LTAT.05.015

Business Process Mining

Lecture 4: Automated Process Discovery

Marlon Dumas

Professor of Information Systems @ University of Tartu

Co-founder @ Apromore
Course Outline

Introduction

Process performance measurement & dashboards

Process mining
  - Automated process discovery
  - Conformance checking
  - Performance mining
  - Variant analysis

Simulation and “what-if” process mining

Data extraction & preparation for process mining

Predictive process monitoring

Business case analysis & project management for process mining

Trends: prescriptive monitoring, causal process mining, robotic process mining
Process Mining

Automated Process Discovery

Performance Mining

Variants Analysis

Conformance Checking

Enhanced process model

discovered process model

event log

event log’

Business rules / normative model

Difference diagnostics

✓/×
Automated Process Discovery

Process Map
(directly follows graph)

<table>
<thead>
<tr>
<th>CID</th>
<th>Task</th>
<th>Time Stamp</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>13219</td>
<td>Enter Loan Application</td>
<td>2007-11-09 T 11:20:10</td>
<td></td>
</tr>
<tr>
<td>13219</td>
<td>Retrieve Applicant Data</td>
<td>2007-11-09 T 11:22:15</td>
<td></td>
</tr>
<tr>
<td>13220</td>
<td>Enter Loan Application</td>
<td>2007-11-09 T 11:22:40</td>
<td></td>
</tr>
<tr>
<td>13219</td>
<td>Compute Installments</td>
<td>2007-11-09 T 11:22:45</td>
<td></td>
</tr>
<tr>
<td>13219</td>
<td>Notify Eligibility</td>
<td>2007-11-09 T 11:23:00</td>
<td></td>
</tr>
<tr>
<td>13219</td>
<td>Approve Simple Application</td>
<td>2007-11-09 T 11:24:30</td>
<td></td>
</tr>
<tr>
<td>13220</td>
<td>Compute Installments</td>
<td>2007-11-09 T 11:24:35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
A process map of an event log is a graph where:

- Each activity is represented by one node
- An arc from activity A to activity B means that B is **directly followed** by A in at least one case in the log
**Anatomy of a Process map**

- **Initial activity** (multiple possible)
- **Activity (and its frequency)**
- **Eventually-folllows relation between two activities (not visualized)**
- **Activity self-loop (rework)**
- **Short loop (possible rework)**
- **Final activity (multiple possible)**

Diagram with nodes and edges illustrating the process flow and relationships between activities. Each node represents an activity, and the edges show the flow and relationships. Numbers next to the nodes indicate the frequency or weight of that activity or relation. The diagram includes direct and eventually-follows relations between activities, with some nodes having self-loops indicating rework.
Nodes in a process map can be coloured, and arcs’ thickness can be changed, to capture:

- **Frequency**: How often a given activity or directly-follows relation occurs?
- **Duration**: processing times and cycle times for activities, waiting times for directly-follow relations
- **Other attributes**: some tools support enhancement by other attributes, e.g. cost, revenue, sales volume etc., if these data attributes are available
Example: Process map with frequency

Event log
5: a, b, c, g, e, h
4: a, b, c, f, g, h
3: a, b, d, g, e, h
3: a, b, d, e, g, h
Example: Process map with frequency

Event log
5: a,b,c,g,e,h
4: a,b,c,f,g,h
3: a,b,d,g,e,h
3: a,b,d,e,g,h
Example: Process map
with frequency

Event log
5: a,b,c,g,e,h
4: a,b,c,f,g,h
3: a,b,d,g,e,h
3: a,b,d,e,g,h

```
Example: Process map
with frequency

Event log
5: a,b,c,g,e,h
4: a,b,c,f,g,h
3: a,b,d,g,e,h
3: a,b,d,e,g,h
```
Example: Process map with frequency

Event log

5: a,b,c,g,e,h
4: a,b,c,f,g,h
3: a,b,d,g,e,h
3: a,b,d,e,g,h
Example: Process map with frequency

Event log

5: a, b, c, g, e, h
4: a, b, c, f, g, h
3: a, b, d, g, e, h
3: a, b, d, e, g, h
## Exercise: Process map

