PROCESS MINING: LEVERAGING EVENT DATA TO UNDERSTAND AND IMPROVE ORGANIZATIONS

Henrik Leopold
Han van der Aa

December 9, 2019
IEEE Big Data 2019, Los Angeles, USA
This tutorial consists of four main parts:

1. **Prerequisites:** What is a process? What is process mining?
2. **Process Discovery:** How can we discover processes from event logs?
3. **Conformance Checking:** How can we check whether an actual process matches the expected process?
4. **Demo & Outlook:** What tool support is available? What are the current trends in research?
PART 1: PREREQUISITES
WHAT IS A PROCESS?

“a collection of activities that take one or more kinds of input and create an output that is of value to the customer”  
[Hammer & Champy 1993]

• Processes are everywhere.

Car production  
Mortgage application  
Medical treatment
HOW TO CAPTURE A PROCESS?

• Processes are typically captured using *process models*.
• A process model essentially consists of a set of activities and execution constraints between them.
• There are several *process model notations*:

  - Event-driven Process Chains
  - Business Process Modeling and Notation
  - UML Activity Diagrams
  - Petri Nets
Petri nets are
- based on strong mathematical foundation.
- allow for formal analysis.
- are “hidden” in high-level notations such as EPCs and BPMN models.

Background
- Foundations developed by Carl Adam Petri in 1962.
- There are a variety of variants and extensions.
- Today: Modelling and analysis of business processes.
Process Mining: Leveraging Event Data to Understand and Improve Organizations, IEEE Big Data 2019
**Rules**

- Connections are directed.
- No connections between two places or two transitions.
- Places may hold zero or more tokens.
A transition is enabled if each of its input places contains at least one token.
An enabled transition can fire (i.e., it occurs).
When it fires it consumes a token from each input place and produces a token for each output place.
• In the new state, make_picture is enabled. It will fire, etc.
• In this way, a Petri net defines how the process can be executed.
• Petri nets allow us to capture typical process patterns.

Sequence

AND-Split

XOR-Split

AND-Join

XOR-Join
WHAT IS PROCESS MINING?
**WHAT IS PROCESS MINING?**

Business Processes

- Support
- Models / Represents

Process Model

- Discovery
- Conformance
- Performance

Software Systems

- Record

Event Log

Corporate Systems support Event Log models / represents Business Processes as they occur in the real world during their execution. This allows for the discovery of new business processes, the conformance checking of real executions, and the performance improvement of executing processes.
PART 2: PROCESS DISCOVERY
**Process Discovery: Goal**

**Recorded** process behavior

**Reflects actual** process behavior

### Event Log to Process Discovery to Process Model

**Event log** → **Process discovery** → **Process model**

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Activity ID</th>
<th>Activity</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>Register request</td>
<td>11.07.2018 – 13:53</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>Register request</td>
<td>11.07.2018 – 14:29</td>
</tr>
<tr>
<td>1</td>
<td>B</td>
<td>Examine thoroughly</td>
<td>11.07.2018 – 14:33</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>Register request</td>
<td>12.07.2018 – 08:45</td>
</tr>
<tr>
<td>1</td>
<td>D</td>
<td>Check ticket</td>
<td>12.07.2018 – 12:29</td>
</tr>
</tbody>
</table>

---

*Process Mining: Leveraging Event Data to Understand and Improve Organizations, IEEE Big Data 2019*
#### Software Systems

**Input: Event Log**

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Activity ID</th>
<th>Activity</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>Register request</td>
<td>11.07.2018 – 13:53</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>Register request</td>
<td>11.07.2018 – 14:29</td>
</tr>
<tr>
<td>1</td>
<td>B</td>
<td>Examine thoroughly</td>
<td>11.07.2018 – 14:33</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>Register request</td>
<td>12.07.2018 – 08:45</td>
</tr>
<tr>
<td>1</td>
<td>D</td>
<td>Check ticket</td>
<td>12.07.2018 – 12:29</td>
</tr>
<tr>
<td>2</td>
<td>D</td>
<td>Check ticket</td>
<td>12.07.2018 – 15:22</td>
</tr>
<tr>
<td>1</td>
<td>E</td>
<td>Decide</td>
<td>13.07.2018 – 15:22</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>Examine thoroughly</td>
<td>13.07.2018 – 19:18</td>
</tr>
<tr>
<td>1</td>
<td>G</td>
<td>Pay compensation</td>
<td>14.07.2018 – 08:15</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>Examine casually</td>
<td>14.07.2018 – 08:18</td>
</tr>
<tr>
<td>3</td>
<td>D</td>
<td>Check ticket</td>
<td>15.07.2018 – 07:21</td>
</tr>
<tr>
<td>3</td>
<td>E</td>
<td>Decide</td>
<td>15.07.2018 – 09:52</td>
</tr>
<tr>
<td>2</td>
<td>E</td>
<td>Decide</td>
<td>15.07.2018 – 10:01</td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>Reject request</td>
<td>16.07.2018 – 18:08</td>
</tr>
</tbody>
</table>

