Adaptation of Service-based Systems
Topics

- Service-based Systems
- Adaptation
  - Classification of Triggers and Approaches
- Adaptation Approaches from our Research
- Summary/Conclusion
The Famous Travel Agency Process

- Choreography of several applications/compositions

© Dimka Karastoyanova
Multi-Scale Human Skeleton Simulation

- Choreography of simulations of skeleton, bone, tissue, and cell scale

Bone Growth Simulation

- Orchestration to simulate bone growth depending on load, exercise, etc.
  - Understand diseases, e.g. fractures
- Based on the Finite Element Method (FEM)
  - Initial and boundary condition
  - Simulation is solved via a PDE as a matrix equation \((Ax=b)\)
Bone Growth Simulation

- Modeled and realized as a choreography
Service-Based Systems

- Service-Based Systems:
  - Choreographies of complex services/service-based applications (SBAs)
  - Services may be choreographies or orchestrations themselves
Service-Based Applications

- **SBA model:**
  - Three layers
  - Current research focus:
    - Model, execute, monitor and adapt on all three levels coherently

- **SBA Life cycle** → includes adaptation
- **Framework for QoS aware, adaptable SBAs**

---

**Choreography**

**Orchestrations**

**Services**

**Modeling**
- Modeling of metrics, KPIs, available adaptation actions etc.

**Business Process**
- Service Compositions
- Services

**Adaptation**
- for optimization of process performance

**Analysis**
- KPI Violations:
  - Analysis
  - Prediction

**Monitoring**
- Runtime monitoring and evaluation of KPIs

© Dimka Karastoyanova
Service-Based Applications (SBAs)

- **Services:**
  - Are units of functionality
  - Described using a unified IDL
  - Independent of implementation technology
  - Self-contained stable service interfaces
  - Virtualization of components

- **SBAs** comprise services
- And follow the principles of the Service Oriented Architecture (SOA) style
- The SOA roles and operations:

  - **Service Requestor**
  - **Find**
  - **Bind**
  - **Publish**
  - **Service Discovery**

- Technology for implementing SBAs is workflows
Service Compositions

- Get Order
- Always!
- Check Customer
- Good Customer?
- Accept Order
- Bad Customer?
- Reject Order
- Customer Address
- Credit Card Information
- Workitem
- Worklist
- What?
- With?
- Who?

© Dimka Karastoyanova
Adaptation of Service Compositions (How?)

- Adaptation on all dimensions
  - Control flow changes
  - Functions changes
  - Exchange human participants
- On model (evolution, versioning) and instance level (ad-hoc changes) (When?)
- SC Model and/or Infrastructure changes to accommodate adaptation (What realization?)
- Many Triggers for Adaptation (Why?)
Adaptation Triggers (Why?)

- Triggers can be generated on each level of an SBA
  - Value for participants - Service Networks
  - QoS and nfp violations, KPIs – SCs and BPs
  - SLAs violations – all layers
  - Compliance violations – all layers
  - Changes in policies – mostly BPs
  - Unavailable services – Service Infrastructure + SCs
  - Context change – all layers
  - Organizational restructuring – mostly business related
  - Law – mostly the BP layer
  - ...

© Dimka Karastoyanova
Next:

- Adaptation of SCs: BPEL’n’Aspects
- SBA Adaptation: KPI Violation prediction and Adaptation
- Process Outsourcing – Fragmentation and Coordination
- Flexible Scientific Workflows
BPEL’ n’ Aspects

- Adaptation of SCs
- Control flow change
- Reaction to any kind of trigger
- Standard-based
- Engine extensions needed
BPEL’n’Aspects

- Use of the AOP paradigm
- Insert WSs (aspect) into an SC (program) as a reaction to event
- Aspects are attached to processes
- WS invocation before, instead or after activities
  - i.e. control flow changes
- Engine publishes process execution events
- The dynamic weaving of the activities/WSs is triggered by these events
Aspect as a WS-Policy

<wsp:PolicyAttachment>
  <wsp:AppliesTo>
    <DomainExpression/>+
  </wsp:AppliesTo>
  (<wsp:Policy>...</wsp:Policy>
   | <wsp:PolicyReference>...
  )+
</wsp:PolicyAttachment>

<ws:Policy>

<wsp:All>
  <a4b:Aspect id="...">*
    <a4b:Advice name="..."
      compensating="true | false"
      alwaysCompensate="true | false"?/>
    <wsa:EndpointReference>...
    </wsa:EndpointReference>
  ...
</a4b:Advice>

<a4b:CompensationAspect aspectId="..."/>?

