Semantic Web Services
SS 2016

Light-weight Annotations

Anna Fensel

30.05.2016

Where are we?

STI · INNSBRUCK

#	Title
1	Introduction
2	Web Science + TourPack project (separate slideset)
3	Service Science
4	Web services
5	Web2.0 services
6	Semantic Web + ONLIM APIs (separate slideset)
7	Semantic Web Service Stack (WSMO, WSML, WSMX)
8	OWL-S and the others
9	Semantic Services as a Part of the Future Internet and Big Data Technology
10	**Lightweight Annotations**
11	Linked Services
12	Applications
13	Mobile Services

Outline

STI · INNSBRUCK

- Motivation
- Technical solution
  - Overview of light-weight SWS annotations
  - WSDL
  - hRESTS
  - SAWSDL
  - MicroWSMO
  - WSMO-Lite
  - Schema.org Action
- Illustration by a larger example
- Extensions
- Summary
- References

3

STI · INNSBRUCK

Motivation

4

Motivation

STI · INNSBRUCK

- The vision of Semantic Web Services (SWS)
  - Automating typical Web Service usage tasks.
  - Resolving heterogeneities issues.
  - Fostering scalability.
- Existing SWS approaches
  - WSMO, OWL-S, SWSF
  - Addressing the aforementioned problems.

, but they are considered as

  - Heavyweight solutions.
  - Introducing new languages founded on a expressive formalisms.
  - Promoting the top-down modeling approach (semantics-first).
  - Grounded usually in WSDL-based services.

5

Motivation

STI · INNSBRUCK

- Need for lightweight service ontologies.
- Directly built on top of the newest W3C standards
  - RDF(S), OWL, SAWSDL
- Promoting the bottom-up modeling approach
  - Augmentation of existing service specifications with semantic descriptions.
- Covering the other grounding approaches (i.e., REST)
  - WSDL-based services
    - 23757 services and 8094 providers according to Service Finder[1]
    - Many of them are used for the intra enterprise integration
  - RESTful-based services
    - 68% RESTful services vs. 19% SOAP services[2] ?!
    - Currently the dominant approach to offer services over the Web

[1] Statistics retrieved from the Service Finder demo on Dec 17th, 2009 @ http://demo.service-finder.eu/statistics
[2] Statistics retrieved from the Programmable Web on Dec 17th, 2009 @ http://www.programmableweb.com/apis

6

Motivation

STI · INNSBRUCK

1. SAWSDL
   - Semantic Annotations for WSDL and XML Schema.
   - New W3C standard.
   - Building on top of WSDL, URIs.
   - Assuming RDF, OWL.
2. Lightweight SWS approaches
   - WSMO modularized.
   - Key pieces of service semantics.
   - Simple semantic representation.

7

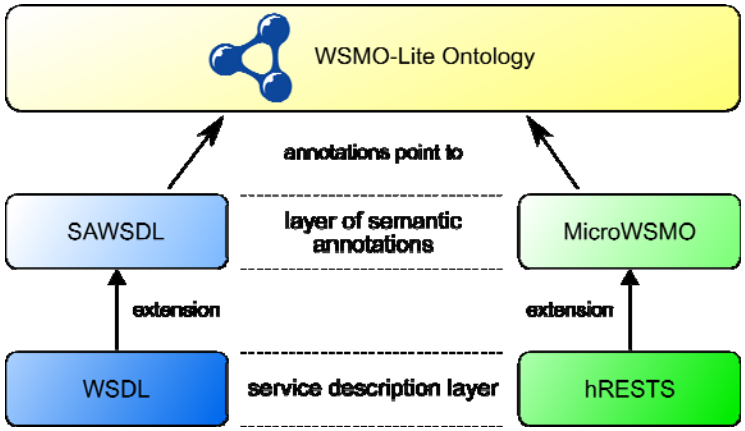
STI · INNSBRUCK

Technical Solution

8

Technical Solution
Overview

STI · INNSBRUCK



9

Technical Solution
Overview

STI · INNSBRUCK

- WSMO-Lite
  - Lightweight semantic descriptions for services on the Web
  - Inspired by WSMO (Lecture #7)
  - Focusing on a subset of WSMO
- From WSDL over SAWSDL to WSMO-Lite
  - Web Service Description Language (WSDL) is the standard to describe Web service interfaces (Lecture #4)
  - Semantically Annotated WSDL (SAWSDL) introduces a set of extension attributes which can be associated to the elements of a WSDL definition
  - Attributes can point to the WSMO-Lite elements

10

Technical Solution
Overview

STI · INNSBRUCK

- From hRESTS over MicroWSMO to WSMO-Lite
  - HTML for RESTful Service Descriptions (hRESTS) is a microformat used to identify basic service constructs (service definition, operations, inputs, outputs) inside HTML page which describes RESTful service.
  - MicroWSMO extends hRESTS with the elements (model, lowering, lifting) which can be associated with the elements of a hRESTS definition
  - Elements can point to the WSMO-Lite elements
  - MicroWSMO and hRESTS are lightweight equivalents for SAWSDL and WSDL for RESTful services.

11

STI · INNSBRUCK

Technical Solution
Web Service Description Language (WSDL)

12

Web Service Description Language
Overview

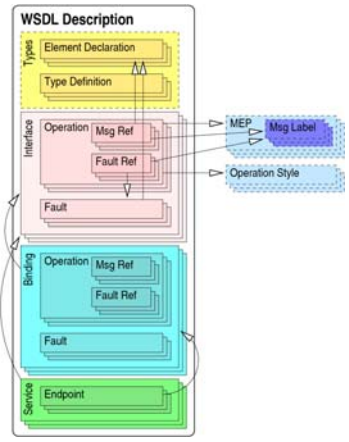
STI · INNSBRUCK

- Web Service Description Language
  - Interface Definition Language (IDL) for Web Services
- Current version: 2.0 @ W3C
  - Version 1.1 still in widespread use
- Interface – reusable, abstract
  - Operations with Message Exchange Patterns (MEPs)
    - In-Out, In-Only, Out-Only, Out-In
- Binding – reusable, concrete
- Service implements an interface
  - Endpoints use bindings

