services and ENVISION
- Types of Web services
  - WSDL and SAWSDL
  - OGC Web services
  - RESTful Web services
- Lightweight semantic annotations for Web services: WSMO-Lite
  - Types of service semantics and WSMO-Lite service description model
  - Attaching semantic annotations to WSDL and RESTful Web services
- Summary and conclusions

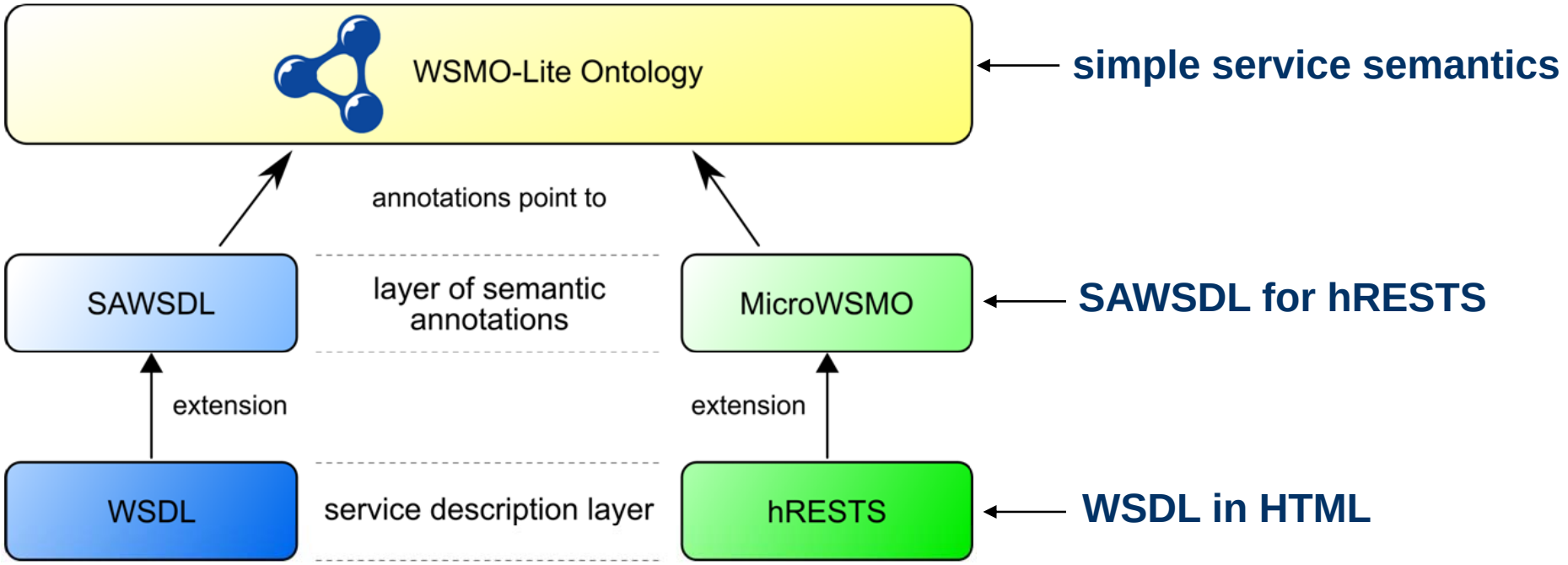
Lightweight semantic annotations for Web services

- Existing approaches (OWL-S, WSMO) are (perceived as) complex in terms of
  - Modeling (adoption coming slowly)
  - Computational complexity (expressivity vs. performance)
- Core ideas:
  - Reduce complexity: Scale down, modularize ⇒ **WSMO-Lite ontology**
  - Include RESTful services: Not only WSDLs are services ⇒ **MicroWSMO**
- Goal: provide a SAWSDL-based unifying frame work (WSMO-Lite) for lightweight semantic descriptions of WSDL and RESTful Web services, in order to support automation of Web service tasks such as discovery, selection, and composition.