<a4b:Pointcut>...
  <a4b:ProcessArtifact type="activity
   | transitionCondition | "...">
  </a4b:ProcessArtifact>
  <a4b:When type="before | instead | after"/>
</a4b:Pointcut>?
</a4b:Aspect>
</wsp:All></ws:Policy>

© Dimka Karastoyanova
Architecture and Prototype

- **BPEL Engine**
  - Publishes relevant events:
    - activity status,
    - variable modification,
    - execution of implicit and explicit CHs

- **Aspect Management Tool**:
  - specify and deploy aspects

- **Weaver**
  - Weaves in aspects

- **Audit component**
  - Stores execution events published by the engine
  - Stores variable values at aspect weaving time needed for later compensation

@ Dimka Karastoyanova
Preventing Violations of KPIs

- Adaptation approach: any
- Adaptation trigger: KPI violations
- Engine extensions needed
- Run time approach
Adaptable and QoS-Aware Service Compositions

- For service compositions implementing business processes:
  - Use adaptation as reaction to the changes in QoS parameters of a Service-Based Application
  - Adaptation trigger: KPI targets are not reached
  - Approach: Analyze the reason for the KPI violation (influential factors) and optimize through adaptation
Runtime

Adaptation Enactment

Selection of an Adaptation Strategy

Preferences Model

List of alt. Adaptation Strategies

Identification of Adaptation Requirements and Strategies

Adaptation Action Model

KPI Dependency Analysis

Checkpoint Model

Classification Model(s)

Runtime KPI Prediction

Metrics Model

Metric Values

Metrics Model

Metric Values

Instance Tree(s)
Modeling

Specify performance goals of the application

KPIs
- Order Fulfillment Time
- Customer Satisfaction
- Order delivered in full ...

Preferences
- Fulfillment Time: 0.2
- Customer Sat.: 0.2
- Shipment Time: 0.05

Usage of weights to specify the relative importance of KPIs and metrics

Weights on characteristics of adaptable entities

based on key metrics + targets

Metrics
- Order In stock
- Supplier Delivery Time
- Shipper Delivery Time, Process Availability, ...

Impact measures characterized by adaptable entities

Adaptation Actions
- Use Supplier A, B, C, ...
- Use Shipper A, B, C, ...
- Skip credit rating check ...

Specify alternatives for adaptable entities and their impact on certain metrics

Application Model contains adaptable entities in the application (variability points)

Check Points
- After order receipt
- After warehouse check
- Before supplier invocation ...

Adaptable Entities
- Supplier Service
- Shipment Service
- Execution of credit rating check activity ...
Analysis and Prediction

- Analyze the influential factors and
- explain the KPI target violations (why?):
  - Explanation model is created using decision tree techniques (machine learning) based on historical process instances (monitoring)

- Result is a KPI dependency tree explaining
  - which metrics (combinations) and
  - which value ranges of those metrics lead to good or bad KPI values
Analysis and Prediction

- Predict the KPI class while the process instance is still running
- Using the prediction model (dependency tree) and runtime information/measured metrics
- Result: an instance tree showing the KPI classes in relation to characterizing metrics of adaptable entities → shows which adaptation actions are needed

Classification Model

Instance Tree

We want to achieve this path
→ Try to ensure that “shipment delivery time” < 2 days
Identify Adaptation Actions

Choose and combine adaptation actions which address adaptation requirements according to their impact model.

<table>
<thead>
<tr>
<th>Id</th>
<th>Adaptation Requirement</th>
<th>Predicted KPI Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SuTime&lt;4,5 AND ShTime&lt;3 AND Su=A</td>
<td>Green AND very good</td>
</tr>
<tr>
<td>2</td>
<td>SuTime&lt;4,5 AND ShTime&lt;2 AND Su=B</td>
<td>Green AND good</td>
</tr>
<tr>
<td>3</td>
<td>SuTime&lt;7,5 AND ShTime&lt;2 AND Su=A</td>
<td>Green AND very good</td>
</tr>
<tr>
<td>4</td>
<td>SuTime&lt;7,5 AND ShTime&lt;2 AND Su=B</td>
<td>Green AND good</td>
</tr>
</tbody>
</table>

© Dimka Karastoyanova
Architecture of the Prototype

Adaptation of the Prototype

Process Engine

QoS Monitor

CEP Engine

Metrics Database

Analysis Database

Classification Model Learner

Decision Tree

Strategy Identifier & Selector

Adaptation Enactor

Adaptation Interface

Check Point Listener

Listener

Delivery Time Shipment

green
red
red
< 2
2 < x < 4
x > 4

Classification Model Learner

Performance Dashboard

25% 50% 75%

Adaptation Dashboard

Prediction Dashboard

Monitoring

Analysis Dashboard

Performance Dashboard
Outsourcing Processes:

- Adaptation of SCs
- Control flow adaptation, split into fragments
- Trigger: Organizational Adaptation
- Design time (and Run time)
- Language extension
- Engine extension, Coordination Infrastructure
- Standard-based
Outsourcing: Splitting Processes

- Outsourcing part of a process using the concept of sub-processes
  - Autonomy of sub-processes
Splitting a Process to Multiple Organizations