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Task Name</th>
<th>Originator</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>File Fine</td>
<td>Anne</td>
<td>20-07-2004 14:00:00</td>
</tr>
<tr>
<td>2</td>
<td>File Fine</td>
<td>Anne</td>
<td>20-07-2004 15:00:00</td>
</tr>
<tr>
<td>1</td>
<td>Send Bill</td>
<td>system</td>
<td>20-07-2004 15:05:00</td>
</tr>
<tr>
<td>2</td>
<td>Send Bill</td>
<td>system</td>
<td>20-07-2004 15:07:00</td>
</tr>
<tr>
<td>3</td>
<td>File Fine</td>
<td>Anne</td>
<td>21-07-2004 10:00:00</td>
</tr>
<tr>
<td>3</td>
<td>Send Bill</td>
<td>system</td>
<td>21-07-2004 14:00:00</td>
</tr>
<tr>
<td>4</td>
<td>File Fine</td>
<td>Anne</td>
<td>22-07-2004 11:00:00</td>
</tr>
<tr>
<td>4</td>
<td>Send Bill</td>
<td>system</td>
<td>22-07-2004 11:10:00</td>
</tr>
<tr>
<td>1</td>
<td>Process Payment</td>
<td>system</td>
<td>24-07-2004 15:05:00</td>
</tr>
<tr>
<td>1</td>
<td>Close Case</td>
<td>system</td>
<td>24-07-2004 15:06:00</td>
</tr>
<tr>
<td>2</td>
<td>Reminder</td>
<td>Mary</td>
<td>20-08-2004 10:00:00</td>
</tr>
<tr>
<td>3</td>
<td>Reminder</td>
<td>John</td>
<td>21-08-2004 10:00:00</td>
</tr>
<tr>
<td>2</td>
<td>Process Payment</td>
<td>system</td>
<td>22-08-2004 09:05:00</td>
</tr>
<tr>
<td>2</td>
<td>Close case</td>
<td>system</td>
<td>22-08-2004 09:06:00</td>
</tr>
<tr>
<td>4</td>
<td>Reminder</td>
<td>John</td>
<td>22-08-2004 15:10:00</td>
</tr>
<tr>
<td>4</td>
<td>Reminder</td>
<td>Mary</td>
<td>22-08-2004 17:10:00</td>
</tr>
<tr>
<td>4</td>
<td>Process Payment</td>
<td>system</td>
<td>29-08-2004 14:01:00</td>
</tr>
<tr>
<td>4</td>
<td>Close Case</td>
<td>system</td>
<td>29-08-2004 17:30:00</td>
</tr>
<tr>
<td>3</td>
<td>Reminder</td>
<td>John</td>
<td>21-09-2004 10:00:00</td>
</tr>
<tr>
<td>3</td>
<td>Reminder</td>
<td>John</td>
<td>21-10-2004 10:00:00</td>
</tr>
<tr>
<td>3</td>
<td>Process Payment</td>
<td>system</td>
<td>25-10-2004 14:00:00</td>
</tr>
<tr>
<td>3</td>
<td>Close Case</td>
<td>system</td>
<td>25-10-2004 14:01:00</td>
</tr>
</tbody>
</table>
Case 1: File fine, Send bill, Process payment, Close case
Case 2: File fine, Send bill, Reminder, Process payment, Close case
Case 3: File fine, Send bill, Reminder, Reminder, Reminder, Process payment, Close case
Case 4: File fine, Send bill, Reminder, Reminder, Process payment, Close case
Limitations of process maps

Process maps over-generalize: some paths of a process map might actually not exist and might not make sense

- Example: Draw the process map of [ abc, adc, afce, afec ] and check which cases it can recognize for which there is no support in the event log.

Process maps make it difficult to distinguish conditional branching, parallelism and loops

- See previous example… or a simpler one: [ abcd, acbd ]

Solution: automated discovery of BPMN models
Discovery of BPMN models

A given configuration of node/arc sliders, and parallelism slider

Note: Apromore uses an algorithm called Split Miner to turn process maps into process models
The BPMN process modelling notation

Based on flowcharts:
- Core set of notation elements
- Each core element has various subtypes

A BPMN process model is a graph consisting of four types of core elements:

- activity
- start
- event
- end
- gateway
- sequence
- flow
An XOR Gateway captures decision points (XOR-split) and points where alternative flows are merged (XOR-join).

**XOR-split** ➔ takes **one** outgoing branch

**XOR-join** ➔ proceeds when **one** incoming branch has completed.
Example: XOR Gateway

Invoice checking process
An AND Gateway provides a mechanism to create and synchronize “parallel” flows.

