Towards Automated Alignment of Web Services to Requirements

Bertrand Verlaine, Yves Dubois, Ivan J. Jureta and Stéphane Faulkner

PReCISE Research Center – Louvain School of Management – University of Namur, Belgium
In a Nutshell

- **Fact**: Selecting a Web Service (WS) means knowing what to look for (Requirement Engineering (RE))
- **Current research goals**: Improving the selection by specifying what the WS customer wants. But: concepts used in RE differ from those used in the service paradigm.
- **Our Approach**: Mapping the two conceptualizations => decreasing the gap between RE and WS requests.
Agenda

1. Our Approach
2. The RE Ontology: CORE
3. The Service Taxonomy: WSDL & WSLA
4. First Mapping Step
5. Mappings Refinement and Formalization
6. Using the Mappings: STR@WS
7. Conclusion
8. Discussions
1. Our Approach

- Choosing appropriate conceptualizations
- Mapping the two requirements representations (2 steps)
2. The RE ontology: CORE (Jureta et al., Applied Ontology, 2009)
CORE concepts (1/2)

- **Communicated information**: propositions communicated by the stakeholder
- **Goal**: captures stakeholder’s desires
  - *Functional goal*: binary desired conditions of the system-to-be
  - *Quality constraint*: non-binary measurable properties of the system-to-be
  - *Softgoal*: non-measurable and vague properties of the system-to-be
- **Plan**: captures stakeholder’s intentions
CORE concepts (2/2)

- **Domain assumption**: captures stakeholder’s beliefs
  - *Functional domain assumption*: binary properties of the system-to-be
  - *Quality domain assumption*: non-binary measurable properties of the system-to-be
  - *Soft domain assumption*: non-measurable properties of the system-to-be

- **Evaluation**: captures stakeholder’s preferences
  - *Comparative evaluation*: comparison of two requirements
  - *Individual evaluation*: appraisal of an individual requirement
Formalization of CORE

Each conceptualization (CORE and WSDL/WSLA) rewritten using Description Logic $\mathcal{SIN}$
3. The Service Taxonomy: WSDL & WSLA

- **WSDL concepts:**
  - *Operation*: definition of the interactions between the WS agents
  - *Binding*: specification of the message format and transmission protocol details
  - *Service*

- **WSLA concepts:**
  - *Metric*: QoS measurement directives
  - *Obligations*: QoS values assurance (SLO) and guaranteed actions
  - *Party and Service definition*
Formalization of WSDL/WSLA

- Link between WSDL and WSLA to build the WSDL/WSLA taxonomy captured by Line 36

- Still using Description Logic $\text{SIN}$
4. First Mapping Step

- Mapping of the WSLA/WSDL concept with CORE concepts
- Example: a WS consumer can desire the presence or absence of a particular service property, captured by a *metric*
  This could be represented in WSLA

<table>
<thead>
<tr>
<th>WSLA concept</th>
<th>WSDL concept</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal</strong></td>
<td><strong>Metric</strong></td>
<td><strong>Obligations</strong></td>
<td><strong>Operation</strong></td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Plan</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Domain assumption</strong></td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td><strong>Evaluation</strong></td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
</tbody>
</table>
5. Mappings Refinement and Formalization

- The mappings between the CORE ontology and the WSDL/WSLA taxonomy
5. Mappings Refinement and Formalization

- The mappings between the CORE ontology and the WSDL/WSLA taxonomy
- Formalization with Distributed Description Logic (DDL) (Borgida & Serafini, Journal on Data Semantics, 2003)

59:  \( \text{CORE:FUNCTIONAL GOAL} \rightarrow \text{WSLA:METRIC} \sqcup \text{WSLA:ACTION GUARANTEE} \sqcup \text{WSDL:OPERATION} \)
60:  \( \text{CORE:QUALITY CONSTRAINT} \rightarrow \text{WSLA:SERVICE LEVEL OBJECTIVE} \sqcup \text{WSDL:BINDING} \)
61:  \( \text{CORE:PLAN} \rightarrow \text{WSLA:METRIC} \sqcup \text{WSLA:ACTION GUARANTEE} \sqcup \text{WSDL:OPERATION} \sqcup \text{WSDL:BINDING} \)
62:  \( \text{CORE:FUNCTIONAL DOMAIN ASSUMPTION} \rightarrow \text{WSLA:METRIC} \)
63:  \( \text{CORE:INDIVIDUAL EVALUATION} \rightarrow \text{WSLA:OBLIGATIONS} \)
64:  \( \text{CORE:COMPARATIVE EVALUATION} \rightarrow \text{WSLA:OBLIGATIONS} \)

- Presence of some gaps in the mappings (i.e., evaluations)
6. Using the Mappings Through a Tool: STR@WS

- **STR@WS** (under development): specifications transcribed from requirements in a WS environment

- **STR@WS’s components:**
  1. RequirementEditor
  2. Translator
  3. OpenFile
  4. SaveFile
Some Screenshots

A

B

C

D

<SLA>
  <Parties> ... </Parties>
  <ServiceDefinition>
    <Operation>
      <SLAParameter name="ResponseTime" type="long" unit="milliseconds">
        <Metric>ResponseTime<Computing</Metric>
      </SLAParameter>
      <Operation>
      </ServiceDefinition>
      <Obligations>
        <ServiceLevelObjective name="ResponseTime" serviceObject="WSDL.SOAP.GetQuote">"/>
        <Obligated>provider</Obligated>
        <Validity>
          <Start>xsd:dateTime</Start>
          <End>xsd:dateTime</End>
        </Validity>
        <Expression>
          <Predicate xsi:type="wsla:Less">"/>
          <SLAParameter>ResponseTime</SLAParameter>
          <Value>400</Value>
          <Predicate>
          </Expression>
          <EvaluationEvent>NewValue</EvaluationEvent>
        </ServiceLevelObjective>
      </Obligations>
    </Obligation>
  </ServiceDefinition>
</SLA>

<OP>receive coordinates</OP>
<OP>Provide postal address</OP>
<AG>Pay per use of $0.01 when the QoS is satisfied</AG>
<SLA>Answer within 600ms</SLA>
<SLA>Answer within 600ms</SLA>
<SLA>Maximum downtime of 10min</SLA>
<SLA>Available all the time (24 hours a day, 7 days a week)</SLA>
<AG>Prefer an answer within 400ms to 600ms</AG>
<AG>Free service if the QoS is not satisfied</AG>
7. Conclusion

- As for other systems, WSs need to be engineered (including the RE step!)
- Linking the RE conceptualization with the WS conceptualization in order to decrease the gap
- Automating the RE for service oriented systems
- Adjusting the service request based on the WS consumer’s requirements a runtime
- Future Work:
  - Using two ontologies (service side!)
  - Choosing two languages to express service specifications (RE and service side)
  - Taking into account the gaps between the expressiveness of the two conceptualizations (CORE > WSDL/WSLA)
8. Discussions

- Any Questions, Suggestions, Pointers, Criticism, Comments, Feedback,… ?

bertrand.verlaine@fundp.ac.be