Model-Driven Performability Analysis of Composite Services

Paolo Bocciarelli, Andrea D'Ambrogio
Dept. of Computer Science, Systems and Production
University of Roma “Tor Vergata”
Roma (Italy)
Outline

- Performability Analysis in a SOA context
- QoS-Enabled WSDL (Q-WSDL)
- Model-Driven Performability Prediction
  - Performance prediction
  - Reliability prediction
  - Combined Approach: Performability prediction
- Example application
- Conclusions
Services Oriented Architectures

- Distributed applications are rapidly converging towards the adoption of a computing paradigm based on *service-oriented architectures* (SOA).

- The service-oriented architecture provides the necessary support for the consolidation of multiple services into a single *composite service* corresponding to the overall process.

- Service providers are interested to describe the *QoS characteristics* of offered services, specifically with regards to the *performance* and the *reliability*.

- An approach is needed to *integrate* the performance and the reliability prediction into services composition processes.
Composite Services

- A composite service can be seen as a set of services that cooperate to execute a process that defines the interaction workflow.

- Different paradigm of collaboration:

- Several languages was proposed to describe interactions between web services. We specifically focus on BPEL
QoS-Enabled WSDL

- A WSDL description is an XML document that contains all the information about service capabilities and invocation mechanisms
- A WSDL does not contain any description of QoS characteristics of Web Service
- A WSDL extension is needed
- The proposed (lightweight) extension is based on a metamodell transformation
- The WSDL and Q-WSDL metamodels are defined by use of the Meta Object Facility (MOF)
Performability Prediction

The proposed method is integrated into a model-driven service composition process
QoS Prediction

- In this work the QoS prediction is specifically carried out in terms of performability (joint analysis of performance and reliability).
- Performance predictions are achieved by use of LQN (Layered Queueing Network) models.
- The Reliability prediction is obtained from the MTTF values specified in Q-WSDL descriptions of component services.
- An algorithm is proposed in order to determine a “Reward Rate” for each candidate configuration, that takes into account both performance and reliability.
Performance Prediction: BPEL Executable Model

- The annotated AD representing the BPEL executable process, is translated into a LQN model
- The stereotypes introduced to extend UML Activity Diagram:

<table>
<thead>
<tr>
<th>Stereotype</th>
<th>Base Metaclass</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>«process»</td>
<td>ActivityPartition</td>
<td>BPEL process coordinator</td>
</tr>
<tr>
<td>«partner»</td>
<td>ActivityPartition</td>
<td>BPEL partner (component web service)</td>
</tr>
<tr>
<td>«receive»</td>
<td>Action</td>
<td>BPEL receive activity</td>
</tr>
<tr>
<td>«invoke»</td>
<td>Action</td>
<td>BPEL invoke activity</td>
</tr>
<tr>
<td>«reply»</td>
<td>Action</td>
<td>BPEL reply activity</td>
</tr>
</tbody>
</table>
Performance Prediction: Mapping of BPEL into LQNM

Basic Interaction

Flow
The transformation of BPEL `switch` and `while` constructs are dealt by associating PAprob/PArep tagged values to Action instances stereotyped as «PAstep», according to the SPT profile.

Tagged values are used to compute the number of calls to LQN entries.
Reliability Prediction

- It is assumed that the MTTF value is specified for each service provided by BPEL partners involved in the composite service.

- By assuming an exponential distribution probability for failures, the reliability associated to each AD node \( a \) (BPEL basic activity) can be computed as:

\[
R_a(t) = e^{-\frac{1}{\text{MTTF}_a}t}
\]
Reliability Prediction: Reduction Rules

**Sequence**

\[ R_a = R_{a_1} \times R_{a_2} \]

**Flow**

\[ R_a = \prod_{i=1}^{n} R_{a_i} \]

**While (loop with k iterations)**

\[ R_a = (R_{a_1})^k \]

**Switch**

\[ R_a = \sum_{i=1}^{n} p_i \times R_{a_i} \]
Performability Prediction: the idea

For each (initial) candidate configuration a reward rate is computed: an overall attribute that combines both the throughput and the reliability of the composite service.
Performability Prediction: The Algorithm

1. Generate the State Transition Diagram (STD)
2. Select a candidate configuration as the initial configuration
3. Use the reliability predictions to obtain the transition probabilities of the STD
4. Calculate the absorbing probabilities $P(CS_i)$ of being in a given working configuration ($i=1..n$) starting from the initial configuration;
5. Use the performance predictions to obtain the performance (in terms of the throughput) $T(CS_i)$ associated to each configuration, and assign it as a reward
6. Obtain the performability prediction in terms of the expected reward rate of the composite web service $CS$ given by:

$$RW(CS) = \sum_{i=1}^{n} P(CS_i) \cdot T(CS_i)$$
Let us consider a composite web service that provides an operation for creating travel plans. The process requires the following services:

- Flight Manager (FM) service
- Accommodation Manager (AM) service
- Transportation Manager (TM) service

Our intention is to show how the performability analysis can lead to results unexpected if performance and reliability attributes are dealt with separately.
Abstract Model

