Business Process Intelligence Course
( Lecture 7 )

Operational Support and Applications of Process Mining

prof.dr.ir. Wil van der Aalst
www.processmining.org
## Overview

### Part I: Preliminaries

<table>
<thead>
<tr>
<th>Chapter 2</th>
<th>Chapter 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Modeling and Analysis</td>
<td>Data Mining</td>
</tr>
</tbody>
</table>

### Part II: From Event Logs to Process Models

<table>
<thead>
<tr>
<th>Chapter 4</th>
<th>Chapter 5</th>
<th>Chapter 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting the Data</td>
<td>Process Discovery: An Introduction</td>
<td>Advanced Process Discovery Techniques</td>
</tr>
</tbody>
</table>

### Part III: Beyond Process Discovery

<table>
<thead>
<tr>
<th>Chapter 7</th>
<th>Chapter 8</th>
<th>Chapter 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conformance Checking</td>
<td>Mining Additional Perspectives</td>
<td>Operational Support</td>
</tr>
</tbody>
</table>

### Part IV: Putting Process Mining to Work

<table>
<thead>
<tr>
<th>Chapter 10</th>
<th>Chapter 11</th>
<th>Chapter 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool Support</td>
<td>Analyzing &quot;Lasagna Processes&quot;</td>
<td>Analyzing &quot;Spaghetti Processes&quot;</td>
</tr>
</tbody>
</table>

### Part V: Reflection

<table>
<thead>
<tr>
<th>Chapter 13</th>
<th>Chapter 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cartography and Navigation</td>
<td>Epilogue</td>
</tr>
</tbody>
</table>
Process mining spectrum

"world" business processes
people machines components organizations
models analyzes

software system
records events, e.g., messages, transactions, etc.
supports/controls
specifies configures implements analyzes

(process) model
discovery conformance enhancement

event logs
refined process mining framework
Refined process mining framework

- People
- Machines
- Business processes
- Organizations
- Documents

“World”

Information system(s)

Event logs
- "Pre mortem"
- "Post mortem"

Current data

Historic data

Provenance

Navigation
- Explore
- Predict
- Recommend

Auditing
- Detect
- Check
- Compare
- Promote

Cartography
- Discover
- Enhance
- Diagnose

Models
- De jure models
  - Control-flow
  - Data/rules
  - Resources/organization

- De facto models
  - Control-flow
  - Data/rules
  - Resources/organization

©Wil van der Aalst TU/e (use only with permission & acknowledgements)
Business process provenance

people
machines
business processes
organizations

“world”

information system(s)

provenance

event logs

“pre mortem”
current data

“post mortem”
historic data

people
machines
business processes
organizations

“world”

information system(s)

provenance

event logs

“pre mortem”
current data

“post mortem”
historic data
Two types of event data: post and pre mortem

• “Post mortem” event data refer to information about cases that have completed, i.e., these data can be used for process improvement and auditing, but not for influencing the cases they refer to.

• “Pre mortem” event data refer to cases that have not yet completed. If a case is still running, i.e., the case is still “alive” (pre mortem), then it may be possible that information in the event log about this case (i.e., current data) can be exploited to ensure the correct or efficient handling of this case.
<table>
<thead>
<tr>
<th></th>
<th>student-related event data</th>
<th>sales-related event data</th>
<th>patient-related event data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>post mortem</strong></td>
<td>Understanding where and why student drop out or deviate. Should the curriculum be redesigned? What are the study bottlenecks?</td>
<td>Understanding where and why customers are lost. Where are the bottlenecks in the sales process? How to redesign the sales process?</td>
<td>Are patients treated in time? Why do different doctors operate in a different way? How to save costs? How to improve service levels?</td>
</tr>
<tr>
<td><strong>pre mortem</strong></td>
<td>What advice can we give a particular student that is likely to drop out? How to signal the lecturer that the exam is likely to be a &quot;massacre&quot; due to inactivity of students?</td>
<td>When to trigger a customer with a partially completed order? When to reroute an order to an account manager?</td>
<td>Predicting the most likely time until surgery. Which doctor should be selected to treat the patient? Should the patient be moved to another hospital (for logistic reasons).</td>
</tr>
</tbody>
</table>
Two types of models: “de jure models” and “de facto models”

- A **de jure model** is normative, i.e., it specifies how things should be done or handled. For example, a process model used to configure a BPM system is normative and forces people to work in a particular way. In other situations, normative models may be ignored by workers ("wallpaper models").