13

Web Service Description Language
WSDL Parts

STI · INNSBRUCK



- Types
  - Defining types and structures used during message exchanges
- Interface
  - Abstract view on what the operation does
  - Contains descriptions of operations and faults
  - References to the type definitions and element declarations
- Binding
  - Defines how an interface is accessed over a network
  - Specifies network details for each operation and fault
- Service
  - Main WSDL component
  - Can have a number of network accessible addresses – endpoints
  - Service implements interface

14

Web Service Description Language
Example

STI · INNSBRUCK

```
<wsdl:description  targetNamespace="http://www.w3.org/2002/ws/sawsdl/spec/wsdl/order#"
     xmlns="http://www.w3.org/2002/ws/sawsdl/spec/wsdl/order#"  xmlns:wsdl="http://www.w3.org/ns/wsdl"
     xmlns:xs="http://www.w3.org/2001/XMLSchema”>
  <wsdl:types>
    <xs:schema targetNamespace="http://www.w3.org/2002/ws/sawsdl/spec/wsdl/order#” elementFormDefault="qualified”>
      <xs:element name="OrderRequest”>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="customerNo" type="xs:integer" />
            <xs:element name="orderItem" type="item" minOccurs="1" maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      …
      <xs:element name="OrderResponse" type="confirmation" />
        <xs:simpleType name="confirmation”>
          <xs:restriction base="xs:string”>
            <xs:enumeration value="Confirmed" />
            <xs:enumeration value="Pending" />
            <xs:enumeration value="Rejected" />
          </xs:restriction>
        </xs:simpleType>
      </xs:schema>
    </wsdl:types>
  <wsdl:interface name="Order”>
    <wsdl:operation name="order" pattern="http://www.w3.org/ns/wsdl/in-out”>
      <wsdl:input element="OrderRequest" />
      <wsdl:output element="OrderResponse" />
    </wsdl:operation>
  </wsdl:interface>
  …
</wsdl:description>
```

15

STI · INNSBRUCK

Technical Solution
HTML for RESTful Service Descriptions (hRESTS)

16

hRESTS
RESTful Web service recap

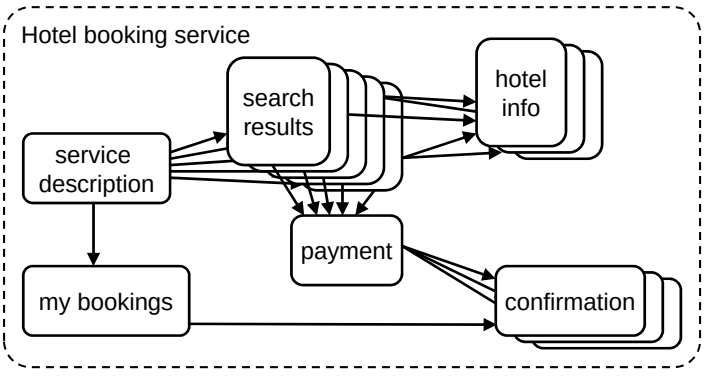
STI · INNSBRUCK

- A RESTful Web service is:
  - A set of Web resources
  - Interacting with GET/POST/PUT/DELETE
  - Interlinked
  - Datacentric, not functionalitycentric
  - Machineoriented
- More details can be found in Lecture #5

17

hRESTS
RESTful Web service recap example

STI · INNSBRUCK

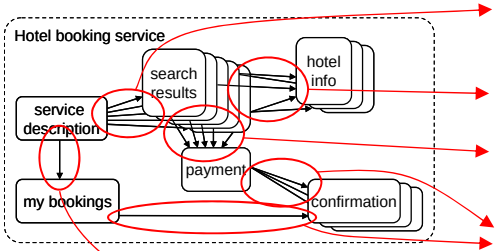


- A service is made up of interlinked resources.
- Links are not only <a href="…">, but also forms, such as the reservation form that leads from search results to the payment resource.

18

hRESTS
RESTful Web service recap example

STI · INNSBRUCK



search(date, city)
→ list of hotels & rates

getHotelDetails(hotel)

reserve(rate, creditCard)
→ confirmationID

getConfirmationDetails(confID)

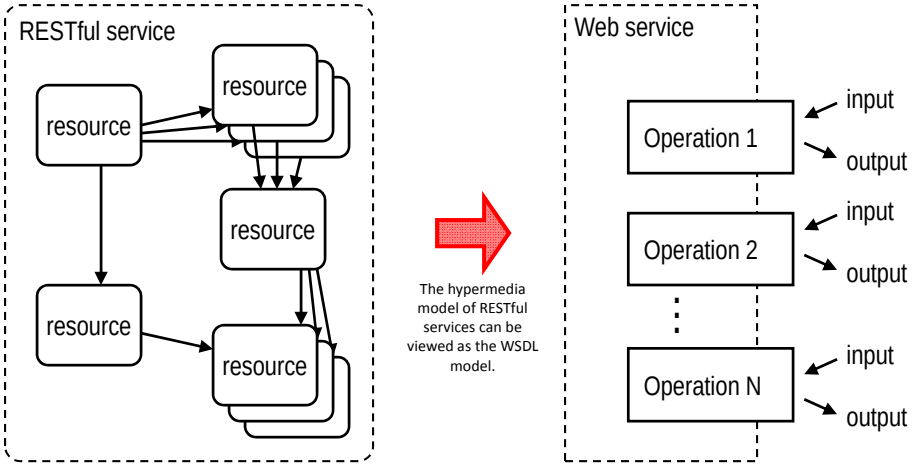
listMyBookings()
→ list of confirmationIDs

nouns vs. verbs

19

hRESTS
RESTful Web service recap example

STI · INNSBRUCK



20

hRESTS
hRESTS Overview

STI · INNSBRUCK

- "There's usually an HTML page"
  - There's no WSDL for Web apps
  - APIs described mostly in **text**
- Identifying machine-readable parts
  - Service, its operations
  - Resource address, HTTP method
  - Input/output data format
- **hRESTS** microformat

21

hRESTS
Definition

STI · INNSBRUCK

- HTML for RESTful Service Description
- Introduces the service model structure
  - `service` (+ `label`)
  - `operations` (+ `address`, `method`)
  - `input`, `output`
- Basis for extensions:
  - MicroWSMO adds semantic annotations

22

hRESTS
Example Description

STI · INNSBRUCK

Description of the ACME Hotels service:

The operation `getHotelDetails` is invoked using the method GET at `http://example.com/h/{id}`, with the ID of the particular hotel replacing the parameter `id`. It returns the hotel details in an `ex:hotelInformation` document.

hRESTS
Example HTML

STI · INNSBRUCK

```
<p>Description of the
  ACME Hotels service:</p>
<p>
  The operation <code>getHotelDetails</code> is
  invoked using the method GET
  at <code>http://example.com/h/{id}</code>,
  with the ID of the particular hotel replacing
  the parameter <code>id</code>.
  It returns the hotel details in an
  <code>ex:hotelInformation</code> document.
</p>
```

hRESTS
Example hRESTS

STI · INNSBRUCK

```
<div class="service" id="svc">
  <p>Description of the
    <span class="label">ACME Hotels</span> service:</p>
  <div class="operation" id="op1">
    <p>
    The operation <code class="label">getHotelDetails</code> is
    invoked using the method <span class="method">GET</span>
    at <code class="address">http://example.com/h/{id}</code>,
    with <span class="input">the ID of the particular hotel replacing
       the parameter <code>id</code>.</span>
    It returns <span class="output">the hotel details in an
       <code>ex:hotelInformation</code> document.</span>
  </p></div>
</div>
```

25

STI · INNSBRUCK

Technical Solution
Semantically Annotated WSDL (SAWSDL)

26

SAWSDL
Overview

STI · INNSBRUCK

- How to add semantic annotations to various parts of a WSDL document
  - Input and output message structures, interfaces and operations.
- Relying to simple extension attributes
  - Compliant to WSDL v2.0 and v1.1, and XML Schema
- Annotations can be used for various purposes:
  - WSDL interfaces and operations with categorization information used to advertise Web services
  - XML Schema types to foster discovery and composition
  - Specifying the data mapping of XML Schema to/from an ontology used during invocation (possible mediation)
- Independent on the ontology expression language and mapping languages

27

SAWSDL
Extension attributes

STI · INNSBRUCK

- Extension attributes of SAWSDL are:
  - *modelReference*
    - Pointers to a concept in some semantic model.
    - Annotates XML Schema type definitions, element declarations, and attribute declarations, WSDL interfaces, operations, and faults.
  - *liftingSchemaMapping*
    - Added to XML Schema element declarations and type definitions for specifying lifting mappings between semantic data and XML.
  - *loweringSchemaMapping*
    - Added to XML Schema element declarations and type definitions for specifying lowering mappings between semantic data and XML.

28

SAWSDL
Extension attributes

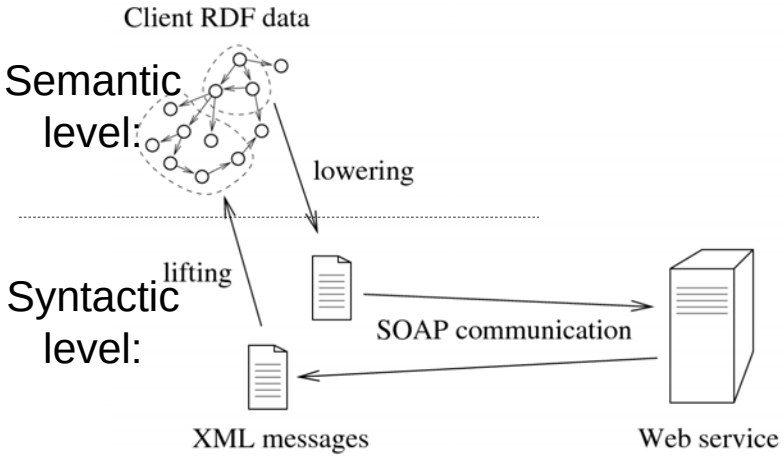
STI · INNSBRUCK

- Multiple semantic annotations can be associated with WSDL element.
- Schema mappings and a model references can contain multiple pointers.
- Multiple schema mappings are interpreted as alternatives
- Multiple model references all apply.
- SAWSDL does not specify any other relationship between them.

29

SAWSDL
Lifting/Lowering

STI · INNSBRUCK



30

SAWSDL
An example

STI · INNSBRUCK

```
<wsdl:description  targetNamespace="http://www.w3.org/2002/ws/sawsdl/spec/wsdl/order#"
      xmlns="http://www.w3.org/2002/ws/sawsdl/spec/wsdl/order#"  xmlns:wsdl="http://www.w3.org/ns/wsdl"
      xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sawsdl="http://www.w3.org/ns/sawsdl">
  <wsdl:types>
    <xs:schema targetNamespace="http://www.w3.org/2002/ws/sawsdl/spec/wsdl/order#" elementFormDefault="qualified">
      <xs:element name="OrderRequest"
        sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder#OrderRequest"
        sawsdl:loweringSchemaMapping="http://www.w3.org/2002/ws/sawsdl/spec/mapping/RDFOnt2Request.xml">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="customerNo" type="xs:integer" />
            <xs:element name="orderItem" type="item" minOccurs="1" maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      …
      <xs:element name="OrderResponse" type="confirmation" />
        <xs:simpleType name="confirmation"
          sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder#OrderConfirmation">
          <xs:restriction base="xs:string">
            <xs:enumeration value="Confirmed" />
            <xs:enumeration value="Pending" />
            <xs:enumeration value="Rejected" />
          </xs:restriction>
        </xs:simpleType>
      </xs:schema>
    </wsdl:types>
  <wsdl:interface name="Order"
    sawsdl:modelReference="http://example.org/categorization/products/electronics">
    <wsdl:operation name="order" pattern="http://www.w3.org/ns/wsdl/in-out"
      sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder#RequestPurchaseOrder">
      <wsdl:input element="OrderRequest" />
      <wsdl:output element="OrderResponse" />
    </wsdl:operation>
  </wsdl:interface>
</wsdl:description>
```

31

STI · INNSBRUCK

Technical Solution
MicroWSMO

32

MicroWSMO
Introduction

STI · INNSBRUCK

- hRESTS effectively creates an anologue of WSDL for RESTful services.
- hRESTS forms the basis for further extensions to annotate service descriptions for further processing.
- MicroWSMO, an extension of hRESTS adds semantics annotations through adopting the SAWSDL extensions:
  - `model` indicates that a link is a model reference, and
  - `lifting, lowering` links to the respective data transformations.
- The model attribute can point to the WSMO-Lite descriptions.

