



Integrated Metadata Support for Web Service Runtimes

Florian Rosenberg, *Philipp Leitner*, Anton Michlmayr
Schahram Dustdar

VitaLab, Distributed Systems Group
Vienna University of Technology

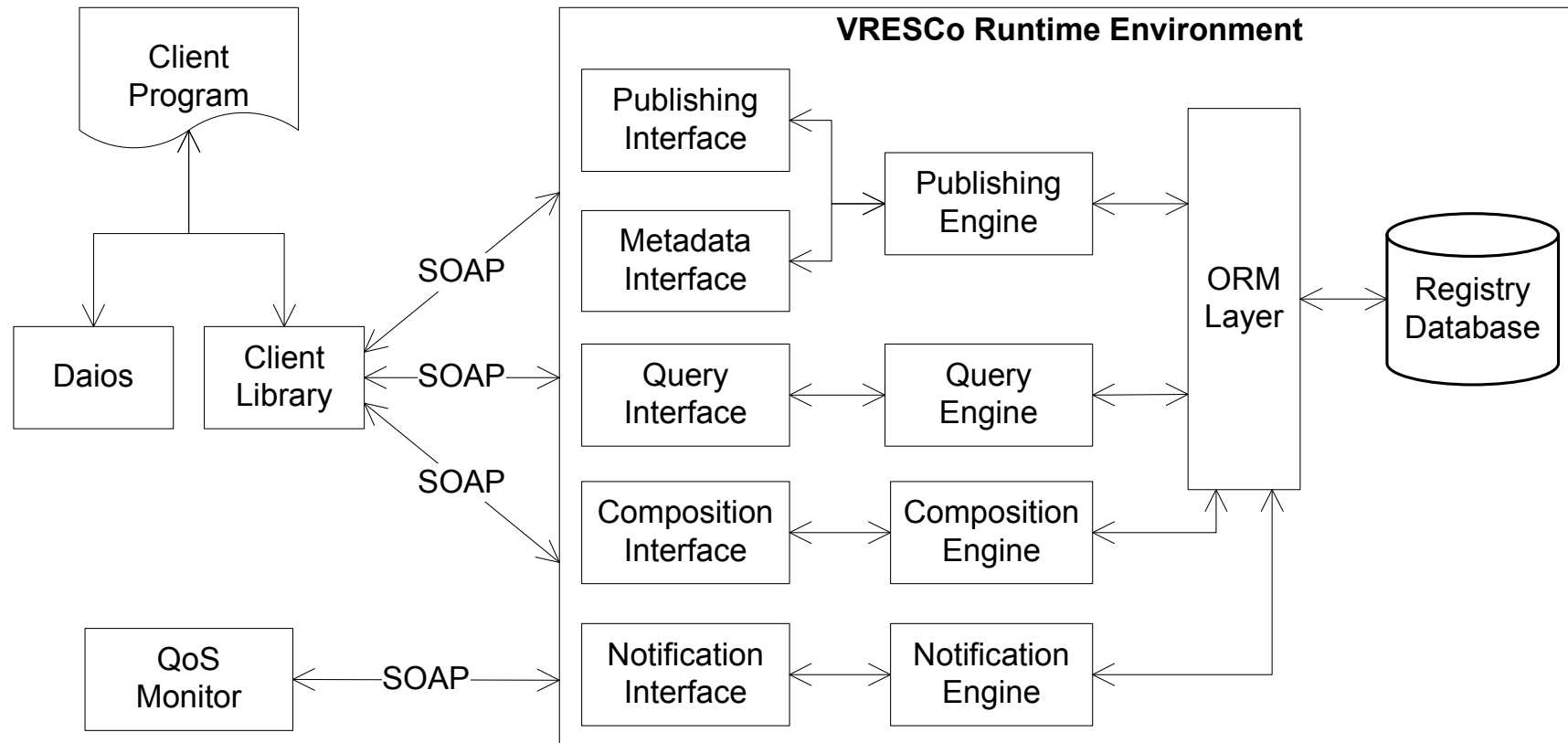
<http://www.infosys.tuwien.ac.at>



Outline

- Introduction to VRESCo
- Motivation
- The VRESCo Metadata Model
- Summary / Outlook