  "a little semantics goes a long way"

WSMO-Lite

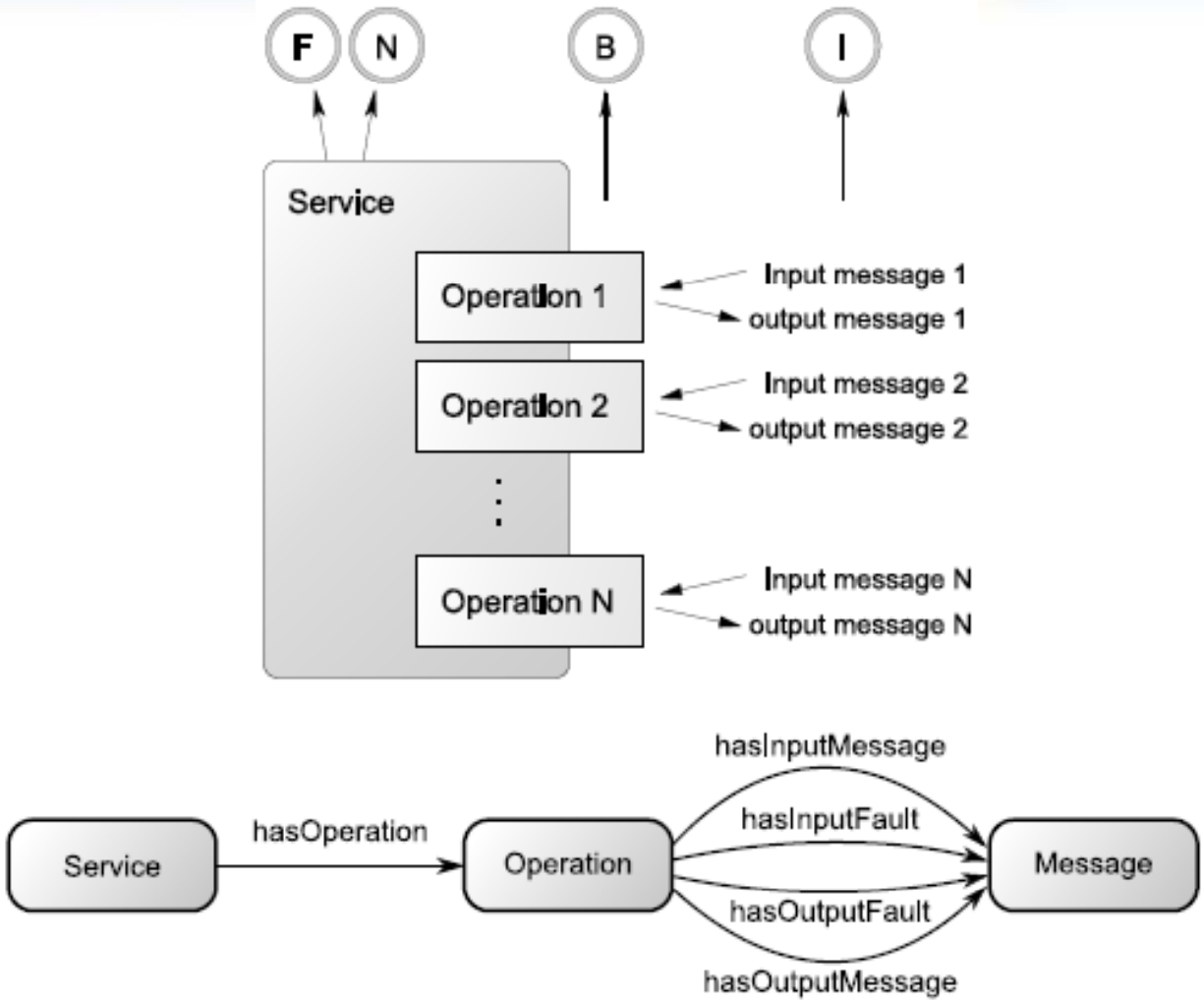
- WSMO-based annotation mechanism for WSDL and RESTful services
- Design principles: proximity to underlying standards, inclusion of RESTful services, minimality, lightweight, modularity



Types of Service Semantics in WSMO-Lite

- **Functional**
  - What the service does
- **Behavioral**
  - How the client talks to the service
- **Information model**
  - For handling data
  - Incl. lifting/lowering
- **Nonfunctional**
  - Policies, QoS, price, location etc.

