Conformance Checking of Service Behavior

A. Rozinat (a.rozinat@tue.nl)

Group Information Systems
Eindhoven University of Technology

IPA Lentedagen 04.04.2007
Motivation

How to check whether interacting parties stick to what they agreed upon?
Outline

1. Process Mining
2. Conformance Checking
   1. Fitness
   2. Appropriateness
3. Case Study Town Hall
3. Application to Web Services
4. Feasibility Study
5. Conclusion
1. Process Mining

- Occurrence of activities:
  - what
  - when
  - who
  - ...

“world”
- business processes
- people
- machines
- components
- organizations

Supports/controls
- specifies
- configures
- implements
- analyzes

Verification
- models
- analyzes

Discovered
- process/system model
- descriptive
- prescriptive

Occurrence of activities:
- what
- when
- who
- ...

Software system
- records
- events, e.g.,
- messages,
- transactions,
- etc.

Event logs
- discovery
- conformance

Descriptive

Prescriptive

X-Ray
- Wafer stepper
- Copier

WFM
- CRM
- ERP
- PDM
2. Conformance Checking

“How compliant are the observed process executions?”
“Is the model a good representation of the observed behavior?”

<table>
<thead>
<tr>
<th>No. of Instances</th>
<th>Log Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>1207</td>
<td>ABDEA</td>
</tr>
<tr>
<td>145</td>
<td>ACDGHFA</td>
</tr>
<tr>
<td>56</td>
<td>ACGDHFA</td>
</tr>
<tr>
<td>23</td>
<td>ACHDFHA</td>
</tr>
<tr>
<td>28</td>
<td>ACDHFA</td>
</tr>
</tbody>
</table>
2. Conformance Checking

Objectives:
- quantitatively measure conformance (i.e., metrics)
- locate deviations

<table>
<thead>
<tr>
<th>No. of Instances</th>
<th>Log Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>1207</td>
<td>ABDEA</td>
</tr>
<tr>
<td>145</td>
<td>ACDGHFA</td>
</tr>
<tr>
<td>56</td>
<td>ACGDHFA</td>
</tr>
<tr>
<td>23</td>
<td>ACHDFA</td>
</tr>
<tr>
<td>28</td>
<td>ACDHFA</td>
</tr>
</tbody>
</table>

Conformance?
2.1 Conformance Checking - Fitness

<table>
<thead>
<tr>
<th>No. of Instances</th>
<th>Log Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>4070</td>
<td>ABDEA</td>
</tr>
<tr>
<td>245</td>
<td>ACDGHFA</td>
</tr>
<tr>
<td>56</td>
<td>ACGDHFA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. of Instances</th>
<th>Log Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>1207</td>
<td>ABDEA</td>
</tr>
<tr>
<td>145</td>
<td>ACDGHFA</td>
</tr>
<tr>
<td>56</td>
<td>ACGDHFA</td>
</tr>
<tr>
<td>23</td>
<td>ACHDFA</td>
</tr>
<tr>
<td>28</td>
<td>ACDHFA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. of Instances</th>
<th>Log Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>BDE</td>
</tr>
<tr>
<td>7</td>
<td>AABHF</td>
</tr>
<tr>
<td>15</td>
<td>CHF</td>
</tr>
<tr>
<td>6</td>
<td>ADBE</td>
</tr>
<tr>
<td>1</td>
<td>ACBGDFAAA</td>
</tr>
<tr>
<td>8</td>
<td>ABEDA</td>
</tr>
</tbody>
</table>
2.1 Conformance Checking - Fitness

No. of Instances | Log Traces
---|---
4070 | ABDEA
245 | ACDGHFA
56 | ACGDHFA

No. of Instances | Log Traces
---|---
1207 | ABDEA
145 | ACDGHFA
56 | ACGDHFA
23 | ACHDFHA
28 | ACDHFA

No. of Instances | Log Traces
---|---
24 | BDE
7 | AABHF
15 | CHF
6 | ADBE
1 | ACGDFAAA
8 | ABEDA
2.1 Measuring fitness: Log replay analysis

<table>
<thead>
<tr>
<th>No. of Instances</th>
<th>Log Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>1207</td>
<td>ABDEA</td>
</tr>
<tr>
<td>145</td>
<td>ACDGHFA</td>
</tr>
<tr>
<td>56</td>
<td>ACGDHFA</td>
</tr>
<tr>
<td>23</td>
<td>ACHDFA</td>
</tr>
<tr>
<td>28</td>
<td>ACDHFA</td>
</tr>
</tbody>
</table>

\[ f = 1 - \frac{23 + 28}{(1207 \cdot 7) + ((145 + 56) \cdot 9) + ((23 + 28) \cdot 8)} \approx 0.995 \]

\[ f = \frac{1}{2} \left( 1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i} \right) + \frac{1}{2} \left( 1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i} \right) \]
2.1 Measuring fitness: Log replay analysis