- **AND-split** → takes all outgoing branches

- **AND-join** → proceeds when all incoming branches have completed
Example: AND Gateway

Airport security check

Diagram:
- Boarding pass received
- Proceed to security check
- Pass security screening
- Proceed to departure level
- Pass luggage screening
- Departure level reached
A company has two warehouses that store different products: Amsterdam and Hamburg.

When an order is received, it is distributed across these warehouses: if some of the relevant products are maintained in Amsterdam, a sub-order is sent there. Likewise, if some relevant products are maintained in Hamburg, a sub-order is sent there. Afterwards, the order is registered and the process completes.
Solution 1

Order distribution process
Solution 2

Order distribution process
An OR Gateway provides a mechanism to create and synchronize \( n \) out of \( m \) parallel flows.

- **OR-split** \( \rightarrow \) takes one or more branches depending on conditions

- **OR-join** \( \rightarrow \) proceeds when all **active** incoming branches have completed

\[ \text{cond}_1 \quad \text{cond}_n \]
Example: OR Gateway

Order distribution process
What join type do we need here?
In the minister's office, when a ministerial inquiry has been received, it is registered into the system. Then the inquiry is investigated so that a ministerial response can be prepared.

The finalization of a response includes the preparation of the response itself by the cabinet officer and the review of the response by the principal registrar. If the registrar does not approve the response, the latter needs to be prepared again by the cabinet officer for review. The process finishes only once the response has been approved.
Abstraction and Filtering

• Real-life processes are really messy
  • Let’s have a look at a few of them…

• Process mining tools use process maps as the main visualization technique for event logs

• These tools also provide three types of operations:
  1. **Abstract** the process map:
  2. **Filter** the traces in the event log…
Examples

Patient Treatment Process @ Hospital
(Sepsis infections)

IT Incident Management @ Bank
Abstraction and Filtering

• Real-life processes are really messy
  • Let’s have a look at a few of them…
Common process mining tools provide abstraction capabilities on top of process maps or process models, to:

- Show only **most frequent** activities
- Show only **most frequent** arcs

Some tools offer further abstraction capabilities to:

- Show only **least frequent** activities
- Show only **least frequent** arcs
- Show only **fastest/slowest** activities
- Show only **fastest/slowest** arcs
Filtering: Case filters

Case variant filter
Retain/remove cases that follow a certain sequence of activities

Case ID filter
Retain/remove cases with a specific ID

Attribute filter
Retain/remove cases that fulfil a given condition:
• Single dimension: e.g. all cases that have an event “Amend purchase order”
• Two-dimensional: e.g. all cases where activity “Amend purchase order” has been performed by Luke or Jane

Timeframe filter
Retain/remove cases that are active/contained/start or end in a given timeframe (e.g. from 1 Jan to 31 July 2019)
Filtering: Case filters

Performance filter

• Retain/remove cases whose performance measure (e.g. case duration, avg waiting time) is above or below a given value (e.g. all cases that last more than 5 days)

• Retain/remove cases where a given node or arc has a given duration range (e.g. all cases whose activity “Create invoice” taken between 10 and 20 min)

Path filter

• Retain/remove cases where there is a pair of events that fulfils a given condition (e.g. “Create invoice” directly or eventually followed by “Approve invoice”). The condition can also involve attributes and time intervals

• Retain/remove cases that start with or finish with an event that fulfils a given condition (e.g. all traces that start with “Purchase order received” and finish with “Order cancelled”)

Rework & repetition filter

• Retain/remove cases where certain activities are performed a given number of times (e.g. keep all cases where activity “Amend purchase order” is done at least twice).
Process maps: Event filters

**Attribute filter**
Retain/remove events that fulfil a given condition (e.g. all events that have a label “Amend purchase requisition” and “Amend request for quotation”)

**Timeframe filter**
Retain/remove events that are contained in a given timeframe (e.g. from 1 Jan to 31 July 2019)
## Automated Process Discovery: Analysis Template

