

# Semantic Web Services

## Assignment 6

Stefan Haselwanter  
Juliette Opdenplatz

May 14, 2018

### 1 Question: What is Process Mediation? (WSMO, WSML, WSMX stack)

What is process mediation? Why do we need it? Explain in the context of semantic web services. Examine the process mediation example in lecture slides and suggest another scenario.

### 2 Answer

Heterogeneity can and most certainly will be a problem in collaborative tasks that depend on a large number of different processes. There can already be problems if there are just two processes involved, that is, one process might depend on data, the other process generates. This is especially a problem when we think about semantic web service and how those services might interact with each other.

To overcome interoperability problems like this we need process mediation. Heterogeneities can appear in terms of data, ontology, process or protocol, thus mediators can be distinguished by respective levels. They might for example mediate either heterogeneous data sources or heterogeneous communication patterns. Therefore process mediation could be conceptualized as a *middleware* that coordinates the interaction between web services.

Of course there are frameworks that offer the ability to implement process mediation. The one of most importance is the WSMO approach:

**Web Service Modelling Ontology (WSMO)** Mediation is a first class citizen. WSMO offers specific means to semantically describe concrete mediation solutions and to directly refer to them when needed (e.g. from ontologies or Web services).

**Web Service Modeling Language (WSML)** A language for the WSMO approach.

1. RDFS and OWL have no support for web services, goals or mediators
2. OWL-S is not expressive enough to cover all aspects of Web Services

**Web Service Execution Environment (WSMX)** An execution environment to enable the WSMO approach.

1. Service offerings, required capabilities and exchanged data are semantically annotated
2. An environment to bridge service providers and requesters
3. Automation of tasks with reasoning

### 3 Example Scenario

So imagine an end-user travel web service that dynamically uses and aggregates other web services and consider that a client uses a different ontology than the web service description of the travel service.

So our example is about ontology mapping, which involves the creation of a set of rules that precisely define how terms from one ontology relate with terms from the other ontology. These rules are expressed using a mapping language which does not alter any ontology involved, because it is only about mapping definitions.

We consider the following example for illustrating terminology mismatches. The ontology used by the requestor contains the concept *station*:

```
concept station
startLocation impliesType boolean
destinationLocation impliesType boolean
name impliesType string
```

The ontology used by the provider contains the concept *route*:

```
concept route
from hasType (0 1) string
to hasType (0 1) string
```

There are two terminological mismatches. The first mismatch occurs because the attribute *startLocation* of the concept *station* corresponds to the attribute *from* of the *route* concept.

And the second mismatch occurs due to the fact that the attribute *destinationLocation* of the concept *station* corresponds to the attribute *to* of the *route* concept. In order to allow automated processing by ontology mapping, we need to create three mapping rules: one for stating the relation between the two concepts and two for imposing the mappings between their attributes.