Vienna Runtime Environment for Service-Oriented Computing





Outline

- Introduction to VRESCo
- Motivation
- Metadata Model
- Summary / Outlook

Introduction

- Metadata is important
 - to **describe** decoupled components
 - to **use** and **compose** decoupled components
- This is well-known in the SOA community
 - Many WS standards define metadata or their handling
 - WSDL, UDDI, WS-MEX, ...

Metadata: one of the cornerstones of SOA

Assumptions

- Basic assumptions:
 - Services represent **low-level** implementations of higher-level business objectives ...
 - ... for the metadata model, it is essential to **abstract** from these implementation details, and ...
 - ... **map** concrete services to business objectives.
 - This mapping is specific for each company.

The mapping is not specified by the service provider, but by every consuming company separately.

Motivation (1)

- Metadata includes:

- Functionality
(what 'does' a service?)
- Inputs
- Outputs
- Preconditions
(for stateful services)
- Postconditions
(for stateful services)
- ... other, more specific
metadata ...

A large, light grey arrow with a black outline, pointing from the metadata list on the left towards the text on the right.

Defined in the
company
domain model

Motivation (2)

What about existing standards?

- **WSDL** is not enough
 - Packed with implementation details
 - Contains only 'interface' information

```
name="KeyOfficer">
  minOccurs="0" maxOccurs="1" name="Position_Title" type="s:string"/>
  minOccurs="0" maxOccurs="1" name="Inverted_Name" type="s:string"/>
  minOccurs="0" maxOccurs="1" name="Actual_Title" type="s:string"/>
  minOccurs="0" maxOccurs="1" name="Label_Name" type="s:string"/>
  minOccurs="0" maxOccurs="1" name="Label_Title" type="s:string"/>
```