### Alternatively:
- process fragment/partition is outsourced to an organization,
- each process is performed autonomously
Projecting Onto Organizations

- **Step 1**: Project process onto organizations
  - Define partitions/fragments
  - Split control connectors

- **Step 2**: Wire/connect the processes to model original business logic
Step 1: Deriving the Wiring

- Splitting Control Links
Deriving the Wiring (2)

- Splitting Control Links

![Diagram showing A splitting into A+ and A- with B connected to A+ and B', with an arrow labeled \( q \) from A to B]

Fault Handler, catching “q=0” as join-fault
Join Nodes

- Splitting Control Links leading to join nodes

© Dimka Karastoyanova
Step 3: Deploy partitions onto infrastructures

- Wires are split control connectors
- No changes in standards necessary if no structured activities are split
Splitting a Scope

- C can only start if A completed successfully
- If C completed successfully D may only run after S₁ completed successfully
- DPE: Link CE may be set to false in case S₁ detects a fault (but S₂ did not)
- S₂ may get compensated although no fault occurred at S₂

⇒ Coordination required
Flexible Scientific Workflows

- Adaptation of SCs
- Control flow change; any available approach
- Reaction to change in simulation workflow model
- Standard-based
- Engine extensions needed
Scientific Workflows

- Scientific Workflows
  - Scientific experiments/computations/simulations modeled and executed as workflows

- Characteristics:
  - deal with huge amounts of data,
  - are often long-running,
  - usually data driven,
  - can integrate multiple data sources (i.e. sensors, data bases, file systems, etc.)

Bone Growth Simulation:

© Dimka Karastoyanova
Differences in life cycles: scientific workflows vs. service compositions

Goal: enable scientific workflows using the life cycle of SCs
Execution of Partially Specified Processes?

- Execution of process instances is based on process model
- The model is considered to be complete by the execution engine
  - i.e. the instances are terminated after the last activity

Initial version of the model

Execute instance

Subsequent version of the model

Execute instance

Next version

© Dimka Karastoyanova
Our Approach: Model-as-you-go

In what workflow life cycle phase we are ... What the scientist experiences ...

Analysis

Modeling

Scientist

Run/Resume

Execution and Monitoring

Suspend

© Dimka Karastoyanova
In what workflow life cycle phase we are ...

What the scientist experiences ...

Computation is suspended at this point
In what workflow life cycle phase we are ...  

What the scientist experiences ... 

Computation is suspended at this point
Model-as-you-go

In what workflow life cycle phase we are ...

What the scientist experiences ...

Analysis

Modeling

Scientist

Run/Resume

Execution and Monitoring

Suspend

Computation is suspended at this point

© Dimka Karastoyanova
Model-as-you-go

In what workflow life cycle phase we are ...

What the scientist experiences ...

Analysis

Scientist

Modeling

Run/Resume

Execution and Monitoring

Suspend

Computation is suspended at this point
Model-as-you-go – Major Extensions

1) All-in-one
   - Modeling
   - Monitoring
   - Execution
   - Adaptation

2) Execution control

3) Adaptation as modeling experience

4) Iteration

5) Breakpoints

6) History of variable data
   - Var A = 1
   - Var A = 2

Failure!
Variable history

III = Instance Information Interface
EI = Event Interface
ECI = Execution Control Interface

Component
Model-as-you-go framework
Interaction between components

© Dimka Karastoyanova
Summary

- Adaptation of Service Based Systems and Applications:
  - Relevant for many application domains
  - Can be carried out
    - on one or more layers of an application / system and
    - across organizational boundaries
  - Is complex - consider dependencies to other aspects – data, domain, SBA architecture, triggers, etc.
  - Still a lot to be done

- Future:
  - Coherent cross-layer adaptation
  - Coherent approach to select adaptation action(s) for a trigger
  - Choreography adaptation, Cross organizational adaptation, split and merge
  - Views for different user types (Business Transactions, Service Networks) and their use for adaptation
References

- Excellence Cluster **SimTech**: [http://www.simtech.uni-stuttgart.de/](http://www.simtech.uni-stuttgart.de/)
- IAAS at the University of Stuttgart: [http://www.iaas.uni-stuttgart.de/](http://www.iaas.uni-stuttgart.de/)
- Wetzstein, Branimir; Zengin, Asli; Kazhamiakin, Raman; Marconi, Annapaoloa; Pistore, Marco; Karastoyanova, Dimka; Leymann, Frank: Preventing KPI Violations in Business Processes based on Decision Tree Learning and Proactive Runtime Adaptation. In: Journal of Systems Integration, January, 2012.
- Guinea, Sam; Kecskemeti, Gabor; Marconi, Annapaola; Wetzstein, Branimir: Multi-layered Monitoring and Adaptation. In Proceedings of ICSOC 2011.