Reshaping ticket based services with process mining

Tamás Molnár
Vodafone Group BI
Contents

• Motivation for doing process mining

• The theory
  – What is process mining
  – Models and reality
  – The sketches of some common algorithms

• The experiences
  – The logs used
  – Mined process maps and interpretations
  – Practical issues
  – Some sort of insights found
Business motivation

There is a Vodafone organisational unit „X” which provides internal services to other VF units.

Individual issues are solved. They are represented by tickets in the event management system.

A small but still measurable percentage of the tickets takes too much time to solve and breaches some SLAs (Service Level Agreements).

X always strives for improving its services, so wished to reduce the number of the breaches.

X wanted to see how complicated its processes are and started 6σ projects.

As a first overview, tickets for two different local markets were analysed using a process mining approach.
Optional activities

Time waste!

Frequent path

Alternative paths

The event flow from the start to the end

The fruit of the work: a process map with mean execution times
What is process mining about?

"Process mining is a process management technique that allows for the analysis of business processes based on event logs. During process mining, specialized algorithms are applied to event log datasets in order to identify trends, patterns and details contained in event logs recorded by an information system. Process mining aims to improve process efficiency and understanding of processes."

Table 1.1 A fragment of some event log: each line corresponds to an event

<table>
<thead>
<tr>
<th>Case id</th>
<th>Event id</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Timestamp</td>
<td>Activity</td>
</tr>
<tr>
<td>1</td>
<td>35654423</td>
<td>30-12-2010:11:02</td>
</tr>
<tr>
<td></td>
<td>35654424</td>
<td>31-12-2010:10:06</td>
</tr>
<tr>
<td></td>
<td>35654425</td>
<td>05-01-2011:15:12</td>
</tr>
<tr>
<td></td>
<td>35654426</td>
<td>06-01-2011:11:18</td>
</tr>
<tr>
<td></td>
<td>35654427</td>
<td>07-01-2011:14:24</td>
</tr>
<tr>
<td>2</td>
<td>35654483</td>
<td>30-12-2010:11:32</td>
</tr>
<tr>
<td></td>
<td>35654485</td>
<td>30-12-2010:12:12</td>
</tr>
<tr>
<td></td>
<td>35654487</td>
<td>30-12-2010:14:16</td>
</tr>
<tr>
<td></td>
<td>35654488</td>
<td>05-01-2011:11:22</td>
</tr>
<tr>
<td></td>
<td>35654489</td>
<td>08-01-2011:12:05</td>
</tr>
</tbody>
</table>

How does it model real event flows? – Let us replay a story!

The mined model representation. „Petri Net”

Initial state: start enabled

Register request executed – 2 branches follow – „AND”

1 and only 1 examination executed – „OR”
How does it model real event flows? – Replay continued.

„Ticket check” MUST be executed

„Pay compensation” – End of process instance

„Decide” executed

We have just replayed one trace version from many possible

1. Register request
2. Examine thoroughly
3. Check ticket
4. Decide
5. Pay compensation
Other uses of replay

Conformance analysis

How faithfully the real traces follow the process model? If there is a prescribed event flow, where do we deviate from that?

By replaying the logs one can gain some indicators of the model – real-world-log conformance.

For example, using the numbers of the tokens emerging during the replay of the log traces:

\[
\text{fitness} = \frac{1}{2} \left( 1 - \frac{\text{#missing tokens}}{\text{# consumed tokens}} \right) + \frac{1}{2} \left( 1 - \frac{\text{#remaining tokens}}{\text{# produced tokens}} \right)
\]
Belongs to data science? – A few subjective aspects

- Data based approach
- Discovery of hidden patterns
- Goal too much for humans – computer aid needed
- Sophisticated math discovery algorithms
- Well defined model performance indicators exist

- Rather descriptive
- Often very small data
- Model validity may be subject to many soft criteria / beliefs
- Model validity may be lost immediately when lessons are learned
How to create one?

• Several common algorithms (algorithm families) exist:
  
  – Alpha
  – Heuristic
  – Tree induction
  – Fuzzy
  – Integer programming based
  – Genetic approach
The Alpha algorithm

Define ordering relations for any activity pairs in a log $L$.

Define the global structure through local structures.

Find for all activity pairs their relationship, based on the log.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>#</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td>#</td>
</tr>
<tr>
<td>c</td>
<td>←</td>
<td>←</td>
<td>#</td>
<td>..</td>
</tr>
<tr>
<td>d</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
</tbody>
</table>

A > B
B directly follows A at least once

A → B
B directly follows A at least once, but A never follows B

A#B:
Neither of them follows the other one directly

A || B:
They can follow each other in both orders

(a) sequence pattern: $a \rightarrow b$

(b) XOR-split pattern: $a \rightarrow b$, $a \rightarrow c$, and $b \rightarrow c$

(c) XOR-join pattern: $a \rightarrow c$, $b \rightarrow c$, and $a \# b$

(d) AND-split pattern: $a \rightarrow b$, $a \rightarrow c$, and $b \mid c$

(e) AND-join pattern: $a \rightarrow c$, $b \rightarrow c$, and $a \mid b$
The Heuristic algorithm

Calculate an asymmetric activity dependency indicator

Closer to ±1, the stronger the relationship

Create dependency graph

A weighted graph, where the nodes are the activities, the edge weights are the dependency measures