- **OWL-S** is ... probably too much

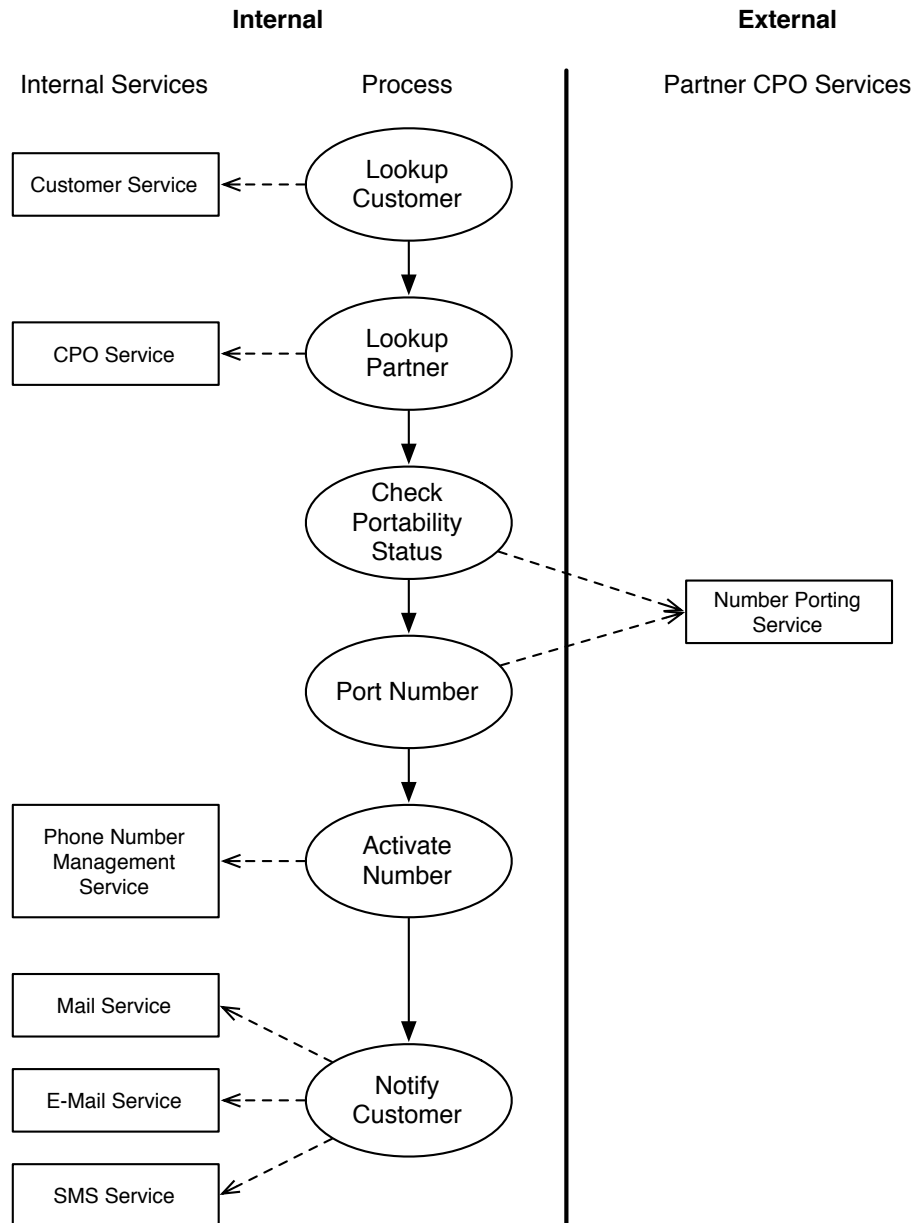
- Hard to use
- Semantic web ideas (public ontologies, etc...) are not widely accepted in business

```
<service:Service rdf:ID="ExpressCongoBuyService">
  <!-- Reference to the Profile -->
  <service:presents rdf:resource="#congo_profile;#Profi

  <!-- Reference to the Process Model -->
  <service:describedBy rdf:resource="#congo_process;#Ex

  <!-- Reference to the Grounding -->
  <service:supports rdf:resource="#congo_grounding;#Con
</service:Service>
```