33

MicroWSMO
Example

STI · INNSBRUCK

```
<div class="service" id="svc">
  <p><span class="label">ACME Hotels</span> is a
    <a rel="model" href="&ex;AccommodationReservationService">
      hotel reservation</a> service.</p> …
  <div class="operation" id="op1"><p> …
    <span class="input">A particular hotel ID replaces the param
      <a rel="model" href="…/onto.owl#Hotel"><code>id</code></a>
      (<a rel="lowering" href="…/hotelID.xsparql">lowering</a>).
    </span>. …
  </p></div>
</div>
```

34

MicroWSMO
Semantics Implied in Web

STI · INNSBRUCK

- Hypermedia → behavioral semantics
  - Links become available through interaction
- Uniform interface → behavioral semantics
  - GET, PUT, DELETE have known effects
  - GET is *safe,* PUT and DELETE *idempotent*
  - POST has no implied semantics
- Self-description → information model
  - Operation output data can specify what it is
    - GRDDL, other semantic annotations

GRDDL is a mechanism for Gleaning Resource Descriptions from Dialects of Languages. This GRDDL specification introduces markup based on existing standards for declaring that an XML document includes data compatible with the Resource Description Framework (RDF) and for linking to algorithms (typically represented in XSLT), for extracting this data from the document.

35

STI · INNSBRUCK

Technical Solution
WSMO-Lite

36

WSMO-Lite
Introduction

STI · INNSBRUCK

- Lightweight semantic descriptions for services on the Web
  - Filling the SAWSDL annotations with concrete semantic service descriptions.
- Working draft of CMS Working Group, v0.3
- Inspired by WSMO ontology
  - Focusing on a subset.
  - Defining gradual extension of SAWSDL.
- Addresses following requirements:
  - Identifies types and simple vocabulary for semantic description of services and languages used to express descriptions.
  - Defines an annotation mechanism for WSDL
  - Provides a bridge between WSDL, SAWSDL and existing domain-specific ontologies such as classification schemas, etc.

37

WSMO-Lite
Types of Service Semantics

STI · INNSBRUCK

- Functional 

  - What the service does
- Behavioral 

  - How the client talks to the service
- Information model 

  - For handling data
  - Incl. lifting/lowering
- Nonfunctional 

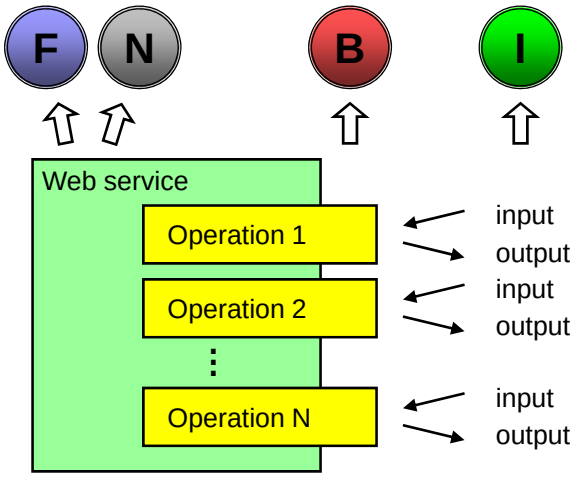
  - Policies, QoS, price, location etc.

38

WSMO-Lite
Semantics in Service Model

STI · INNSBRUCK

- Using SAWSDL *modelReferences*, the four kinds of semantics are attached to the service model:
  - functional (F) and nonfunctional (N) properties to the service,
  - behavioral (B)properties to the operations, and
  - information model (I) properties to the input and output messages (and the fault messages as well).



39

WSMO-Lite
Functional Semantics

STI · INNSBRUCK

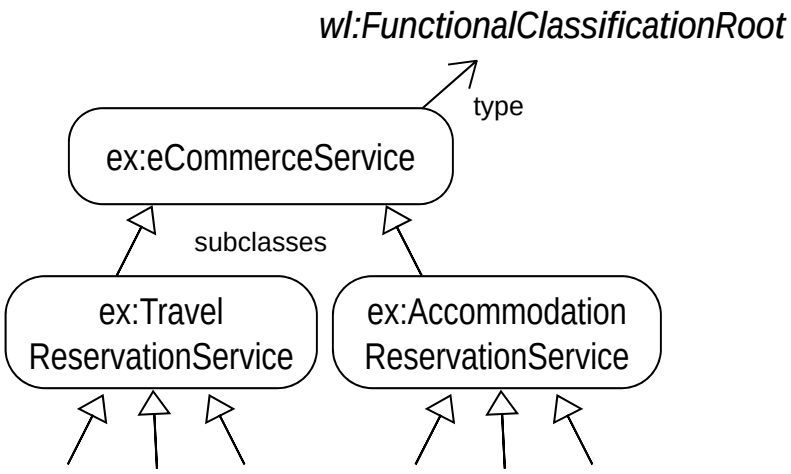
- For service discovery, composition

- *Category*
  - Functionality categorization
  - E.g. eCl@ss, UDDI
  - Or tagging, folksonomies
- *Capability*
  - Precondition, Effect
  - Using WSML rule languages

40

WSMO-Lite
Category Example

STI · INNSBRUCK



41

WSMO-Lite
Category Example (RDF)

STI · INNSBRUCK

ex:eCommerceService
rdf:type *wl:FunctionalClassificationRoot* .
ex:TravelReservationService
rdfs:subClassOf ex:eCommerceService .
ex:AccommodationReservationService
rdfs:subClassOf ex:eCommerceService .
…

42

WSMO-Lite
Capability Example

STI · INNSBRUCK

```
ex:RomaHotelReservationPrecondition
    rdf:type  wl:Condition ;
    rdf:value """
        ?request
           [ numberOfGuests  hasValue  ?guests
             and  city  hasValue  ?city ]
           memberOf  ReservationData
  and ?guests <= 10
  and ?city = 'Roma'
                              """^^wsml:AxiomLiteral .
```

43

WSMO-Lite
Nonfunctional Semantics

STI · INNSBRUCK

- 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 .
```

44

WSMO-Lite
Behavioral Semantics

STI · INNSBRUCK

- 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
- Example functional category for operations:
  WebArch interaction safety

45

WSMO-Lite
Information Semantics

STI · INNSBRUCK

- For invocation, composition, data mediation
- Not constrained, any ontologies
- Refer to course *Semantic Web*

46

WSMO-Lite
The ontology expressed in RDFS

STI · INNSBRUCK

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#> .

wsl:Ontology a rdfs:Class;
   rdfs:subClassOf owl:Ontology.
wsl:FunctionalClassificationRoot rdfs:subClassOf rdfs:Class.
wsl:NonFunctionalParameter a rdfs:Class.
wsl:Condition a rdfs:Class.
wsl:Effect a rdfs:Class.
```

WSMO-Lite service ontology in RDFS, serialized in Notation 3.