WSMO-Lite Web service description model

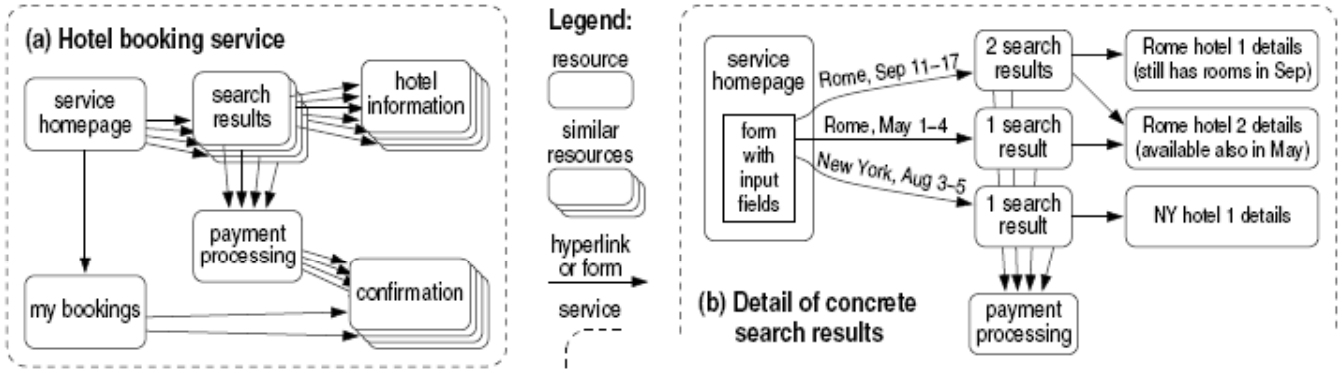


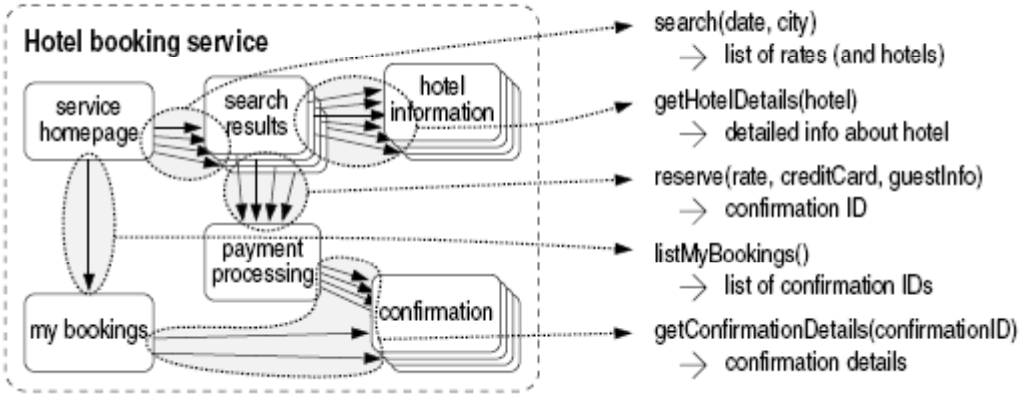
Modeling RESTful services

- Mapping of RESTful services to WSMO-Lite service model:

RESTful services	WSMO-Lite service model
Service (*a group of resources*)	Service
Resource	– (*disregarded on the semantic level*)
Resource method	Operation
Method request/response	Operation input/output
Hyperlink	– (*treated as part of message data*)

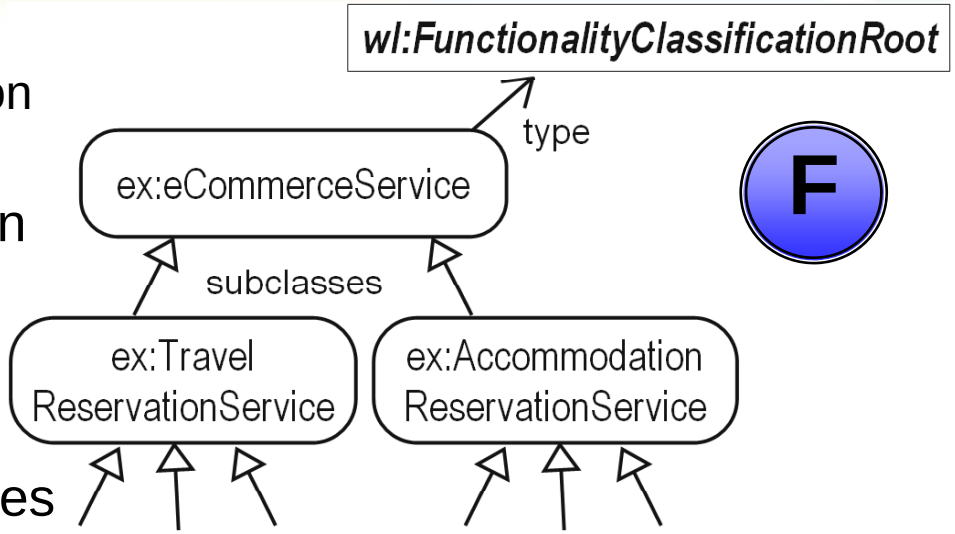
- Example:





Functional and nonfunctional semantics

- Functional Semantics
  - For service discovery, composition
  - *Category*
    - Functionality categorization
    - E.g. eCl@ss
  - *Capability*
    - *wl:Condition, wl:Effect*
    - Using WSML rule languages



- Nonfunctional Semantics
  - For ranking and selection
  - Not constrained, any ontologies
  - Example:

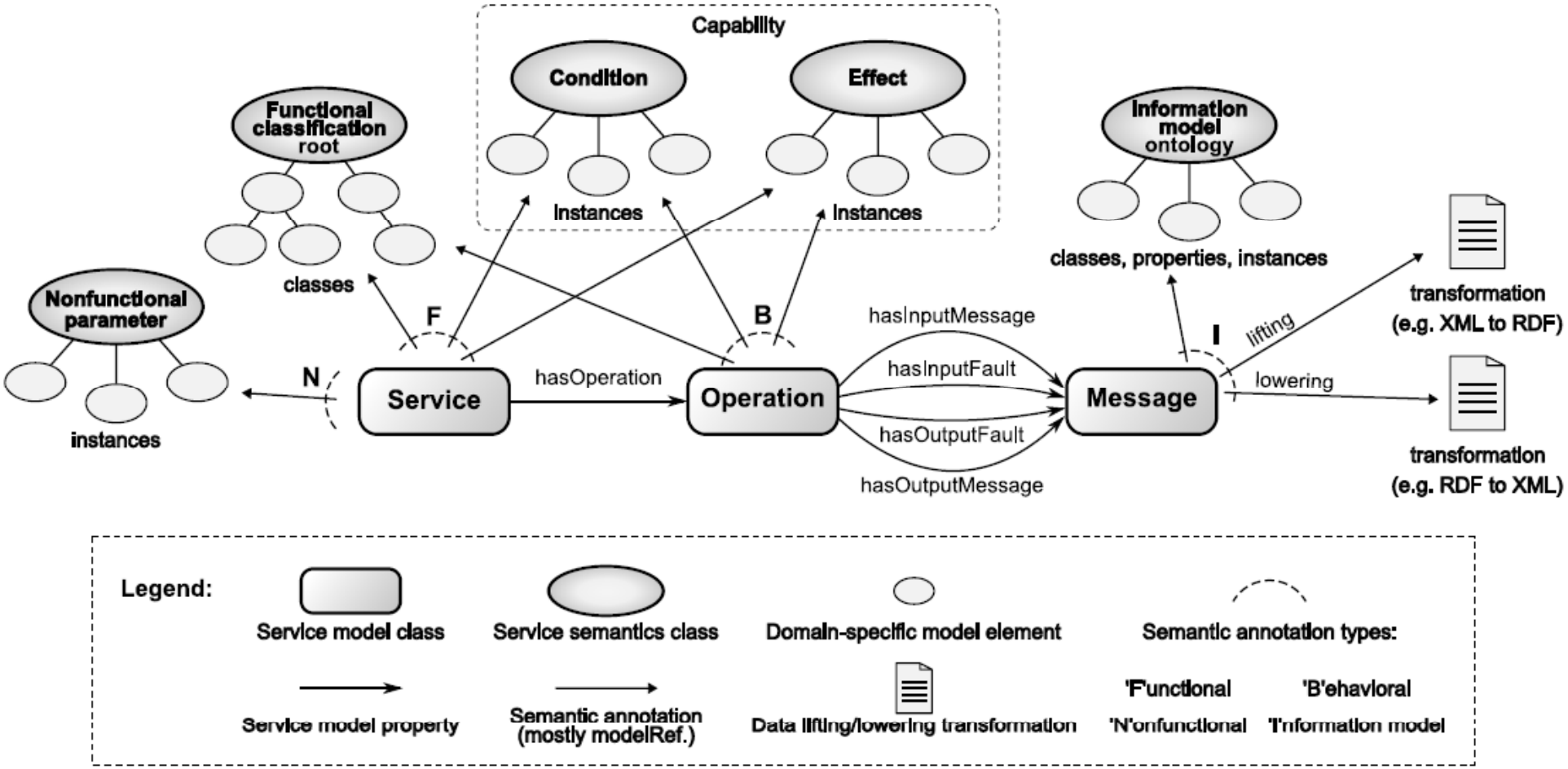
```
ex:PriceSpecification
        rdfs:subClassOf  wl:NonFunctionalParameter .
ex:ReservationFee
        rdf:type  ex:PriceSpecification ;
        rdf:value  "15"^^ex:euroAmount .
```