Filter the dependency graph

By absolute number of succeeding each other and By the dependency indicator

Identify splits and joins by replaying the traces

Find optimal alignments
Determine activity binding strengths
Apply final filtering by frequency of input/output bindings

Number of successions

\(|a \rightarrow_L b|\)

Dependency indicator

\(|a \Rightarrow_L b| = \begin{cases} \frac{|a >_L b| - |b >_L a|}{|a >_L b| + |b >_L a| + 1} & \text{if } a \neq b \\ \frac{|a >_L a|}{|a >_L a| + 1} & \text{if } a = b \end{cases}\)
Inductive tree

Create the directly follows / eventually follows graphs weighted by occurrence numbers

Optionally filter the graph for infrequent behavior

Find some special cut in the graph, create partitions linked in some special way to each other

Repeat the process for the partitions until individual activities remain

Sequence

Exclusive choice

Loop

Parallel
Now let us return to our real case – the X department

Just to recall....

„X” provides internal services to other VF units

A small but still measurable percentage of the tickets takes too much time to solve and breaches some SLAs (Service Level Agreements)

X always strives for improving its services, so wished to reduce the number of the breaches

X wanted to see how complicated its processes are and started 6σ projects

As a first overview, tickets for two different local markets were analysed using a process mining approach
# A typical log from X

Events of the same type may follow each other.

Further possible events:
- Forwarded Internally Solution provided, Automatically completed

Observation: the number of event types is low, still there is an abundance of trace variants

<table>
<thead>
<tr>
<th>Event</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>09.08.2016 09:34:02</td>
</tr>
<tr>
<td>In Process1</td>
<td>09.08.2016 09:34:28</td>
</tr>
<tr>
<td>In Process2</td>
<td>09.08.2016 09:37:34</td>
</tr>
<tr>
<td>In Process3</td>
<td>09.08.2016 14:31:11</td>
</tr>
<tr>
<td>In Process4</td>
<td>09.08.2016 16:45:20</td>
</tr>
<tr>
<td>In Process5</td>
<td>10.08.2016 12:11:06</td>
</tr>
<tr>
<td>Forwarded Externally</td>
<td>10.08.2016 12:11:12</td>
</tr>
<tr>
<td>Answer Received</td>
<td>08.09.2016 12:31:32</td>
</tr>
<tr>
<td>Completed</td>
<td>19.10.2016 17:28:03</td>
</tr>
</tbody>
</table>
A mined process for some activity „Z”

The series of events starts with „Open”

Then:
- Always there is an in process step
- Perhaps there is a Forwarded internally (it may be skipped silently)
- The ordering of the In process and the Forwarding as arbitrary (they are parallel)

Then:
- A „Forwarded externally” event MAY follow
- Which is preceded in some number of the cases by a „Customer action”

Then:
- Some other events MAY come

Then:
- Process is completed

We have a map – has it been worth mining it?

Yes, because we have got ONE description of the typical REAL process behavior albeit there were 34 trace variants in the data
What is more ... two processes for the same activity Z!
Execution frequencies...
… and average execution times
The resolution process for the same issue type Z for another country is totally different.

For country B separate breach – insla processes were not mined. But it is not what makes the difference.
Resolution processes for other issues – lot of differences again
### Practical issues

#### Importance of cleaning

- Appropriate event data preparation was as influential to success than the way as the mining algorithm was used

#### Sensitivity to parameters

- The process graphs changed quite suddenly at some thresholds

#### Event sojourn times

- As there was only one (starting) time recorded for each event, sometimes it took consideration to correctly interpret elapsed times

#### Speed

- With the actual ticket numbers (max 1200 / issue type) speed was not an issue

#### Buy-in

- Although a map seems to be self-explanatory, business stakeholders need thorough explanations
Summary of the insights gained

- Presence of lots of orphan traces
- Many trace variants for one problem type
- Very long trace variants exist
- A few tickets are open for extremely long time
- Repetitions of activity blocks
- Ticket inflow rate is almost constant
- The typical process flows which are able to replay 97+% of the (not-orphan) cases
- Structural differences between the event flows for breaching and in-sla cases
- Event types taking long time for the sla breaching cases
- Reopens happen somewhat later for the breach cases than for the in-sla cases

Are all necessary standards in place? Are standards known and followed?

Are all necessary checkpoints in place?

Opportunity for simple workforce planning, considering special times during the year?

How far these are from beliefs and standards?

Where can it be the easiest and most profitable to change the processes?

Would it useful to forego it and check fast customer opinion in cases when it is risky that there will be a reopen, in order not to waste time?
I want to do it! – Well then you need ...

A website
• WWW.PROCESSMINING.ORG

A software
• PROM
• Celonis, Disco

A book
• van der Aalst, Wil M.P. Process mining

An online course
• Introduction to Process Mining with ProM (FutureLearn) (Joos Buijs)
• Process Mining: Data science in Action (Coursera) (van der Aalst)

Be aware!
• There are a lot of possible topics & issues not mentioned in this presentation!

A log
• Look around!

The author of the presentation does not claim that the here mentioned sources are the only available sources or the best ones among all possible sources. But they can be useful for sure.
Thank you!