EVENT-DRIVEN PROCESS ENGINES

Matthias Weidlich, Jan Sürmeli
Process-Driven Architectures
Permits visual modeling of processes including the annotation of model elements with produced and consumed event types.

**Task: Modeling**
- **Process Modeler**
  - **Read/write**
  - Consumes: Process Model Repository
  - Produces: PackageScanned

**Task: Execution**
- **Process Model Repository**
  - Read
  - Consumes: PackageScanned
  - Produces: PackageScanned
- **Process Execution Engine**
  - Read
  - Consumes: events occurring during process execution
  - Produces: matched complex events
- **Complex Event Processing Engine**
  - Read
  - Consumes: defined by a query to be evaluated over event streams, based on data attributes, time frame, and sequence patterns
  - Produces: matched complex events based on queries over incoming events
- **Environment**
  - Occurring events
  - Sensors and monitoring devices noticing and producing events

**Event Type Repository**
- **Simple Event Type**
- **Complex Event Type**
- Defined by data attributes

**Event Matching**
- Matches complex events based on queries over incoming events.
Primary goal: Extend an existing execution engine for BPMN models

- Import of extended process models from the process repository
- Sending produced events to a CEP engine
- Receiving and reacting on events matched by a CEP engine

Nice to have: Also extend the respective visual modeler

- Annotation of model elements with produced/consumed event types
- Retrieval, creation, and editing of event types stored in the event type repository
- Export of process models to the process repository
Event processing platform built on top of Esper

- Definition of monitoring queries
- Linking them to process models
- Monitoring statistics

Developed at Hasso Plattner Institute

Live system: https://bpt-lab.org/unicorn

Code base: https://github.com/bptlab/Unicorn
All engines

- Are open source
- Support BPMN 2.0
- Come with an editor
- Provide comparable functionality
- Show differences in terms of APIs, setup, etc.

Camunda (Java EE, REST): https://www.camunda.org

Processmaker (PHP, REST): http://wiki.processmaker.com

Bonitasoft (Java EE, REST): http://www.bonitasoft.com
<table>
<thead>
<tr>
<th>Tentative Dates</th>
<th>Phases</th>
<th>Meeting</th>
<th>Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.04.2016</td>
<td>Organisation and planning</td>
<td>all</td>
<td>Specification</td>
</tr>
<tr>
<td>until 10.05.2016</td>
<td>Domain and Requirement Analysis</td>
<td></td>
<td>File format &amp; test cases</td>
</tr>
<tr>
<td></td>
<td>Design: Interfaces, inter-team data structures, file formats and possible test cases</td>
<td>self-organised</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design: Intra-team data structures, architecture, algorithms</td>
<td>self-organised</td>
<td></td>
</tr>
<tr>
<td>31.05.2016</td>
<td>Inter-team presentation of design</td>
<td>all</td>
<td>System Design</td>
</tr>
<tr>
<td>14.06.2016</td>
<td>Inter-team presentation and mutual testing of prototype I</td>
<td>all</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Intermediate debugging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>05.07.2016</td>
<td>Inter-team presentation and mutual testing of prototype II</td>
<td>all</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Final debugging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19.07.2016</td>
<td>Final presentation</td>
<td>all</td>
<td>Final implementation</td>
</tr>
<tr>
<td></td>
<td>TBD Project closing</td>
<td></td>
<td>Final documentation</td>
</tr>
</tbody>
</table>
All deliverables **per team**

Deliverables for intermediate stages

- **Lightweight**, but shall enable progress tracking
- May be done in github wiki

Final deliverables

- Code base, packaged for easy deployment
- Final project report
Short round

- Why this project?
- Workload this semester
- Process background (BPMN)
- Event background (query languages)
- System design/modelling (UML, ER)
- Open source licenses
- Technologies
  - Languages (Java, JS)
  - Version control and build frameworks (GIT, Maven)
  - API usage (SOAP, REST)
  - File formats (XML, XSD)
- Project experience
Each team comprises 2-3 members working on one specific engine.

**Synchronization** of teams for common design decisions:

- Exchange format and data structures for process models
  - Classification of model elements into consumers/producers
  - Modeling syntax vs. execution semantics
- Exchange format and data structure for event types
  - Simple event types defined by simple data attributes
  - Complex event types defined by CEP queries
Who is in?

What are the groups (Camunda, Processmaker, Bonitasoft)?

Please commit to this project by the end of this week (Friday 22nd) and

• Send your github user name (hupdagit@gmail.com)
• Register in Goya

Further questions?