An Illustrative Example



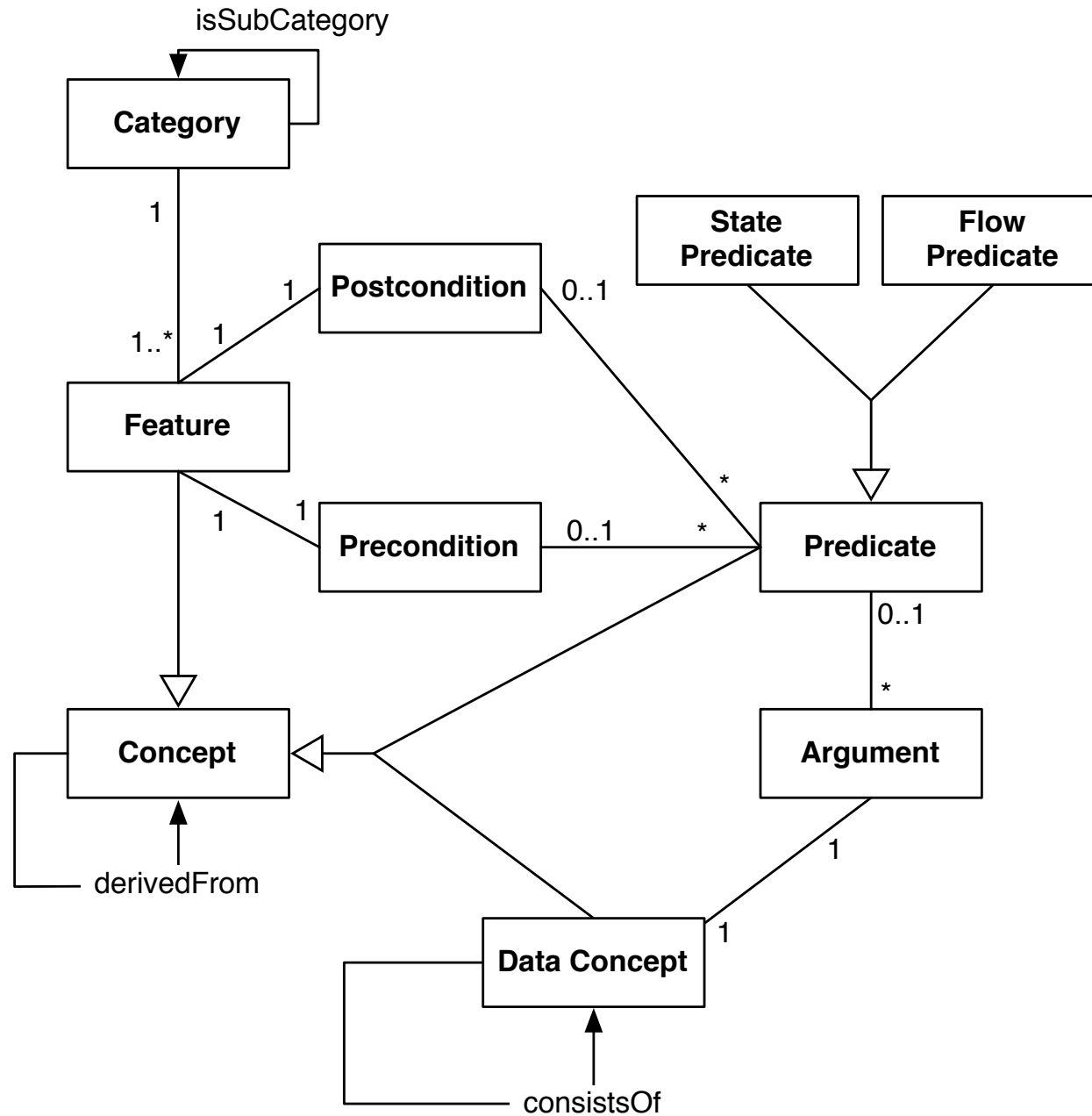
- No 1:1 mapping of activities to services
- Services may 'do the same thing', but differ in implementation
- Possible alternatives are **not static**