Behavioral and information semantics

- Behavioral Semantics
  - For invocation, composition, process mediation
  - Functionalities on operations
    - Capabilities, categories
  - Client selects operation to invoke next
    - Instead of being strictly guided by an explicit process

- Information Semantics
  - For invocation, composition, data mediation
  - Not constrained, any ontologies
  - Marked as *wl:Ontology*

Annotating the WSMO-Lite service model



Annotating WSDL with WSMO-Lite

Sem. type	WSMO-Lite svc. model	WSDL component
F	Service	Service or Interface
N	Service	Service
B	Operation	Interface Operation
I	Message	Element Declaration or Type Definition

```
<wsdl:service name="RomaHotels" interface="HotelReservation"
    sawsdl:modelReference="http://example.org/onto#AccommodationF
                http://example.org/onto#RomaHotelsPrecondition
                http://example.org/onto#RomaHotelsEffect" >
  <wsdl:endpoint ... />
</wsdl:service>
```

```
<wsdl:interface name="HotelReservation"
    sawsdl:modelReference="http://example.org/onto#AccommodationF
                http://example.org/onto#HotelReservationPrecondition
                http://example.org/onto#HotelReservationEffect" >
  ... <!-- operations come here -->
</wsdl:interface>
```

```
<wsdl:service name="RomaHotels" interface="HotelReservation"
  sawsdl:modelReference="http://example.org/onto#RomaHotelsPricePerRese
</wsdl:service>
```

```
<wsdl:interface name="HotelReservation">
  <wsdl:operation name="search"> ... </wsdl:operation>
  <wsdl:operation name="reserve"
      sawsdl:modelReference="http://example.org/onto#EffectReservationConfirmed">
    ...
  </wsdl:operation>
  <wsdl:operation name="cancelReservation"
      sawsdl:modelReference="
            http://example.org/onto#PreconditionReservationConfirmed
            http://example.org/onto#EffectReservationCancelled">
```

```
  <wsdl:types><xs:schema ...>
    <xs:element name="searchRequest"
        sawsdl:modelReference="http://example.org/onto#ReservationDates
                               http://example.org/onto#Destination"
        sawsdl:loweringSchemaMapping="http://example.org/searchRequest.xsp" >
      ...
    </xs:element>
  </xs:schema></wsdl:types>

  <wsdl:interface name="HotelReservation">
    <wsdl:operation name="search">
      <wsdl:input element="searchRequest" />
      <wsdl:output element="listOfRates" />
    </wsdl:operation>
    ...
  </wsdl:interface>
```

Annotating RESTful Services with WSMO-Lite

- No accepted WSDL for RESTful services
  - But there's usually an HTML page
- Adding machine-readable annotation
  - **hRESTS**
- And hooks for F,N,B,I semantics
  - **MicroWSMO**

hRESTS

- A typical RESTful Web service description:

Operation description	HTML source
ACME Hotels service API	ACME Hotels service API
Operation getHotelDetails	Operation getHotelDetails
Invoked using the method GET at	Invoked using the GET at
`http://example.com/h/{id}`	<code>http://example.com/h/{id}</code>
Parameter: `id` - the identifier of	Parameters: <code>id</code>
the particular hotel	- the identifier of the particular hotel
Output value: hotel details in an	Output value: hotel details in
`ex:hotelInformation` document	an <code>ex:hotelInformation</code> document

- hRESTS - HTML Classes
  - service
  - operation
  - address
  - method
  - input
  - output
  - label

```
1   <div class="service" id="svc">
2     <h1><span class="label">ACME Hotels</span> service API</h1>
3     <div class="operation" id="op1">
4       <h2>Operation <span class="label">getHotelDetails</span></h2>
5       <p> Invoked using the <span class="method">GET</span>
6         at <code class="address">http://example.com/h/{id}</code><br/>
7         <span class="input">
8           <strong>Parameters:</strong>
9           <code>id</code> – the identifier of the particular hotel
10        </span><br/>
11        <span class="output">
12          <strong>Output value:</strong> hotel details in an
13          <code>ex:hotelInformation</code> document
14        </span>
15      </p>
16  </div></div>
```