- A **de facto model** is descriptive and its goal is not to steer or control reality. Instead, de facto models aim to capture reality. Insights may be used for prediction, etc.
Visualizing "de jure" and "de facto" models

- reality
  - aims to describe reality as it really is
- de facto model
- de jure model
  - aims to describe reality as it should be
**Cartography: Process models as maps**

- **Discover.** This activity is concerned with the extraction of (process) models.
- **Enhance.** When existing process models (either discovered or hand-made) can be related to events logs, it is possible to enhance these models.
- **Diagnose.** This activity does not directly use event logs and focuses on classical model-based analysis.
Auditing: Confronting model and reality

- **Detect.** Compares de jure models with current “pre mortem” data. The moment a predefined rule is violated, an alert is generated (online).
- **Check.** The goal of this activity is to pinpoint deviations and quantify the level of compliance (offline).
- **Compare.** De facto models can be compared with de jure models to see in what way reality deviates from what was planned or expected.
- **Promote.** Promote parts of the de facto model to a new de jure model.
Navigation: Supporting and guiding process execution

- **Explore.** The combination of event data and models can be used to explore business processes at run-time.
- **Predict.** By combining information about running cases with models, it is possible to make predictions about the future, e.g., the remaining flow time and the probability of success.
- **Recommend.** The information used for predicting the future can also be used to recommend suitable actions (e.g. to minimize costs or time).
operational support

detect

predict

recommend
Operational support: online process mining using “pre mortem” event data

- **Navigation**
  - Explore
  - Predict
  - Recommend

- **Auditing**
  - Detect
  - Check
  - Compare
  - Promote

- **Cartography**
  - Discover
  - Enhance
  - Diagnose

**Current State**

- Known past: a, b
- Unknown future: c, d

**Detect**: b does not fit the model (not allowed, too late, etc.)

**Predict**: some prediction is made about the future (e.g., completion date or outcome)

**Recommend**: based on past experiences c is recommended (e.g., to minimize costs)
Running example: For simplicity we focus on control-flow and time

<table>
<thead>
<tr>
<th>case id</th>
<th>trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>\langle a^{12}<em>{\text{start}}, a^{19}</em>{\text{complete}}, b^{25}<em>{\text{start}}, d^{32}</em>{\text{complete}}, d^{33}<em>{\text{complete}}, e^{35}</em>{\text{start}}, e^{40}<em>{\text{complete}}, h^{50}</em>{\text{start}}, h^{54}_{\text{complete}} \rangle</td>
</tr>
<tr>
<td>2</td>
<td>\langle a^{17}<em>{\text{start}}, a^{23}</em>{\text{complete}}, d^{30}<em>{\text{start}}, d^{38}</em>{\text{complete}}, e^{50}<em>{\text{start}}, e^{59}</em>{\text{complete}}, g^{70}<em>{\text{start}}, g^{73}</em>{\text{complete}} \rangle</td>
</tr>
<tr>
<td>3</td>
<td>\langle a^{25}<em>{\text{start}}, a^{30}</em>{\text{complete}}, c^{32}<em>{\text{start}}, c^{38}</em>{\text{complete}}, d^{40}<em>{\text{start}}, d^{35}</em>{\text{complete}}, e^{45}<em>{\text{start}}, e^{50}</em>{\text{complete}}, f^{50}<em>{\text{start}}, f^{55}</em>{\text{complete}}, b^{60}<em>{\text{start}}, d^{35}</em>{\text{complete}}, d^{67}<em>{\text{complete}}, e^{80}</em>{\text{start}}, e^{87}<em>{\text{complete}}, g^{90}</em>{\text{start}}, g^{98}_{\text{complete}} \rangle</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- **a** start
- **b** examine thoroughly
- **c** examine casually
- **d** check ticket
- **e** decide
- **f** reinitiate request
- **g** pay compensation
- **h** reject request
- **end**
Transition system (with start/complete)
Operational support: Detect
Example

\[ \langle a_{start}^{12}, a_{complete}^{19}, b_{start}^{25}, d_{start}^{26} \rangle \]

\[ \langle a_{start}^{12}, a_{complete}^{19}, b_{start}^{25}, d_{start}^{26} \rangle \]

<table>
<thead>
<tr>
<th>case id</th>
<th>trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>\langle a_{start}^{12}, a_{complete}^{19}, b_{start}^{25}, d_{start}^{26}, c_{start}^{32}, a_{complete}^{33}, e_{start}^{35}, e_{complete}^{40}, h_{start}^{50}, h_{complete}^{54} \rangle</td>
</tr>
<tr>
<td>2</td>
<td>\langle a_{start}^{17}, a_{complete}^{23}, d_{start}^{28}, c_{start}^{32}, e_{start}^{38}, e_{complete}^{59}, e_{start}^{50}, e_{complete}^{70}, g_{start}^{70}, g_{complete}^{73} \rangle</td>
</tr>
<tr>
<td>3</td>
<td>\langle a_{start}^{25}, a_{complete}^{30}, c_{start}^{32}, a_{complete}^{35}, d_{start}^{40}, a_{complete}^{45}, d_{complete}^{50}, f_{start}^{50}, d_{complete}^{55} \rangle</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Intermezzo: Declarative process models

- **Procedural approach**: nothing is allowed unless specified.
- **Declarative approach**: everything is allowed unless explicitly prohibited.
- Declarative approaches are based on (temporal) constraints.
- **Declare** (http://www.win.tue.nl/declare/) is a language a toolset grounded in Linear Temporal Logic (LTL) on finite traces.
Declare: Basic idea

Informally: no $A$ and $B$ in the same case.

Alternatively regular expressions can be used.