...
# FROM EVENT LOG TO TRACES

## Abstract to Sequences

<table>
<thead>
<tr>
<th>Trace - Case 1</th>
<th>Trace - Case 2</th>
<th>Trace - Case 3</th>
<th>Trace - Case N</th>
</tr>
</thead>
<tbody>
<tr>
<td>a Register request</td>
<td>a Register request</td>
<td>a Register request</td>
<td>a Register request</td>
</tr>
<tr>
<td>b Examine thoroughly</td>
<td>d Check ticket</td>
<td>b Examine thoroughly</td>
<td>c Examine casually</td>
</tr>
<tr>
<td>d Check ticket</td>
<td>c Examine casually</td>
<td>d Check ticket</td>
<td>d Check ticket</td>
</tr>
<tr>
<td>e Decide</td>
<td>e Decide</td>
<td>e Decide</td>
<td>e Decide</td>
</tr>
<tr>
<td>g Pay compensation</td>
<td>g Pay compensation</td>
<td>h Reject request</td>
<td>g Pay compensation</td>
</tr>
</tbody>
</table>

**Log L** = \[<a,b,d,e,g>^5, <a,d,c,e,g>^{10}, <a,b,d,e,h>^3, <a,c,d,e,h>^8, <a,c,d,e,f,d,b,e,g>^2] \]
Goal is to discover a model that:

1. Allows for behavior seen in log (*fitness*)
2. Is as simple as possible (*simplicity*)
3. Balances between overfitting and underfitting (*precision vs. generalization*)

### Model Quality

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABDEI</td>
<td>1207</td>
</tr>
<tr>
<td>ACDGHFI</td>
<td>145</td>
</tr>
<tr>
<td>ACGDHFI</td>
<td>56</td>
</tr>
<tr>
<td>ACGDHFI</td>
<td>56</td>
</tr>
<tr>
<td>ACHDFI</td>
<td>23</td>
</tr>
<tr>
<td>ACDHFI</td>
<td>28</td>
</tr>
</tbody>
</table>

Model A: Overfitting and complex

Model B: Underfitting
Key Idea of Discovery

Recognize patterns in the order relations between activities

Pattern 1: Sequence

... Activity A Activity B Activity A ...

Pattern 2: Concurrency

... Activity C Activity C ...
Activity D Activity E Activity E Activity D ...

Pattern 3: Exclusivity

... Activity F Activity F ...
Activity G Activity H Activity H Activity G ...

Process Mining: Leveraging Event Data to Understand and Improve Organizations, IEEE Big Data 2019
Every trace starts with “a”

\[ L_2 = \langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle \]
Every trace ends with “d”

$L_2 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle,$ $\langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle]$
"b" and "c" always occur together without a particular order

\[ L_2 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle] \]
Every “e” is followed by an “f”, every “f” is preceded by an “e”

\[ L_2 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle] \]
DISCOVERY EXAMPLE

\[ L_2 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle] \]
Every trace starts with “a”

\[ L_2 = [\langle a,b,c,d \rangle^3, \langle a,c,b,d \rangle^4, \langle a,b,c,e,f,b,c,d \rangle^2, \langle a,b,c,e,f,c,b,d \rangle, \langle a,c,b,e,f,b,c,d \rangle^2, \langle a,c,b,e,f,b,c,e,f,c,b,d \rangle] \]
Every trace ends with “d”

$$L_2 = \{ \langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle \}$$
“b” and “c” always occur together without a particular order

\[ L_2 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle] \]
Every “e” is followed by an “f”, every “f” is preceded by an “e”

\[ L_2 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \\
\langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle] \]
**DISCOVERY CHALLENGES**