MicroWSMO semantic annotations

- An extension of hRESTS
  - *model* indicates that the link is a model reference
  - *lifting* and *lowering* then denote links to the respective data transformations
- Example:

```
<div class="service" id="svc">
  <h1><span class="lab
    <p>This service is a
      <a rel="model" hre
        hotel reservation<
    </p>
  <div class="operation"
    <h2>Operation <spa
    <p> Invoked using the
      at <code class="add
      <span class="input"
        <strong>Paramete
        <a rel="model" hr
          <code>id</cod
          (<a rel="lowerir
          lowering</a>)
      </span><br/>
      <span class="output
        <strong>Output v
        <code>ex:hotelInfo
      </span>
    </p>
</div></div>
```

```
@prefix ex: <http://example.com/serviceDescription.html#> .
@prefix hr: <http://www.wsmo.org/ns/hrests#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sawsdl: <http://www.w3.org/ns/sawsdl#> .
@prefix wsdlx: <http://www.w3.org/ns/wsdl-extensions#> .

ex:svc a wl:Service ;
  rdfs:isDefinedBy <http://example.com/serviceDescription.html> ;
  rdfs:label "ACME Hotels" ;
  sawsdl:modelReference <http://example.com/ecommerce/hotelReservation> ;
  hr:hasOperation ex:op1 .
ex:op1 a wl:Operation ;
  rdfs:label "getHotelDetails" ;
  hr:hasMethod "GET" ;
  sawsdl:modelReference wsdlx:SafeInteraction .
  hr:hasAddress "http://example.com/h/{id}"^^hr:URITemplate ;
  wl:hasInputMessage [
    a wl:Message ;
    sawsdl:modelReference <http://example.com/data/onto.owl#Hotel> ;
    sawsdl:loweringSchemaMapping <http://example.com/data/hotelID.xsparql>
  ] ;
  wl:hasOutputMessage [
    a wl:Message ;
  ] .
```

Automation algorithms

Algorithm: matchmaker for WSMO-Lite functional classifications
Inputs: set S of known services annotated with functionality categories,
goal category c_g
Result: the set M of tuples $\langle$*service, match degree from* $\{=, \supseteq, \subseteq, \cap\}\rangle$