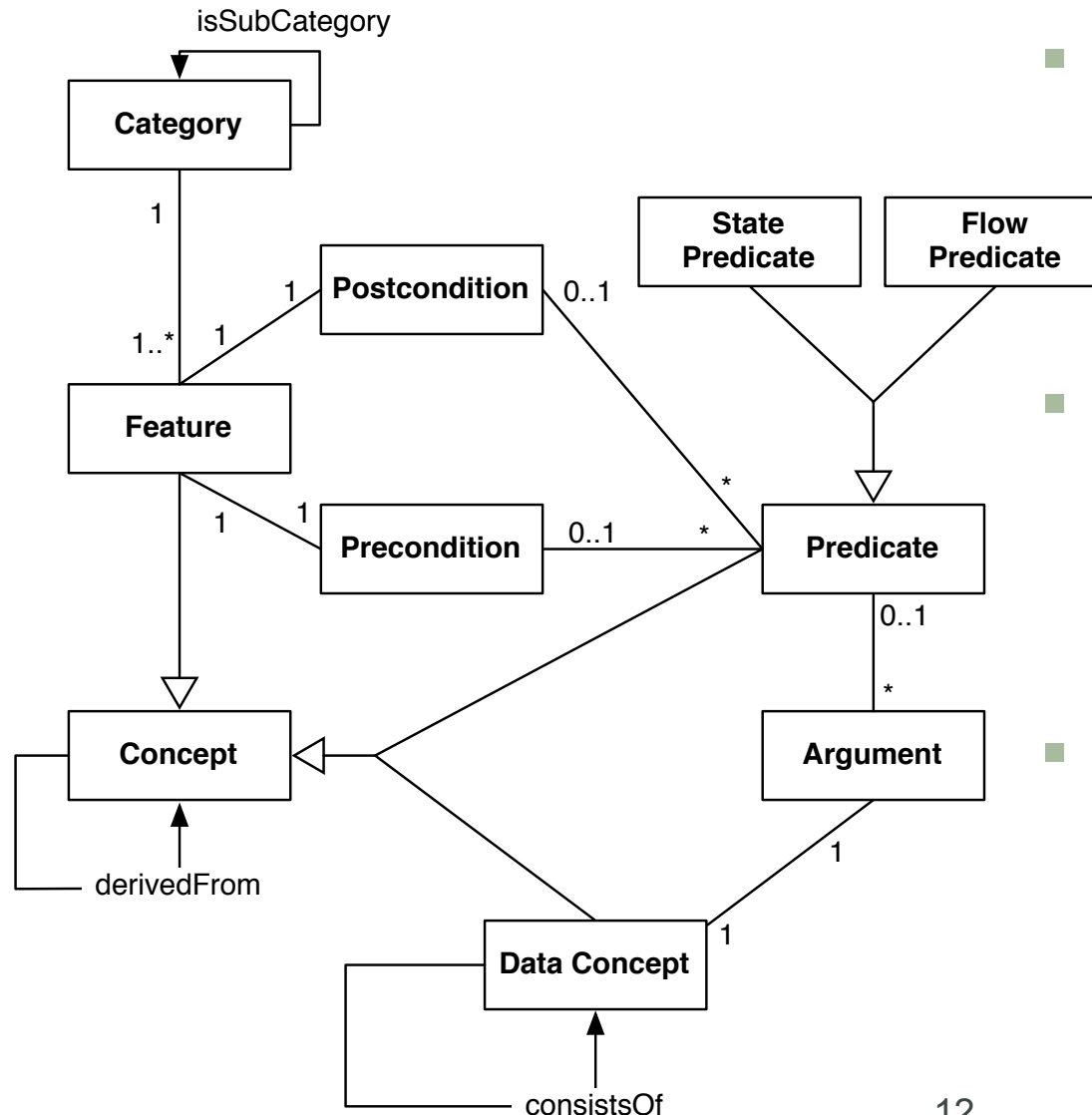
Outline

- Introduction to VRESCo
- Motivation
- The VRESCo Metadata Model
- Summary / Outlook

VRESCo Metadata Model (1)



VRESCo Metadata Model (1)



- **Features** are activities in the domain
 - *‘port number’, ‘notify customer’, ...*

- **Categories** are used to logically group features
 - *‘phone number porting’, ...*