47

WSMO-Lite
The ontology expressed in RDFS - Explanation

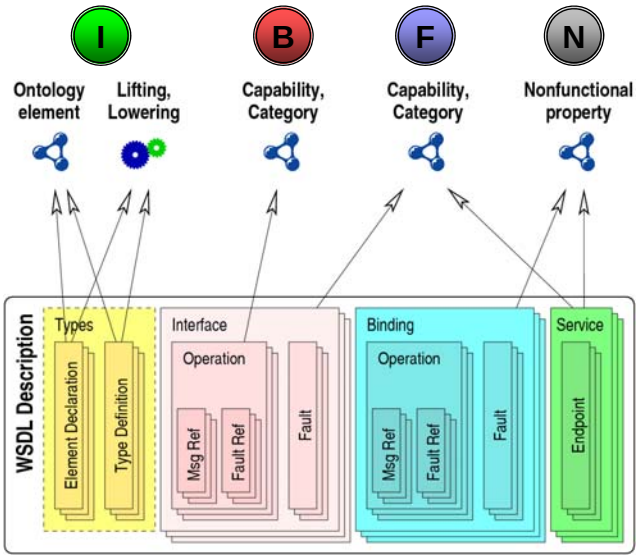
STI · INNSBRUCK

- Semantics of the WSMO-Lite elements:
  - wsl:Ontology
    - Defines a container for a collection of assertions about the information model of a service.
    - wsl:Ontology is a subclass of owl:Ontology limited to such ontologies that may serve as information models.
    - OWL ontology meta-data such as comments, version control and inclusion of other ontologies, are also allowed on wsl:Ontology.
  - wsl:FunctionalClassificationRoot
    - marks a class that is a root of a classification which also includes all the RDFS subclasses of the root class (the actual functional categories).
    - A classification (taxonomy) of service functionalities can be used for functional description of a service.
  - wsl:NonFunctionalParameter
    - specifies a place holder for a concrete domain- specific nonfunctional property.
  - wsl:Condition and wsl:Effect
    - together form a capability in a functional service description. Both are expected to use a concrete logical language to describe the logical expressions for conditions and effects.

48

WSMO-Lite
WSMO-Lite in SAWSDL

STI · INNSBRUCK



49

WSMO-Lite
WSMO-Lite in SAWSDL

STI · INNSBRUCK

- WSDL has more detail than our simplified service model, therefore there are more ways of attaching semantics.
- Information semantics can be anywhere in the schema inside the types section.
- Functional semantics can be either on the interface (general, reusable) or on the service (specific), and
- Nonfunctional parameters can be on a binding (general, reusable) or on the service (specific).

50

WSMO-Lite
WSMO-Lite in SAWSDL - Example

STI · INNSBRUCK

```
<wsdl:description>
  <wsdl:types>  <xs:schema>
    <xs:element name="ReservationRequest"
        sawsdl:modelReference="&ex;Reservation"
        sawsdl:loweringSchemaMapping="&ex;ResMapping.xsparql" … />
  </xs:schema>  </wsdl:types>
  <wsdl:interface  name="HotelReservations"
        sawsdl:modelReference=
          "&ex;AccommodationReservationService">
    <wsdl:operation name="searchForRooms"
        sawsdl:modelReference="&wsdlx;SafeInteraction">
      …
    </wsdl:operation>
    …
  </wsdl:interface>
  <wsdl:service  name="RomaHotels"  interface="HotelReservations"
        sawsdl:modelReference="&ex;RomaHotelReservationPrecondition
                               &ex;ReservationFee"   … />
</wsdl:description>
```

I
I
F
B
F
N

WSMO-Lite
WSMO-Lite in MicroWSMO

STI · INNSBRUCK

```
<div class="service" id="svc">
  <p><span class="label">ACME Hotels</span> is a
    <a rel="model" href="&ex;AccommodationReservationService">
      hotel reservation</a> service.</p> …
  <div class="operation" id="op1"><p> …
    <span class="input">A particular hotel ID replaces the param
      <a rel="model" href="…/onto.owl#Hotel"><code>id</code></a>
      (<a rel="lowering" href="…/hotelID.xsparql">lowering</a>).
    </span>. …
  </p></div>
</div>
```

F
B
I

Technical Solution

schema.org Action

What is Schema.org?

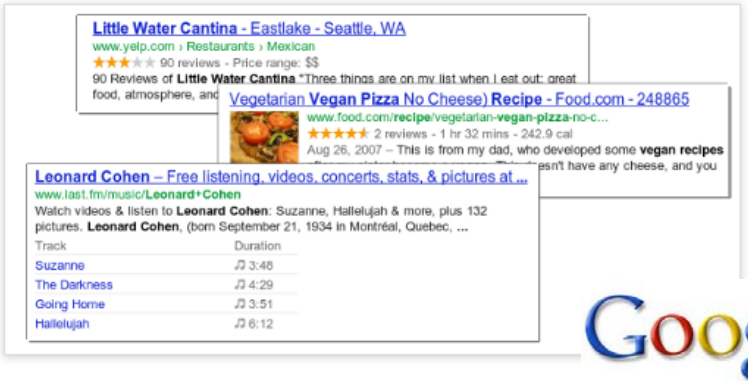
- **Schema.org** provides a collection of shared vocabularies.
- Launched in June 2011 by Bing, Goolge and Yahoo
- Yandex joins in November 2011
- Purpose:

Create a common set of schemas for webmasters to mark-up with structured data their websites.

Motivation: What for?

STI · INNSBRUCK

- Lead to the generation of rich snippets in search engine results more attractive for the users



55

Motivation: What for?

