From Conceptual to Executable BPMN Process Models
A Step-by-Step Method

Marlon Dumas
University of Tartu
marlon.dumas@ut.ee

Slides prepared in collaboration with Marcello La Rosa (QUT)
With contributions from Remco Dijkman (TU/e)
Where are we?

Process identification

Process architecture

Process discovery

As-is process model

Process analysis

Insights on weaknesses and their impact

Process redesign

To-be process model

Process monitoring and controlling

Executable process model

Process implementation

Conformance and performance insights

To be completed

Process architecture

The image describes a process improvement framework, consisting of several stages: Process identification, Process architecture, Process discovery, Process monitoring and controlling, As-is process model, Process analysis, Process redesign, and Process implementation. Each stage is connected, forming a cycle that represents the continuous improvement process.
The business-engineering gap...

Process discovery

Process identification

Process analysis

Process implementation

Process monitoring and controlling

Process redesign

The business-engineering gap...

Executable process model

To-be process model
Two sides of the BPM story

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 experts
- 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

- Service task
- Script task
- User task
- Manual task

Automated tasks

User task

Manual 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

-> Isolate them and automate the rest
Alternative: isolate manual tasks
Alternative: isolate manual tasks

Segment 1

1. Application submitted electronically
2. Check application for completeness
3. Verify English language test
4. Batch application to admissions committee
5. Application batched
6. Application rejected

Segment 2

1. Assess application
2. Meeting day
3. Meeting completed

Segment 3

1. Update student record
2. Application approved
3. Accept application
4. Application accepted
5. Application rejected
6. Application result available
7. Reject application
8. Application not approved
Prescription fulfillment process:
• Once the prescription passes the insurance check, it is assigned to a technician who collects the drugs from the shelves and puts them in a bag with the prescription stapled to it.
• After that, the bag is passed to the pharmacist who double-checks that the prescription has been filled correctly.
• After this quality check, the pharmacist seals the bag and puts it in the pick-up area.
• When a customer arrives to pick up their prescription, a technician retrieves the prescription and asks the customer for their payment.

Assume the pharmacy system automates this process. Identify the type of each task and link manual tasks to the system.
3. Complete the process model

(cf. Last week’s lecture)

**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

**Principle**: BPMSs add value if they coordinate handovers of work between resources.

- Aggregate any two consecutive tasks assigned to the same performer
- Split tasks if they require different performers
An exception to the rule
Our example...

After Step 4
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 process variables
-> Service details
-> Code snippets
-> Participant assignment rules and user interface structure
-> Task, event and sequence flow expressions
-> BPMS-specific: work queues, forms, connectors...
Business Process Management System
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
• May import from conceptual process modeling tools
Example process modeling tools
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 process participants and allows participants to commit to these work items
- Handles participants’ work queues and work item priorities
- May provide social network capabilities
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
Example monitoring & administration tools
External Services

• 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...

![Diagram of BPMS and external services](image-url)
Example external services

Bosch Visual Rules editor
## BPMS Landscape

<table>
<thead>
<tr>
<th>Big vendors</th>
<th>Other closed-source</th>
<th>Commercial open-source</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM BPM</td>
<td>Appian BPMS</td>
<td>Bonita Open Solution</td>
</tr>
<tr>
<td>Oracle BPMS</td>
<td>BizAgi BPM Suite</td>
<td>Camunda Fox</td>
</tr>
<tr>
<td>Microsoft BizTalk, Wf</td>
<td>Bosch inubit Suite</td>
<td>Intalio</td>
</tr>
<tr>
<td>SAP NetWeaver BPM</td>
<td>OpenText BPM</td>
<td>JBoss jBPM</td>
</tr>
<tr>
<td>Software AG webMethods</td>
<td>Perceptive BPMONE</td>
<td>Shark</td>
</tr>
<tr>
<td>Pagaystems PegaRULES</td>
<td>Progress Savvion (cloud)</td>
<td>YAWL</td>
</tr>
</tbody>
</table>
Cheat sheet

1. Control flow > specify sequence flow expressions...
2. Data flow > specify data types and data mappings
3. Resources > specify participants assignment rules, service details...
1. Model Process

New Process
Create a new process using Bizagi Process Modeler.
Learn More...

Edit Process
Edit the business process you have already created.
Learn More...

Import Process
Have you already got models? Import them into Bizagi Studio.
Learn More...

Import Templates
Install and customize ready-to-use templates.
Learn More...