I. Problem

- Let be \( I_Q, O_Q, R_Q \) a user query with:
  - \( I_Q \) a set of input attributes,
  - \( O_Q \) a set of output attributes and,
  - \( R_Q \) an execution risk indicating if the obtained results can be compensated (semantically undone) or not by the user.
- In the registry, each Web service (WS) is described (using OWL-S) by its:
  - functional property (i.e. set of input and output attributes),
  - transactional property (see Figure 1).

User query admissibility

- Goal: Verify that the WS registry contains
  (i) at least one WS whose TP satisfies \( R_Q \) and whose input are included in \( I_Q \),
  (ii) WSs, whose TP satisfies \( R_Q \) allowing to produce all output attributes of \( O_Q \).

Identification of the useful WSs

- Goal:
  - Define, for each attribute \( a \) of \( O_Q \), all the transactional sequences of WSs producing \( a \), using the transactional rules defined by [2],
  - Produce a WS \( WSDN \) containing only the transactional sequences for each \( a \).

Example: Let be \( Q = \{ A_1 \} \), \( O_Q = \{ A_1 \} \), \( R_Q = \) “no compensation”, with the registry of Figure 2

II. Our Contribution

- An extended Colored Petri Net (CPN) formalism representing the WS composition and incorporating transactional WSs properties,
- A CPN-based automatic WS selection algorithm satisfying the user query functional conditions expressed as input and output attributes, and transactional properties expressed as a risk level.

III. Colored Petri Net representation of the WS registry

Called Web Service Dependency Net (WSDN) with:
- Places: the input and the output attributes of the WSs,
- Transitions: the WSs,
- Color: the transactional properties of the corresponding WS or composite WS.

Example of WS registry:

Example of WS registry:

<table>
<thead>
<tr>
<th>WS Name</th>
<th>Input</th>
<th>Output</th>
<th>TP</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_1</td>
<td>A_1</td>
<td>A_1</td>
<td>y</td>
</tr>
<tr>
<td>A_2</td>
<td>A_2</td>
<td>A_2</td>
<td>y</td>
</tr>
</tbody>
</table>

Figure 2: The WSDN representing the WS registry of the above table.

Note: Duplication of some places in order to prevent exclusive transitions (places with only one successor).

IV. Our automatic WS selection approach

Inputs: The CPN representing the registry WSDN and the user query \( Q \)

Output: A CPN corresponding to a Transactional Composite WS satisfying \( Q \).

Figure 3: The steps of our approach.

V. Conclusion and future work

Our contribution: a transactional-driven WSs selection and composition algorithm where functional conditions, expressed as input and output attributes, and transactional properties, expressed as a risk level, are considered at the same time to compute adequate transactional WS compositions.

Current extension: incorporating QoS parameters (e.g., execution time, price) in the Quality function without changing the algorithm.

Acknowledgments

This work was supported by the Franco-Venezuelan CNRS-FONACIT project N°22782 and the ANR PEPS project N°JC07_18038.

References