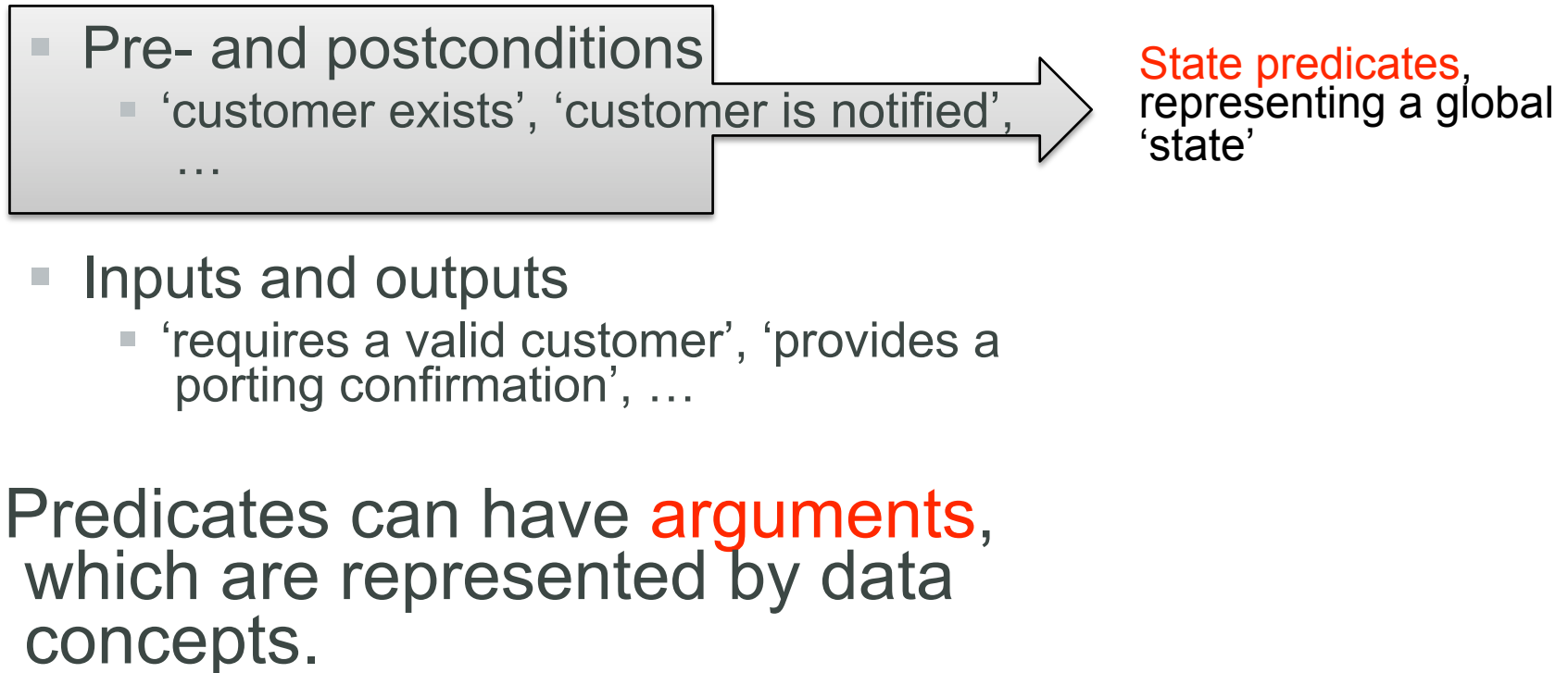
- **Data Concepts** are entities in the domain, which can be defined recursively
 - *‘telephone number’, ‘notification’, ...*

VRESCo Metadata Model (2)

- **Predicates** are used to define
 - Pre- and postconditions
 - 'customer exists', 'customer is notified', ...
 - Inputs and outputs
 - 'requires a valid customer', 'provides a porting confirmation', ...
- Predicates can have **arguments**, which are represented by data concepts.

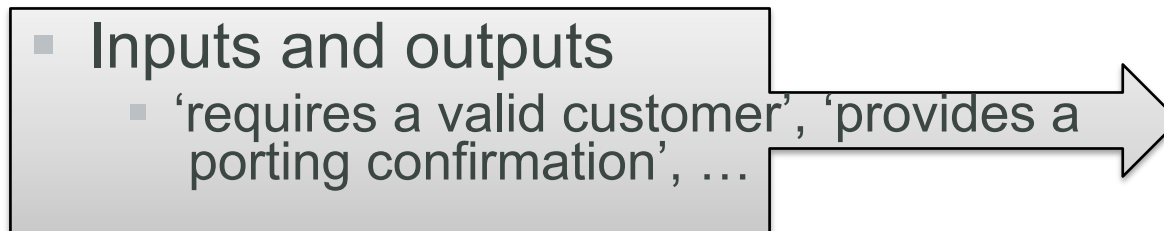
VRESCo Metadata Model (2)

- **Predicates** are used to define



VRESCo Metadata Model (2)