STI · INNSBRUCK

- Webmasters can use schema.org to mark up their web pages (creating enriched snippets) in a way that is recognized by major search engines.
- The enriched snippets enable search engines to understand the information on web pages that results in richer and more attractive search results for the users. This is done in order make it easier for users to find relevant and right information on the web.

schema.org

56

Data Model

STI · INNSBRUCK

- Derived from RDFS
- Based on:
  - Set of Types (classes)
    - Organized in a hierarchy
    - Each type (class) might be a sub-class of several types (classes)
  - Properties
    - Each property can have 1 or more items as domains
    - Each property can have 1 or more items as range

57

Data Model

STI · INNSBRUCK

- Canonical representation in RDFa
  - http://schema.org/docs/schema_org_rdfa.html
- Schema.org can be extended
- Schema.org properties can be used in other contexts
- The type hierarchy presented in Schema.org is not intended to be a 'global ontology' of the world.

58

Schema.org vocabularies

STI · INNSBRUCK

- Most popular vocabularies relates to…
  - CreativeWork
    - Book, Movie, Recipe, TVSeries, Review…
  - Embedded non-text objects: AudioObject, ImageObject,
  - Event
    - Food Event, Dance Event, Festival, SportsEvent…
  - Organization
  - Person
  - Place, Local Business, Hotel, Restaurant ...
  - Product, Offer
- All types of vocabularies can be found in : **http://schema.org/docs/full.html**

59

Schema.org vocabularies

STI · INNSBRUCK

- Support the following DataTypes
  - Boolean
    - False
    - True
  - Date
  - Date Time
  - Number
    - Float
    - Integer
  - Text
    - URL
  - Time

60

Schema.org vocabularies

STI · INNSBRUCK

- For each item, Schema.org describes:
  - A list of own properties, range (datatype or item) and description
  - A list of inherited properties
  - A list of properties for which instances of the selected item may appear as values
  - A list of subclasses (more specific types)
  - Example of usage

61

Schema.org vocabularies

STI · INNSBRUCK

Thing > Organization > LocalBusiness

A particular physical business or branch of an organization. Examples of LocalBusiness include a restaurant, a particular branch of a restaurant chain, a branch of a bank, a medical practice, a club, a bowling alley, etc.

Property	Expected Type	Description
Properties from LocalBusiness		
branchOf	Organization	The larger organization that this local business is a branch of, if any.
currenciesAccepted	Text	The currency accepted (in ISO 4217 currency format).
openingHours	Duration	The opening hours for a business. Opening hours can be specified as a weekly time range, starting with days, then times per day. Multiple days can be listed with commas ',' separating each day. Day or time ranges are specified using a hyphen '-'. - Days are specified using the following two-letter combinations: Mo, Tu, We, Th, Fr, Sa, Su. - Times are specified using 24:00 time. For example, 3pm is specified as 15:00. - Here is an example: <time itemprop="openingHours" datetime="Tu,Th 16:00-20:00">Tuesdays and Thursdays 4-8pm</time>. - If a business is open 7 days a week, then it can be specified as <time itemprop="openingHours" datetime="Mo-Su">Monday through Sunday, all day</time>.
paymentAccepted	Text	Cash, credit card, etc.
priceRange	Text	The price range of the business, for example $$$.
Properties from Organization		
address	PostalAddress	Physical address of the item.
aggregateRating	AggregateRating	The overall rating, based on a collection of reviews or ratings, of the item.
brand	Brand or Organization	The brand(s) associated with a product or service, or the brand(s) maintained by an organization or business person.
contactPoint	ContactPoint	A contact point for a person or organization. Supersedes contactPoints.
department	Organization	A relationship between an organization and a department of that organization, also described as an organization (allowing different urls, logos, opening hours). For example: a store with a pharmacy, or a bakery with a cafe.
dissolutionDate	Date	The date that this organization was dissolved.

62

How to mark-up with schema.org?

STI · INNSBRUCK

- Schema.org can be used to enrich the web sites with the following formats:
  - Microdata (most popular)
    - Tags introduced within HTML 5
    - Based on Item descriptions
    - Itemscope, Itemtype, Itemprop
  - RDFa
  - JSON-LD

63

Example I

STI · INNSBRUCK

Vocabulary – schema.org

- **Example***:
  - Imagine you have a page about the movie Avatar—a page with a link to a movie trailer, information about the director, and so on. Your HTML code might look something like this:

```
<div>
        <h1>Avatar</h1>
        <span>Director: James Cameron (born August 16, 1954)</span>
         <span>Science fiction</span>
        <a href="../movies/avatar-theatrical-trailer.html">Trailer</a>
 </div>
```

* http://schema.org/docs/gs.html

64

Example I

STI · INNSBRUCK

- Thing > Creative Work > Movie
  - Particular properties

Thing > CreativeWork > Movie

A movie.

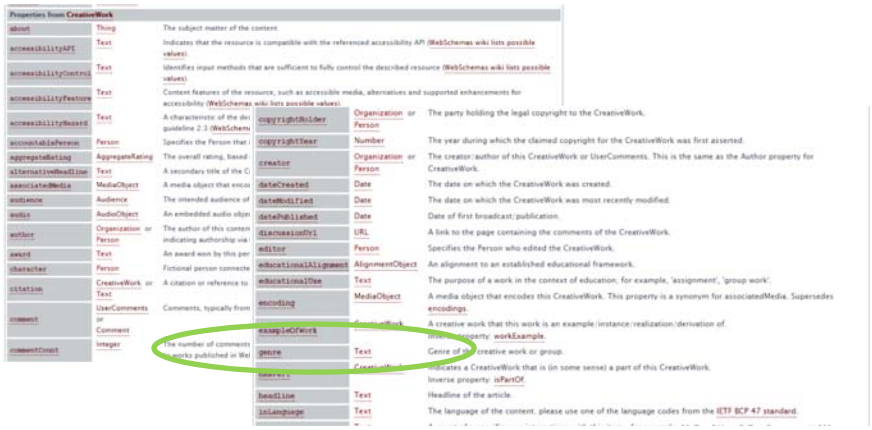
Property	Expected Type	Description
Properties from Movie		
actor	Person	An actor, e.g. in tv, radio, movie, video games etc. Actors can be associated with individual items or with a series, episode, clip. Supersedes actors.
director	Person	A director of e.g. tv, radio, movie, video games etc. content. Directors can be associated with individual items or with a series, episode, clip. Supersedes directors.
duration	Duration	The duration of the item (movie, audio recording, event, etc.) in ISO 8601 date format.
musicBy	MusicGroup or Person	The composer of the soundtrack.
productionCompany	Organization	The production company or studio responsible for the item e.g. series, video game, episode etc.
trailer	VideoObject	The trailer of a movie or tv/radio series, season, episode, etc.