- Data quality issues
- Balancing over- and underfitting
- Spaghetti models
**Discovery Algorithms**

Key differences:

- Representation format
- How to tackle challenges
- Unique pros and cons
- No silver bullet
PART 3: CONFORMANCE CHECKING
MOTIVATION

Various drivers for conformance assessment

- Corporate governance, risk, compliance, and legislation such as the Sarbanes-Oxley (US), Basel II/III (EU)
- ISO 9001:2008 requires organizations to model their operational processes
- Business alignment: make sure that the information systems and the real business processes are well aligned
Basic idea of conformance checking

Event Log

Local diagnostics

Global conformance measures

Process Model

Local diagnostics

Process Mining: Leveraging Event Data to Understand and Improve Organizations, IEEE Big Data 2019
Replaying trace “ABEG”
• Use four counters:
  – \( p \) = produced tokens
  – \( c \) = consumed tokens
  – \( m \) = missing tokens (consumed while not there)
  – \( r \) = remaining tokens (produced but not consumed)
Initialization and finalization:

- In the beginning a token is produced for the source place: \( p = 1 \)
- At the end a token is consumed from the sink place (also if not there): \( c' = c = 1 \).
Replay Example

\[
t = \langle a, d, c, e, h \rangle
\]

\[
\begin{align*}
\text{p} &= 1 \\
\text{c} &= 0 \\
\text{m} &= 0 \\
\text{r} &= 0
\end{align*}
\]
Replay Example (cont.)

\[ t = \langle a, d, c, e, h \rangle \]
\[ t = \langle a, d, c, e, h \rangle \]
**REPLAY EXAMPLE (CONT.)**

**t = <a, d, c, e, h>**

```
p = 4  
c = 3  
m = 1  
r = 0
```
t = <a, d, c, e, h>
t = <a, d, c, e, h>

p = 6

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>a</td>
</tr>
<tr>
<td></td>
<td>p1</td>
</tr>
<tr>
<td></td>
<td>c</td>
</tr>
<tr>
<td>m</td>
<td>d</td>
</tr>
<tr>
<td></td>
<td>e</td>
</tr>
<tr>
<td>p3</td>
<td>g</td>
</tr>
<tr>
<td></td>
<td>h</td>
</tr>
<tr>
<td></td>
<td>end</td>
</tr>
</tbody>
</table>

p = 6
c = 5
m = 1
r = 0
Replay Example (cont.)

\[ t = \langle a, d, c, e, h \rangle \]

\[
fitness(t, M) = \frac{1}{2} \left( 1 - \frac{1}{6} \right) + \frac{1}{2} \left( 1 - \frac{1}{6} \right) = 0.8333
\]
Diagnostics

Process Mining: Leveraging Event Data to Understand and Improve Organizations, IEEE Big Data 2019

\[ \text{fitness}(L, M) = 0.9504 \]
Main limitation of token replay is that it does not take a wholistic perspective.

Token replay is the basis for state-of-the-art alignment techniques.

\[ t = \langle a, c_1, c_2, e_1, e_2, e_3 \rangle \]
PART 4: DEMO & OUTLOOK
Process Mining Tools

Open source

- ProM6 (Java): [www.promtools.org](http://www.promtools.org)
- PM4Py (Python): [www.pm4py.org](http://www.pm4py.org)
- Many state-of-the-art algorithms
- Stand-alone, plugin architecture

Commercial

- User friendly
- Integrated analyses
- High performance
- Academic licenses available

Process Mining: Leveraging Event Data to Understand and Improve Organizations, IEEE Big Data 2019
DEMO
OUTLOOK

• Beyond control-flow analysis
• Dealing with data quality issues
• Privacy-aware process mining
• From reactive to predictive analysis
**CONCLUSION**

**Business Processes**

**Software Systems**

**support**

**models / represents**

**Process Model**

**Event Log**

**Discovery**

**Conformance**

**Performance**

**Event Log**
Questions?

Prof. Dr. Henrik Leopold
henrik.leopold@the-klu.org
www.henrikleopold.com

Dr. Han van der Aa
han.van.der.aa@hu-berlin.de
www.hanvanderaa.com