<table>
<thead>
<tr>
<th>What?</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Flow Analysis</strong></td>
<td>Visualize the most frequent case variant(s) using the Case inspector</td>
</tr>
<tr>
<td>1. Analyze the process structure &amp; main</td>
<td>Use abstraction sliders to focus on the most frequent activities and</td>
</tr>
<tr>
<td>case variants</td>
<td>dependencies.</td>
</tr>
<tr>
<td>2. Identify parallelism, branching points &amp;</td>
<td>Switch to BPMN view; inspect the behavior around the gateways</td>
</tr>
<tr>
<td>rework loops</td>
<td>Check the dotted lines in the process map (arcs emanating from the</td>
</tr>
<tr>
<td>3. Analyze case entry and exit points, and</td>
<td>start event or leading to the end event). Unexpected dotted lines</td>
</tr>
<tr>
<td>check for incomplete cases</td>
<td>indicate some cases are incomplete</td>
</tr>
<tr>
<td></td>
<td><strong>Filtered Flow Analysis</strong></td>
</tr>
<tr>
<td></td>
<td>Use event filtering to retain/remove subsets of activities, e.g.</td>
</tr>
<tr>
<td></td>
<td>separate automated or non-core activities, or separate activities</td>
</tr>
<tr>
<td></td>
<td>belonging to two different systems.</td>
</tr>
<tr>
<td><strong>Frequency analysis</strong></td>
<td>Use the Activity inspector</td>
</tr>
<tr>
<td>Analyze the most frequent activities and</td>
<td>Use the color-coding to find most frequent arcs</td>
</tr>
<tr>
<td>relations</td>
<td>Consider switching between frequency metrics (max, avg, ...)</td>
</tr>
<tr>
<td></td>
<td>Use the Activities tab in performance dashboard (Apromore EE)</td>
</tr>
<tr>
<td><strong>Handoff analysis</strong></td>
<td>Switch between perspectives</td>
</tr>
<tr>
<td>Analyze handoffs between workers, teams,</td>
<td>Use the Resource and Other Attributes tabs in the Dashboard</td>
</tr>
<tr>
<td>groups, org units</td>
<td></td>
</tr>
</tbody>
</table>
Flow Analysis

In the Process Discoverer, you can visualize the most frequent case variants using the case inspector. You can also view the activities performed in a single case.

1. Analyze the process structure and main case variants

You can also download the case details in a CSV file format.
Flow Analysis

Apromore allows us to abstract a process map by “Case frequency” or “Average Duration”.

1. Analyze the process structure and main case variants

We can adjust the complexity of the discovered map by increasing and/or decreasing the frequency or duration of nodes and arcs that are visualized in our process map.

If we abstract by “Case Frequency” and shift the arcs or nodes slider towards the left, more edges/nodes with low case frequency will be removed from the process map.

If we abstract by “Average Duration” and shift the arcs or nodes slider towards the left, more edges/nodes with low average duration will be removed from the process map.
Flow Analysis

In some processes, certain tasks are repeated – this can be done in form of direct repetitions or through the “back and forth” between two or more activities.

2. Identify parallelism, branching points and rework loops

- It seems there is an indirect repetition between “Inform User” and “Test Repair”.
- It seems there is an indirect repetition between “Inform User” and “Repair Complex”.
- It appears that “Archive Repair” repeats. Is it really the case?
Flow Analysis

BPMN models allows us to separate parallelism from repetition.

2 Identify parallelism, branching points and rework loops

The level of the displayed parallelism can be adjusted via the abstraction settings.

This enables an analysis of complex processes and ensures you can always adjust the level of abstraction that is necessary for your objective.
Flow Analysis

Apromore allows us to identify incomplete cases by visualizing dotted arcs leading to an end event in a process map.

3. Analyze case exit points and check for incomplete cases.

“Inform Users”, “Repair (Complex)” and “Test Repair” are the activities involved in incomplete cases since there are dotted arcs that directly lead to the end event.
Filtered Flow Analysis

In the Process Discoverer you can also apply filters to analyze the results more precisely and to limit the number of the cases. There are a variety of filtering options available to you.

1. Analyze different components or slices of the process separately

We can make use of the “Case Variant” filter to retain/remove the case variants based on a particular condition.

For example, we want to retain up to 20 case variants to analyze the most common behavior of the process.
Filtered Flow Analysis

In the Process Discoverer, we can also apply filters to analyze the results more precisely and to limit the number of the cases. There are a variety of filtering options.

1. **Analyze different components or slices of the process separately**

The “Path” filter can be used to identify bottlenecks or to analyze the transitions between the activities performed.