65

Example I

STI · INNSBRUCK

- Inherited properties (from Creative Work and Thing)



66

Example I

STI · INNSBRUCK

- Inherited properties (from Creative Work and Thing)

Properties from Thing		
additionalType	URL	An additional type for the item, typically used for adding more specific types from external vocabularies in microdata syntax. This is a relationship between something and a class that the thing is in. In RDFa syntax, it is better to use the native RDFa syntax - the 'typeof' attribute - for multiple types. Schema.org tools may have only weaker understanding of extra types, in particular those defined externally.
alternateName	Text	An alias for the item.
description	Text	A short description of the item.
image	URL or ImageObject	An image of the item. This can be a URL or a fully described ImageObject.
name	Text	The name of the item.
potentialAction	Action	Indicates a potential Action, which describes an idealized action in which this thing would play an 'object' role.
sameAs	URL	URL of a reference Web page that unambiguously indicates the item's identity. E.g. the URL of the item's Wikipedia page, Freebase page, or official website.
url	URL	URL of the item.

Example I

STI · INNSBRUCK

Vocabulary – schema.org

- Example with microdata*:

```
<div itemscope itemtype ="http://schema.org/Movie">
        <h1 itemprop="name"&g;Avatar</h1>
        <div itemprop="director" itemscope itemtype="http://schema.org/Person">
                Director:
                <span itemprop="name">James Cameron</span>
                (born
                <span itemprop="birthDate">August 16, 1954)</span>
         </div>
        <span itemprop="genre">Science fiction</span>
        <a href="../movies/avatar-theatrical-trailer.html" itemprop="trailer">Trailer</a>
</div>
```

* http://sche

Example II

STI · INNSBRUCK

- Hotel information

69

Schema.org Actions

STI · INNSBRUCK

- From Nouns to Verbs: Actions
  - Object → potential actions
  - Constraints on actions
  - E.g., ThorMovie → Stream, Buy, …
- Introducing time: Roles
  - E.g., Joe Montana played for the SF 49ers from 1979 to 1992 in the position QuarterBack

Adapted from R.V. Guha, Google.

70

Action - Definition

STI · INNSBRUCK

Action

Thing > Action

An action performed by a direct agent and indirect participants upon a direct object. Optionally happens at a location with the help of an inanimate instrument. The execution of the action may produce a result. Specific action sub-type documentation specifies the exact expectation of each argument/role.

See also blog post and Actions overview document.

Usage: Between 100 and 1000 domains

[more...]

Property	Expected Type	Description
Properties from Action		
actionStatus	ActionStatusType	Indicates the current disposition of the Action.
agent	Organization or Person	The direct performer or driver of the action (animate or inanimate). e.g. *John* wrote a book.
endTime	DateTime	The endTime of something. For a reserved event or service (e.g. FoodEstablishmentReservation), the time that it is expected to end. For actions that span a period of time, when the action was performed. e.g. John wrote a book from January to *December*. Note that Event uses startDate/endDate instead of startTime/endTime, even when describing dates with times. This situation may be clarified in future revisions.
error	Thing	For failed actions, more information on the cause of the failure.
instrument	Thing	The object that helped the agent perform the action. e.g. John wrote a book with *a pen*.

More specific types of Action exist: AchieveAction, AccessAction, ConsumeAction…

71

Example

STI · INNSBRUCK

```
<!--  John created an exercise plan with Steve. -->
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "CreateAction",
  "agent": {
    "@type": "Person",
    "name": "John"
  },
  "result": {
    "@type": "ExercisePlan",
    "name": "John's weight loss plan"
  },
  "participant": {
    "@type": "Person",
    "name": "Steve"
  }
}
</script>
```

72

Related Resources

STI · INNSBRUCK

- TopBraidComposer
  - Schema.org vocabularies already included
  - http://www.topquadrant.com/tools/modeling-topbraid-composer-standard-edition/
- GetSchema.org
  - http://getschema.org/index.php?title=Main_Page
- Schema 101: how to implement schema.org
  - http://www.searchenginejournal.com/schema-101-how-to-implement-schema-org-markups-to-improve-seo-results/58210/
- Hydra - an effort to simplify the development of interoperable, hypermedia-driven Web APIs: https://www.w3.org/ns/hydra/
  - Currently active working group, work in progress!

73

STI · INNSBRUCK

Illustration by a larger example

74

Illustration by a Larger Example
Service Description

STI · INNSBRUCK

- A telecommunication company wants to enter a highly competitive video on demand market.
- The market is formed around the service provisioning platform based on WSMO-Lite service description
- The company has a WSDL service offering video on demand subscription functionality
  - The service formulates basic concepts in the domain such as Customer, Service, Network Connection, etc. in the form of XML Schema.
  - The service provides basic subscription operation.
- The service can processed a subscription request only if a customer already has a network connection with the appropriate bandwidth.
- The company also wants to associate its pricing schema related to the service consumption.

75

Illustration by a Larger Example
WSMO-Lite Description

STI · INNSBRUCK

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#> .
@prefix ex: <http://example.org/onto#> .
@prefix xs: <http://www.w3.org/2001/XMLSchema#> .
@prefix wsml: <http://www.wsmo.org/wsml/wsml-syntax#> .