<table>
<thead>
<tr>
<th>No. of Instances</th>
<th>Log Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>4070</td>
<td>ABDEA</td>
</tr>
<tr>
<td>245</td>
<td>ACDGHFA</td>
</tr>
<tr>
<td>56</td>
<td>ACGDHFA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. of Instances</th>
<th>Log Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>1207</td>
<td>ABDEA</td>
</tr>
<tr>
<td>145</td>
<td>ACDGHFA</td>
</tr>
<tr>
<td>56</td>
<td>ACGDHFA</td>
</tr>
<tr>
<td>23</td>
<td>ACHDFA</td>
</tr>
<tr>
<td>28</td>
<td>ACDHFA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. of Instances</th>
<th>Log Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>BDE</td>
</tr>
<tr>
<td>7</td>
<td>AABHF</td>
</tr>
<tr>
<td>15</td>
<td>CHF</td>
</tr>
<tr>
<td>6</td>
<td>ADBE</td>
</tr>
<tr>
<td>1</td>
<td>ACGDFA</td>
</tr>
<tr>
<td>8</td>
<td>ABEDA</td>
</tr>
</tbody>
</table>

\[ f = 1.0 \]

\[ f \approx 0.955 \]

\[ f \approx 0.540 \]
2.2 Conformance Checking - Appropriateness

100 % fitness but not sufficiently specific from behavioral point of view.

<table>
<thead>
<tr>
<th>No. of Instances</th>
<th>Log Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>1207</td>
<td>ABDEA</td>
</tr>
<tr>
<td>145</td>
<td>ACDGHF</td>
</tr>
<tr>
<td>56</td>
<td>ACAGDFH</td>
</tr>
<tr>
<td>23</td>
<td>ACHDF</td>
</tr>
<tr>
<td>28</td>
<td>ACDHFA</td>
</tr>
</tbody>
</table>

100 % fitness but not represented in structurally suitable way.
2.3 Case Study - Town Hall
2.3 Case Study - Town Hall
2.3 Case Study - Town Hall
2.3 Case Study - Town Hall
2.3 Case Study - Town Hall
3. Application to Web Services

- Service-oriented systems are composed of services that are:
  - independently developed and operated
  - interact with one another exclusively through message exchanges

⇒ Expected behavior may deviate
  - e.g., service receives reply of the wrong type, messages may be received in the wrong order, ... 

Question: “Do all services in a service-oriented system operate as expected?”
3. Application to Web Services - Approach

Ingredients:

- Translation from BPEL to Petri nets
- Mapping from SOAP messages to Event log
3.1 From BPEL to Petri Nets

```
<process name="unreachableTask"
  targetNamespace="http://samples.otn.com"
  suppressJoinFailure="yes"
  xmlns:tns="http://samples.otn.com"
  xmlns:services="http://services.otn.com"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process">
  <flow name="FL" suppressJoinFailure="yes">
    <links>
      <link name="x1"/>
      <link name="x2"/>
    </links>
    <switch name="SW">
      <case>
        <invoke name="A1">
          <sources><source linkName="x1"/></sources>
        </invoke>
      </case>
      <otherwise>
        <invoke name="A2">
          <sources><source linkName="x2"/></sources>
        </invoke>
      </otherwise>
    </switch>
    <invoke name = "A3">
      <targets>
        <joinCondition>
          bpws:getLinkStatus(‘x1’) and bpws:getLinkStatus(‘x2’)  
        </joinCondition>
        <target linkName="x1"/>
        <target linkName="x2"/>
      </targets>
      <invoke>
      </flow>
  </process>
```

Legend:
- Basic Activity
- Flow
- Switch
- Control Link
3.1 From BPEL to Petri Nets

Translation is feature-complete:
- maps all BPEL aspects on a Petri-net model

3.2 Monitoring and Correlating Messages

From SOAP messages to Event log:
- Process instance (PI)
- Label denoting the BPEL communication action (MT)
4. Feasibility Study - Overview

- abstract BPEL process
- BPEL2PNML + WofBPEL
- WF-net

Service Behavior
- SOAP Messages
- Conformance?
- Oracle BPEL Server
- Oracle BPEL Server

1. TCP tunneling + Conversion
2. MXML Event Log
3. Conformance Checker (ProM)
4.1 From BPEL to WF-nets (1)

- Translation: BPEL2PNML
4.1 From BPEL to WF-nets (2)

- Reduction: WofBPEL
4.2 Observing Service Behavior

Implementation via Oracle BPEL:
- Supplier service
- Customer service

→ could have been implemented on any other platform! (e.g., Java, .NET, etc.)