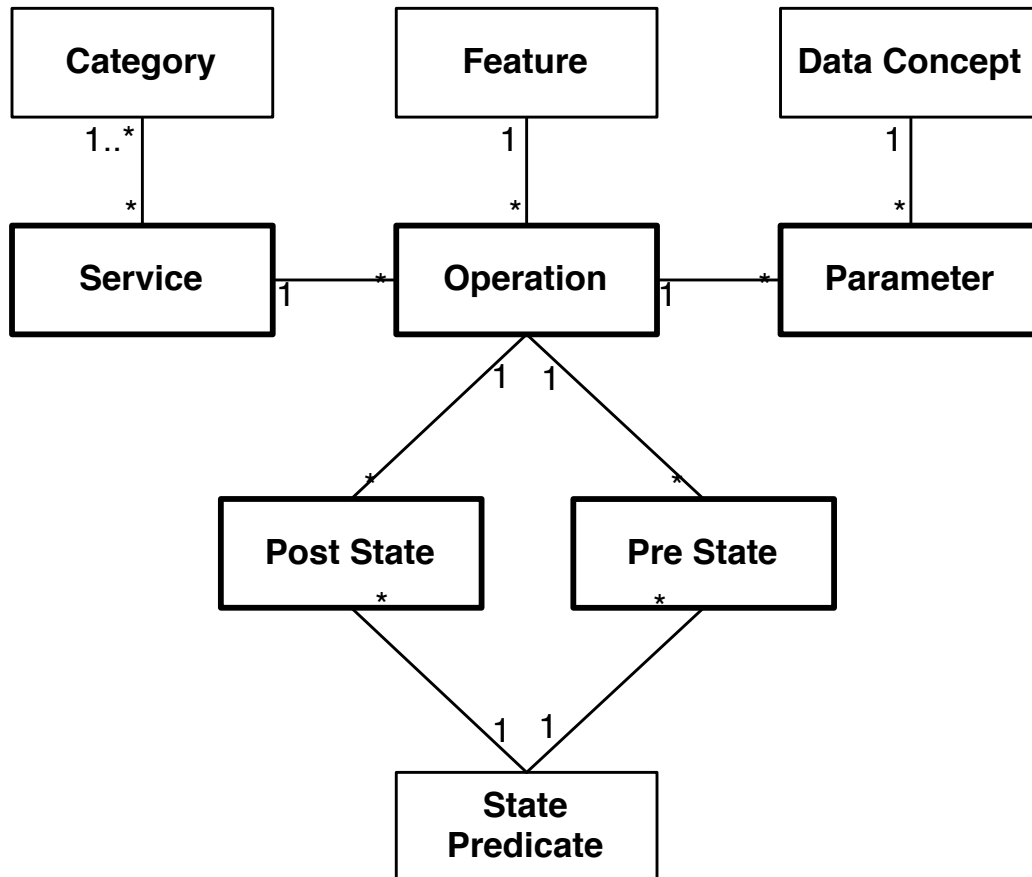
- **Predicates** are used to define
 - Pre- and postconditions
 - 'customer exists', 'customer is notified', ...



Flow predicates,
representing the flow
of information

- Predicates can have **arguments**, which are represented by data concepts.

Mapping Services (1)



- Services are mapped to categories (* ... [1; *])
- Operations are mapped to features (currently we assume a 1:1 mapping)
- Operations can have state requirements
- Operation params are mapped to data concepts
 - Usually, **transformations** will be necessary

Mapping Services (2)

- So how does mapping really work?

- 4 Steps:
 1. Define how params map to data concepts
 2. [Define state requirements for stateful operations]
 3. Define how operations map to features
 - Simple, since possible features are already limited by param mapping
 4. Define transformation functions for parameters, where necessary

Outline

- Introduction to VRESCo
- Motivation
- The VRESCo Metadata Model
- Summary / Outlook

Summary

- Metadata is an **important aspect** of SOA
- In VRESCo, we enrich service descriptions with an **reasonable** amount of **expressive** metadata

- Future work:
 - Use this metadata for
 - *Semi-automated service composition*
 - How can services be connected sensefully?
 - Flow of data concepts, ...
 - *Automated mediation / exchange of services*
 - Resolving interface and protocol differences
 - ...
 - Enhance **tool support** for mapping