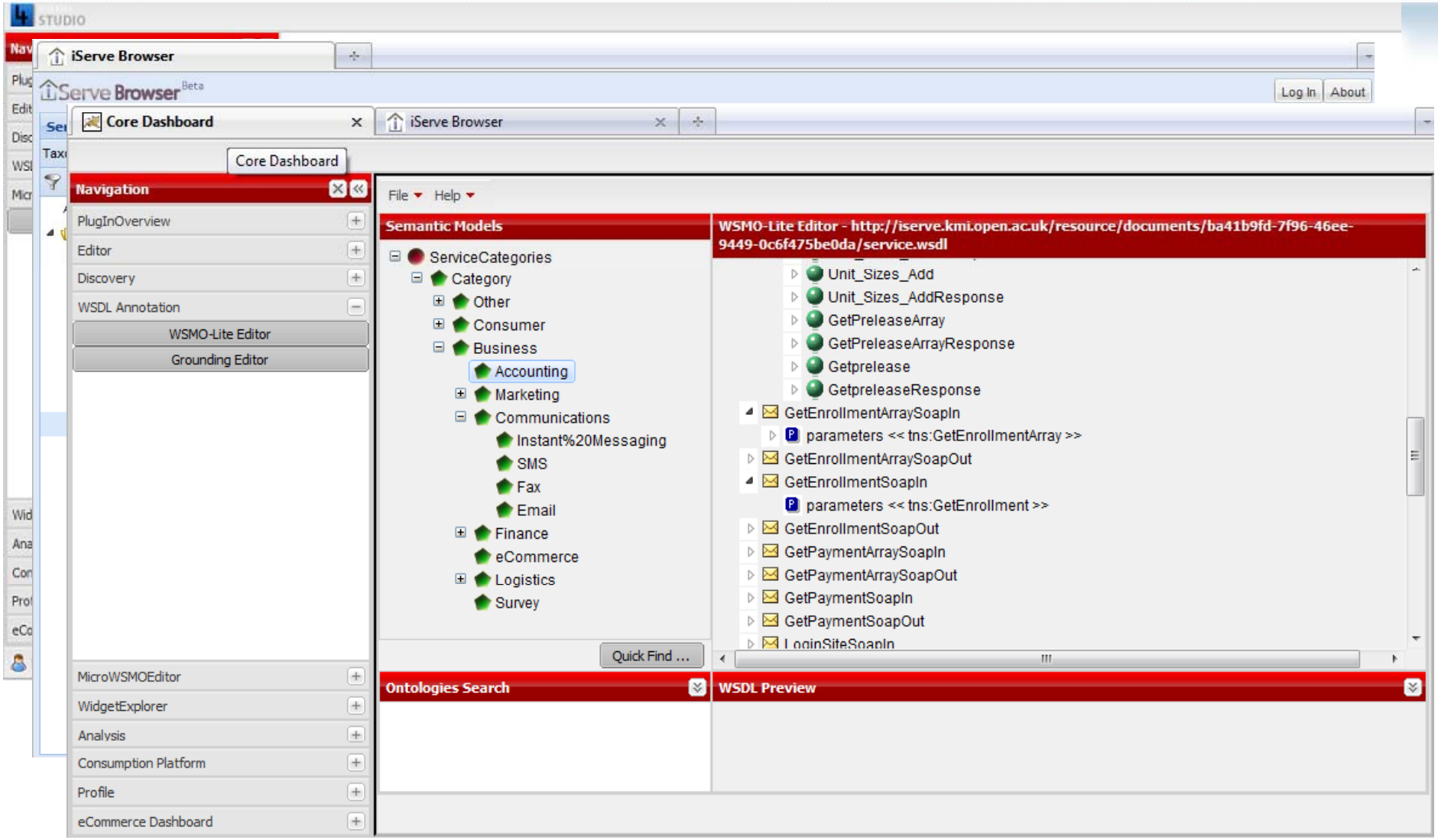
Algorithm: Web service ranking with WSMO-Lite nonfunctional properties
Inputs: set S of known services annotated with nonfunctional properties,
set $R_\mathcal{G}$ of user preferences, and $D_\mathcal{G}$ with the input data.
Output: a set of Web services along with their ranking values

Algorithm: forward-search Web service composition for WSMO-Lite
Inputs: set S of known services annotated with capabilities,
$\psi^{\mathit{eff}}(x_1, \ldots, x_n)$ defining the goal models,
$D_\mathcal{G}$ with the initial constants.
Output: a list of Web services to be applied in a sequential composition

...

WSMO-Lite tools

(Demo: SOA4All Studio http://www.soa4all.eu/)



Outline

- Environmental Models
  - Examples in Hazard Risk Assessment: Oil Spill and Landslide
  - Models as (Web) Services and ENVISION
- Types of Web services
  - WSDL and SAWSDL
  - OGC Web services
  - RESTful Web services
- Lightweight semantic annotations for Web services: WSMO-Lite
  - Types of service semantics and WSMO-Lite service description model
  - Attaching semantic annotations to WSDL and RESTful Web services
- Summary and conclusions

Summary and conclusions

- Environmental models are important for decision making
  - Models' current limitations hinder their reuse and interoperability
  - A platform of interoperating models is needed
- ENVISION – a platform for exposing environmental models as (Web) services
- Semantic annotations for Web services: key to automate service usage
- Core types of services on the Web: WSDL, OGC, and RESTful
  - Number of public services is increasing
  - The majority of services have no semantic annotations
- WSMO-Lite
  - Simplifies the creation of semantic descriptions of Web services
  - Defines an annotation mechanism for WSDL and RESTful services
  - Provides the bridge between WSDL, RESTful services, and (existing) domain-specific ontologies such as classification schemas, domain ontology models
  - Tools and automation algorithms available for WSMO-Lite (implementations ongoing)
- Open issues
  - Creation of semantic annotations, scalability of automation algorithms
  - User involvement in the service usage automation process
  - Attaching WSMO-Lite annotations to OGC Web service
  - …

Thank you!

Q&A

Acknowledgements

- This work is partly funded by the ENVISION project http://www.envision-project.eu/
- The WSMO-Lite semantic annotation approach to SWS has been developed in collaboration with Jacek Kopecky, Tomas Vitvar, and others, partly in the context of the Conceptual Model for Services Working Group, the SOA4All project, and other semantic Web services projects