**Result**: Observation of one out of many possible communication scenarios
4.2 Creating Service Behavior - JDeveloper
4.2 Creating Service Behavior - BPEL Worklist
4.2 Observing Service Behavior - SOAP to MXML

//Process@id  Value of //ServiceName@xmlns:snns
//Process@description  Added manually

...<Process
    id="http://services.qut.com/Supplier"
    description="Supplier 1.1, using Customer 1.1 as customer stub"
>
...

  <ProcessInstance
    id="bpel://localhost/default/Customer~1.1/301-BpInv0-BpSeq0.3-3"
    description="Instance 301"
  >
...

  <AuditTrailEntry>
    <WorkflowModelElement>order</WorkflowModelElement>
    <EventType>complete</EventType>
    <Timestamp>2005-10-20T11:54:09-00:00</Timestamp>
  </AuditTrailEntry>
...
4.2 Observing Service Behavior - SOAP to MXML

```xml
//ProcessInstance@id Value of //MessageID
//ProcessInstance@description "Instance " concatenated with value of //MessageId@orabpel:rootId

...<Process
    id="http://services.qut.com/Supplier"
    description="Supplier 1.1, using Customer 1.1 as customer stub"
>
...

<!--ProcessInstance
    id="bpel://localhost/default/Customer~1.1/301-BpInv0-BpSeq0.3-3"
    description="Instance 301"
>
...

<!--AuditTrailEntry>
    <WorkflowModelElement>order</WorkflowModelElement>
    <EventType>complete</EventType>
    <Timestamp>2005-10-20T11:54:09-00:00</Timestamp>
<!--AuditTrailEntry>
...
4.2 Observing Service Behavior - SOAP to MXML

//WorkflowModelElement Value of SOAPAction
//EventType "complete"
//TimeStamp (Converted) timestamp of response (as the request does not contain any timestamp)

...<Process
    id="http://services.qut.com/Supplier"
    description="Supplier 1.1, using Customer 1.1 as customer stub"
  >
...
...<ProcessInstance
    id="bpel://localhost/default/Customer~1.1/301-BpInv0-BpSeq0.3-3"
    description="Instance 301"
  >
...
...<AuditTrailEntry>
  <WorkflowModelElement>order</WorkflowModelElement>
  <EventType>complete</EventType>
  <Timestamp>2005-10-20T11:54:09-00:00</Timestamp>
</AuditTrailEntry>

...
4.3 Conformance Checking

- Desirable and undesirable scenarios for the supplier service execution

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Fitness</th>
<th>Log trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>(order, orderResponse)</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>(order, orderResponse, orderResponse, orderResponse)</td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>(order, orderResponse, change, orderChangeResponse)</td>
</tr>
<tr>
<td>4</td>
<td>1.0</td>
<td>(order, orderResponse, orderResponse, change, orderChangeResponse)</td>
</tr>
<tr>
<td>5</td>
<td>1.0</td>
<td>(order, orderResponse, change, orderResponse, orderChangeResponse)</td>
</tr>
<tr>
<td>6</td>
<td>0.625</td>
<td>(order)</td>
</tr>
<tr>
<td>7</td>
<td>0.749</td>
<td>(order, orderResponse, change)</td>
</tr>
<tr>
<td>8</td>
<td>0.905</td>
<td>(orderResponse)</td>
</tr>
<tr>
<td>9</td>
<td>1.0</td>
<td>(order, orderResponse, change, orderResponse, orderChangeResponse)</td>
</tr>
<tr>
<td>10</td>
<td>0.759</td>
<td>(order, change, orderChangeResponse)</td>
</tr>
<tr>
<td>11</td>
<td>0.0</td>
<td>(change)</td>
</tr>
<tr>
<td>12</td>
<td>0.914</td>
<td>(order, orderResponse, change, orderChangeResponse, change)</td>
</tr>
<tr>
<td>13</td>
<td>0.971</td>
<td>(order, orderResponse, change, change, orderChangeResponse)</td>
</tr>
</tbody>
</table>
4.3 Conformance Checking

- Analysis scenario 8: (orderResponse)

(a) The fitness analysis of scenario No. 8 shows that ‘orderResponse’ was not ready to be executed when it occurred (tokens were missing), and that ‘order’ was expected to occur but did not happen (tokens were remaining)
4.3 Conformance Checking

- Behavioral Appropriateness Analysis scenarios 1-5

(b) The behavioral appropriateness analysis based on the desirable scenarios reveals that the model allows for more behavior than expected. Due to intermediate states it is possible to send an ‘orderResponse’ after a ‘change’ request has been received.
5. Conclusion

- Conformance checking particularly relevant in a service-based environment due to the independence of services

- Feasible to check conformance of an abstract BPEL process using existing tool chain (BPEL2PNML, WofBPEL, ProM Conformance Checker)