- The abstract model of the composite web service is built as a first step of the model-driven performability analysis.
- Let us suppose that two different candidate services are available to bind the TM service.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>TM\textsubscript{A}</th>
<th>TM\textsubscript{B}</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CarReservation</td>
<td>120 ms</td>
<td>90 ms</td>
</tr>
<tr>
<td>time demand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CabInfo</td>
<td>115 ms</td>
<td>84 ms</td>
</tr>
<tr>
<td>time demand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network bit rate</td>
<td>10 Mb</td>
<td>100 Mb</td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTTF</td>
<td>10\times10^8</td>
<td>7.2\times10^7</td>
</tr>
<tr>
<td>R(1year)</td>
<td>0.961</td>
<td>0.645</td>
</tr>
</tbody>
</table>
Performance Prediction

If the prediction activity is limited to performance-related attributes, the choice of the alternative denoted as $\text{TM}_B$ is to be preferred as initial configuration for composite service.
Perfomability Prediction: Step1: Definition of STD

- The State Transition Diagram represents the possible configurations that the composite service may undergo before experimenting a failure.
- The states represent the working configurations, while the transitions represent the probabilities to remain in a configuration or move to a different one.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS₀</td>
<td>Composite Service in the failed state</td>
</tr>
<tr>
<td>CS₁</td>
<td>Composite Service including TMₐ</td>
</tr>
<tr>
<td>CS₂</td>
<td>Composite Service including TMₐ</td>
</tr>
</tbody>
</table>

The diagram shows the states and transitions between them.
At the second step of the algorithm, a candidate initial configuration is selected (two alternative \( \text{STD}_A \) and \( \text{STD}_B \)).

At the third step, the transition probabilities in the STD are obtained by applying the reliability prediction method.

<table>
<thead>
<tr>
<th>( P )</th>
<th>( \text{STD}_A )</th>
<th>( \text{STD}_B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{11} )</td>
<td>0.746</td>
<td>0.746</td>
</tr>
<tr>
<td>( p_{22} )</td>
<td>0.385</td>
<td>0.385</td>
</tr>
<tr>
<td>( p_{12} )</td>
<td>0.039</td>
<td>0.385</td>
</tr>
<tr>
<td>( p_{21} )</td>
<td>0</td>
<td>0.355</td>
</tr>
<tr>
<td>( P_{10} )</td>
<td>0.215</td>
<td>0.254</td>
</tr>
<tr>
<td>( p_{20} )</td>
<td>0.615</td>
<td>0.260</td>
</tr>
</tbody>
</table>
Perfomability Prediction: Step2-3: Values for STD_A

- $p_{11} = 0.746$ (reduction rules)
- $p_{12} = 1 - R_{TMA}(1\text{year}) = 0.039$
- $p_{10} = 1 - p_{11} - p_{12} = 0.215$
- $p_{21} = 0$
- $p_{22} = 0.385$ (reduction rules)
- $p_{20} = 1 - p_{22} - p_{21} = 0.615
**Perfomability Prediction: Step4: Absorbing Probabilities**

- The fourth step of the algorithm calculates the absorbing probabilities $P(CS_i)$ of being in a given working configuration starting from the initial configuration $CS_1$ (in the STD$_A$ case) or from the initial configuration $CS_2$ (in the STD$_B$ case).

<table>
<thead>
<tr>
<th>Alternative</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD$_A$</td>
<td>$P(CS_1) = p_{11}$</td>
</tr>
<tr>
<td></td>
<td>$P(CS_2) = p_{12} * p_{22}$</td>
</tr>
<tr>
<td></td>
<td>$P(CS_0) = p_{12} * p_{20} + p_{10}$</td>
</tr>
<tr>
<td>STD$_B$</td>
<td>$P(CS_1) = p_{21} * p_{11}$</td>
</tr>
<tr>
<td></td>
<td>$P(CS_2) = p_{22}$</td>
</tr>
<tr>
<td></td>
<td>$P(CS_0) = p_{21} * p_{10} + p_{20}$</td>
</tr>
</tbody>
</table>
Perfomability Prediction: Step 5-6: Expected Reward Rate

- At the fifth step, the performance prediction is carried out to obtain the performance (in terms of the throughput) T(CSi) associated to each configuration.
- At the sixth step, the expected reward rate is obtained by use of:

\[ RW(\text{CS}) = \sum_{i=1}^{n} P(\text{CS}_i) \cdot T(\text{CS}_i) \]

The performability prediction shows that an initial configuration with \( \text{TM}_A \) is to be preferred, in contrast with what obtained from the performance prediction.
Conclusions

- Describing and predicting the QoS of composite services is a challenging and strategic issue in SOA contexts.

- We have introduced a model-driven approach for predicting the performability of composite services specified by use of BPEL.

- Our approach exploits an already available method for performance prediction founded on Q-WSDL.

- We have introduced a model-driven method for the reliability prediction of composite services.

- Such method has been then combined to the performance-related one in order to obtain a combined prediction quantified in terms of performability.

- Work is in progress to implement the proposed method by use of existing performability evaluation tools.
Thank You!