MTAT.03.231 Business Process Management

Lecture 10
Process Automation

Marlon Dumas
Professor of Information Systems @ University of Tartu
Co-Founder of Apromore
Where are we?
From “conceptual” to executable models

Conceptual “to-be” process models
- are made by domain experts
- provide a basis for communication amongst relevant stakeholders
- must be understandable
- must be intuitive and may leave room for interpretation
- contain purely a relevant set of process information

Executable process models
- are made by IT specialists
- provide input to a process enactment system - BPMS
- must be machine readable
- must be unambiguous and should not contain any uncertainties
- contain further details that are only relevant to implementation

“to-be executed” process model
Bridging the gap: A five-step method

1. Identify the automation boundaries
2. Review manual tasks
3. Complete the process model
4. Adjust task granularity
5. Specify execution properties

Adapted from teaching material of Remco Dijkman, TU/e.
Running example
running example
1. Identify the automation boundaries

**Principle:** not all parts of a process can be automated.

-> Start by identifying each task’s type:

1. Automated tasks
2. User tasks
3. Manual tasks
In BPMN: specify task markers

- Automated tasks
- User task
- Manual task

- Service task
- Script task
- User task
- Manual task
- Send task
- Receive task
In our example...
2. Review manual tasks

*Principle*: if it can’t be seen by the BPMS, it doesn’t exist.

• Find ways to support manual tasks via IT:
  - via user task
  - via automated task

• Or leave them out of the automated process.
3. Complete the process model

**Principle 1:** exceptions are the rule.
- Consider incomplete paths
- Rules of thumb
  - If we send something to another party, what happens if they do not respond? What happens if the response comes late? What happens if they do not respond the way we expect?
  - For each task: Can it go wrong and what happens if it goes wrong?
  - For each external party: Have we captured all messages or queries they might send us? (use CRUD)

**Principle:** no data = no decisions, no tasks handover.
- Specify all (electronic) business objects
- For each task, determine which business objects it creates, reads, updates, delete (CRUD)
- For each decision, determine which objects it needs
In our example...
4. Adjust task granularity

The goal is to coordinate handovers of work between resources, therefore:

- Aggregate consecutive tasks assigned to the same performer
- Split tasks if they require different performers
An exception to the rule
Our example...
Bridging the gap: one task at a time

1. Identify the automation boundaries
2. Review manual tasks
3. Complete the process model
4. Adjust task granularity
5. Specify execution properties
5. Specify execution properties

- Process variables, messages, signals, errors
- Task and event variables and their mappings to (global) process variables
- Calls to external systems (via service interfaces/API endpoints)
- Code snippets
- Participant assignment rules
- User interface specification (e.g. forms)
- Task, event and sequence flow expressions
Business Process Management System

BPMS

- Process modeling tool

Process model repository

- Administration & monitoring tools

Execution engine

- Execution logs

Worklist handler

External services
Process modeling tool

- To create and modify executable process models (by specifying execution properties)
- To store and retrieve automation solutions from a process model repository
Example: Process model editor
Execution Engine

- Instantiates executable process models (also called “cases”)
- Orchestrates distribution of work items to process participants and software services in order to execute a business process from start to end
- Logs execution data
Worklist Handler

• Imagine it as an “inbox”
• Offers work items to workers and allows them to “pick” these work items
• Handles work queues and priorities
Example worklist handlers
Administration & Monitoring Tools

- To manage automation solutions
- To configure access to system components
- To monitor participants availability and performance of process cases
External Services / API endpoints

• Expose a service interface with which the engine can interact
• The engine provides the invoked service with the necessary data it will need to perform the activity for a specific case
• Examples: rules engine, email or Twitter notification, DB connector, CRM connector…
## BPMS Landscape

<table>
<thead>
<tr>
<th>Closed source</th>
<th>Open-source</th>
</tr>
</thead>
<tbody>
<tr>
<td>• BIC Platform</td>
<td>• Bonita Platform</td>
</tr>
<tr>
<td>• BizAgi</td>
<td>• Camunda</td>
</tr>
<tr>
<td>• IBM BPM</td>
<td>• Flowable</td>
</tr>
<tr>
<td>• Oracle BPMS</td>
<td>• JBoss jBPM</td>
</tr>
<tr>
<td>• Pega Platform</td>
<td></td>
</tr>
<tr>
<td>• webMethods</td>
<td></td>
</tr>
</tbody>
</table>
Camunda Architecture

- **Business Analyst / Developer**
  - Modeler
  - Tasklist
  - Custom Application
    - REST
    - REST / Java API
      - Engine (BPMN, CMMN, DMN)
        - Database

- **End User**
- **Operator (technical/business)**
  - Cockpit
  - REST
  - REST

- **Administrator**
  - Admin
  - REST
Cheat sheet

1. Control flow  >  specify sequence flow expressions...
2. Data flow    >  specify data types (or *data fields*) and data mappings
3. Resources    >  specify participants assignment rules, service calls...

- ERP
  - Senior Finance Officer
  - Finance Department
    - Check Invoice
    - Mismatches
    - Enter Invoice Details
      - mismatch exists
      - no mismatches
    - Block Invoice
      - Invoice received
      - Invoice posted
      - Post Invoice
      - Invoice blocked

- Invoice InvoiceReport
  - Invoice Invoice DB