 <> a wsl:Ontology.ex:Customer a rdfs:Class .
ex:hasService a rdf:Property ;
     rdfs:domain ex:Customer ;
     rdfs:range ex:Service .
ex:Service a rdfs:Class .
ex:hasConnection a rdf:Property ;
     rdfs:domain ex:Customer ;
     rdfs:range ex:NetworkConnection .
ex:NetworkConnection a rdfs:Class .
ex:providesBandwidth a rdf:Property ;
     rdfs:domain ex:NetworkConnection ;
     rdfs:range xs:integer .
ex:VideoOnDemandService rdfs:subClassOf ex:Service .

ex:VideoOnDemandSubscriptionPrecondition a wsl:Condition ;
     rdf:value """ ?customer[ex#hasConnection hasValue ?connection]memberOf ex#Customer and
     ?connection[ex#providesBandwidth hasValue ?y]memberOf ex#NetworkConnection and ?y >
     1000"""^^wsml:AxiomLiteral .
…
```

76

Illustration by a Larger Example
WSMO-Lite Description

STI · INNSBRUCK

```
…
ex:VideoOnDemandSubscriptionEffect a wsl:Effect ;
     rdf:value """ ?customer[ex#hasService hasValue ?service]memberOf ex#Customer and ?service memberOf
     VideoOnDemandSubscription"""^^wsml:AxiomLiteral .

wsml:AxiomLiteral a rdfs:Datatype .

ex:PriceSpecification rdfs:subClassOf wsl:NonFunctionalParameter .
ex:VideoOnDemandPrice a ex:PriceSpecification ;
     ex:pricePerChange "30"^^ex:euroAmount ;
     ex:installationPrice "49"^^ex:euroAmount .

ex:SubscriptionService a wsl:FunctionalClassificationRoot .
ex:VideoSubscriptionService rdfs:subClassOf ex:SubscriptionService .
ex:NewsSubscriptionService rdfs:subClassOf ex:SubscriptionService .
```

77

Illustration by a Larger Example
WSMO-Lite Description

STI · INNSBRUCK

- The description defines a simple ontology for a telecommunication service:
  - the capability for a concrete Video on Demand subscription service (the condition says that the customer must have a network connection with some minimal bandwidth, the effect says that the customer is subscribed to the service);
  - a nonfunctional property describing the pricing;
  - a simple functionality classification with three categories;
  - defines also the wsml:AxiomLiteral data type for WSML-Flight logical expressions so that a client can correctly process them according to the WSML specification.

78

Illustration by a Larger Example
SAWSDL Annotations - Fragments

STI · INNSBRUCK

```
<xs:element name="NetworkConnection" type="NetworkConnectionType"
    sawsdl:modelReference="http://example.org/onto#NetworkConnection"
    sawsdl:loweringSchemaMapping="http://example.org/NetCn.xslt"/>
```

Reference annotation linking to the class from the service information model ontology and transformation annotation referencing mapping used to lower semantic descriptions to an appropriate communication format.

```
<wsdl:interface name="NetworkSubscription"
      sawsdl:modelReference="http://example.org/onto#VideoSubscriptionService
                             http://example.org/onto#VideoOnDemandSubscriptionPrecondition
                             http://example.org/onto#VideoOnDemandSubscriptionEffect" >
```

Functional annotation including references to the capabilities and categories applied to the interface definition.

79

Illustration by a Larger Example
SAWSDL Annotations - Fragments

STI · INNSBRUCK

```
<wsdl:service name="ExampleCommLtd"
      interface="NetworkSubscription"
      sawsdl:modelReference="http://example.org/onto#VideoOnDemandPrice">
   <wsdl:endpoint ...
</wsdl:service>
```

Non functional model reference annotation associated with the concrete service.

80

STI · INNSBRUCK

Summary

81

Summary

STI · INNSBRUCK

- Existing SWS approaches are
  - Heavyweight solutions.
  - Introducing new languages founded on a expressive formalisms.
  - Promoting the top-down modeling approach (semantics-first).
  - Grounded usually in WSDL-based services.
- Light-weight annotations for SWS include
  - WSDL
    - Common and standard way to describe Web Service interface.
  - hRESTS
    - Microformat used to identify basic service constructs (service definition, operations, inputs, outputs) inside HTML page which describes RESTful service.
  - SAWSDL
    - Set of extension attributes which can be associated to the elements of a WSDL definition.
  - MicroWSMO
    - Extends hRESTS with the elements (model, lowering, lifting) which can be associated with the elements of a hRESTS definition.
  - WSMO-Lite
    - Lightweight ontology for semantic descriptions of services on the Web.
  - Schema.org Action
    - Lightweight SWS annotation usable with schema.org

82

Extensions

STI · INNSBRUCK

- Implementation
  - WSMO-Lite and MicroWSMO automation
    - SOA4All Studio
    - WSMX
- Tool support for service annotation
  - SOA4All Studio and others
- Standardization of WSMO-Lite through CMS Group
- Google potentially developing schema.org Action for scenarios such as direct booking from Google's user interfaces

83

References

STI · INNSBRUCK

- Mandatory reading:
  - J. Kopecky , T. Vitvar , C. Bournez , J. Farrell: SAWSDL: Semantic Annotations for WSDL and XML Schema, IEEE Internet Computing, IEEE CS Press, 2007, 11 (6).
  - J. Kopecky , K. Gomadam , T. Vitvar: hRESTS: an HTML Microformat for Describing RESTful Web Services, The 2008 IEEE/WIC/ACM International Conference on Web Intelligence (WI2008), IEEE CS Press, November, 2008, Sydney, Australia.
  - T. Vitvar , J. Kopecky , J. Viskova , D. Fensel: WSMO-Lite Annotations for Web Services , The 5th European Semantic Web Conference 2008, Springer, June, 2008, Tenerife, Spain.
  - SAWSDL: http://w3.org/TR/sawsdl
  - WSMO-Lite: http://cms-wg.sti2.org/TR/d11
  - hRESTS & MicroWSMO: http://cms-wg.sti2.org/TR/d12
  - Schema.org Action https://schema.org/Action
- Further reading:
  - Stavrakantonakis, I., Fensel, A., & Fensel, D. (2014, September). Matching Web Entities with Potential Actions. In *Proceedings of the Poster & Demo Paper track International Conference on Semantic Systems (SEMANTiCS'14), CEUR-WS* (Vol. 1224).
  - RDF: http://w3.org/TR/rdf-primer
  - RDFS: http://w3.org/TR/rdf-schema
- Wikipedia links:
  - REST: http://en.wikipedia.org/wiki/REST

84

Next Lecture

STI · INNSBRUCK

#	Title
1	Introduction
2	Web Science + TourPack project (separate slideset)
3	Service Science
4	Web services
5	Web2.0 services
6	Semantic Web + ONLIM APIs (separate slideset)
7	Semantic Web Service Stack (WSMO, WSML, WSMX)
8	OWL-S and the others
9	Semantic Services as a Part of the Future Internet and Big Data Technology
10	Lightweight Annotations
11	**Linked Services**
12	Applications
13	Mobile Services

85

Questions?

STI · INNSBRUCK

86