For example, click on the “Path” filter to retain those cases where the activity “Inform User” eventually follows “Repair (Complex)”.
Filtered Flow Analysis

In the Process Discoverer you can also apply filters to analyze the results more precisely and to limit the number of the cases. There are a variety of filtering options available to you.

In order to understand a process, we might want to identify outliers. We can use the “Performance” filter for this purpose.

For example, retain only those cases that gets completed in at most 1 hour.
Filtered Flow Analysis

Process discovery is often used in conjunction with filters to examine different “slices” of the business process.

1. Analyze different components or slices of the process separately

We can also create a filter based on the value of a particular attribute.

In order to achieve this we can choose the attribute from the drop-down list of the “Attribute” filter from the “Event” tab.

For example, in order to remove those events where the value of “Resource” is “Solver S1” and “Solver S2”.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>SolverC1</td>
<td>534</td>
<td>3.448%</td>
</tr>
<tr>
<td>SolverC2</td>
<td>514</td>
<td>3.319%</td>
</tr>
<tr>
<td>SolverC3</td>
<td>402</td>
<td>2.590%</td>
</tr>
<tr>
<td>SolverS1</td>
<td>502</td>
<td>3.829%</td>
</tr>
<tr>
<td>SolverS2</td>
<td>438</td>
<td>3.210%</td>
</tr>
<tr>
<td>SolverS3</td>
<td>460</td>
<td>3.17%</td>
</tr>
<tr>
<td>System</td>
<td>7242</td>
<td>46.107%</td>
</tr>
<tr>
<td>Tester1</td>
<td>902</td>
<td>5.429%</td>
</tr>
<tr>
<td>Tester2</td>
<td>904</td>
<td>5.439%</td>
</tr>
<tr>
<td>Tester3</td>
<td>510</td>
<td>3.17%</td>
</tr>
<tr>
<td>Tester4</td>
<td>788</td>
<td>5.069%</td>
</tr>
<tr>
<td>Tester5</td>
<td>644</td>
<td>3.95%</td>
</tr>
<tr>
<td>Tester6</td>
<td>876</td>
<td>5.527%</td>
</tr>
</tbody>
</table>
Frequency Analysis

Use the color-coding to find the most frequent arcs.

1. Analyze the most frequent activities and relations

Time performance statistics are visualized via labels on activities and arcs, as well as via colours and line thickness (on a red scale) for activities and arcs.

The transition between “Analyze Defect” and “Inform User” takes the maximum amount of time.
Frequency Analysis

Apromore can visualize the process map by using color-coding techniques to find the most frequent arcs.

1. Analyze the most frequent activities and relations

By selecting the “max” option from the “Frequency” drop-down menu displays the maximum time an activity takes place.

The “Repair (Complex)” activity took the maximum amount of time.
Frequency Analysis

Apromore can visualize the process map by using color-coding techniques to find the most frequent arcs.

1. Analyze the most frequent activities and relations

By selecting the “min” option from the “Frequency” dropdown menu displays the maximum time an activity takes place.

The “Repair (Simple)”, “Analyze Defect” and “Test Repair” activities takes the minimum amount of time.
Handoff Analysis

Poor handovers can reduce, among other things, the efficiency of processes. Delays due to the pure transfer between process steps can be identified with the handoff analysis.

As soon as a process is executed by several employees or departments, there may be waiting times due to the transfer between process steps and resources. This can increase the cycle time of the entire process and therefore create inefficiencies.

Potential reasons for long handoffs:

- Inadequate handoff / missing information
- No notification that a task is waiting
- Too many open tasks
Handoff Analysis

Switch between perspectives to analyze handoffs

1. Analyze handoffs between workers, teams, groups and organizational units

By changing the perspective to “Resource”, we can analyse the handoff between resources.
Handoff Analysis

The Resource tab in the Performance Dashboard can also be used to analyze the handoff.

1. **Analyze handoffs between workers, teams, groups and organizational units**

Time taken by resources in different parts of our process can be analysed by clicking on the “Resources” perspective in the performance dashboard.
Handoff Analysis

Attribute filters (with secondary criterion) and Path filters can be used to deepen the handoff analysis.

1. **Analyze handoffs between workers, teams, groups and organizational units**

Handoff analysis between resources can also be identified by using a “Path” filter where “Repair (Complex)” and “Test Repair” are performed by different resources.