<table>
<thead>
<tr>
<th>Name</th>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventually</td>
<td>$\Diamond F$</td>
<td>$F$ has to hold eventually, e.g., $[F,A,B,C,D,E]$, $[A,B,C,F,D,E]$, $[ABFCDFEF]$, etc.</td>
</tr>
<tr>
<td>always</td>
<td>$\Box F$</td>
<td>$F$ has to always hold, e.g., $[F,F,F,F,F,F]$.</td>
</tr>
</tbody>
</table>
Example: "existence response"

Informally: no A without a B in the same case.

\[\Diamond(A) \Rightarrow \Diamond(B)\]

- **OK:**
  - [ ]
  - [A,B,C,D,E]
  - [A,A,A,C,D,E,B,B,B]
  - [B,B,A,A,C,D,E]
  - [B,C,D,E]

- **NOK**
  - [A]
  - [A,A,C,D,E]
Example: "response"

Informally: any A is followed by a B in the same case.

- **OK:**
  - [ ]
  - [A,B,C,D,E]
  - [A,A,A,B,C,D,E]
  - [B,B,A,A,B,C,D,E]
  - [B,C,D,E]
- **NOK**
  - [A]
  - [B,B,B,B,A,A]
Example: "precedence"

OK:
- [ ]
- [A,B,C,D,E]
- [A,A,A,A,C,D,E,B,B,B]
- [A,A,C,D,E]

NOK:
- [B]
- [B,A,C,D,E]

Informally: any B is preceded by an A in the same case.
• **c1**: no payment and rejection in the same case.
• **c2**: any payment is preceded by a decision.
• **c3**: any rejection is preceded by a decision.
• **c4**: any registration is followed by a payment or rejection.

**c4** is a branching constraint
Declare specifications for detecting violations

- **Satisfied:** the LTL formula evaluates to true for the current partial trace.
- **Temporarily violated:** the LTL formula evaluates to false, however, there is a longer trace that evaluates to true.
- **Permanently violated:** the LTL formula evaluates to false for current trace and all its extensions.
Conflicting constraints

- A Declare specification is satisfied for a case if all of its constraints are satisfied.
- A Declare specification is temporarily violated by a case if for the current partial trace at least one of the constraints is violated, however, there is a possible future in which all constraints are satisfied.
- A Declare specification is permanently violated by a case if no such future exists.

Note that c1, c2, and c3 imply that e cannot be executed without permanently violating the specification.
End of intermezzo

- **Detect**: compares a "de jure" model with "pre mortem" data.
- If a lot of flexibility is allowed, one may use a declarative model.
- Specify what is not allowed rather than what should happen next.
Operational support: Predict

- Enterprise information system
- Partial trace
- Predicted completion date: 25-4-2011
- Operational support system
- Predictive model
- Event log
- Learn
Examples of predictions

- the predicted **remaining flow time** is 14 days;
- the predicted probability of meeting the legal **deadline** is 0.72;
- the predicted **total cost** of this case is 4500 euro;
- the predicted probability that activity "a" will **occur** is 0.34;
- the predicted probability that person "r" will **work on** this case is 0.57;
- the predicted probability that a case will be **rejected** is 0.67; and
- the predicted **total service time** is 98 minutes.
Annotated transition system

case id | trace
---|---
1 | \[a_{\text{start}}, a_{\text{complete}}, b_{\text{start}}, d_{\text{complete}}, e_{\text{start}}, e_{\text{complete}}, h_{\text{start}}, h_{\text{complete}}\]
2 | \[a_{\text{start}}, a_{\text{complete}}, d_{\text{start}}, c_{\text{start}}, d_{\text{complete}}, e_{\text{start}}, e_{\text{complete}}, g_{\text{start}}, g_{\text{complete}}\]
3 | \[a_{\text{start}}, a_{\text{complete}}, c_{\text{start}}, d_{\text{complete}}, e_{\text{start}}, e_{\text{complete}}, f_{\text{start}}, f_{\text{complete}}, b_{\text{start}}, a_{\text{complete}}, d_{\text{start}}, d_{\text{complete}}, c_{\text{start}}, c_{\text{complete}}, d_{\text{start}}, d_{\text{complete}}, e_{\text{start}}, e_{\text{complete}}, g_{\text{start}}, g_{\text{complete}}\]

... ... ...

t = time entering state
\(e\) = elapsed time (since first event)
\(r\) = remaining time (until last event)
\(s\) = sojourn time (time in state)
Annotated transition system

case 1

t = time entering state
e = elapsed time (since first event)
r = remaining time (until last event)
s = sojourn time (time in state)

case 2
Collect results per state

elapsed times: [21, 21, 15, 42, … ]
remaining times: [21, 35, 58, 31, … ]
sojourn times: [2, 12, 5, 13, … ]
average remaining flow time is 42.56
Operational support: Recommend

- Enterprise information system
- Partial trace
- Operational support system
- Model
- Event log

Suggestion: do x

Recommendations:
- x (85% certainty)
- y (12% certainty)
- z (3% certainty)

Learn recommendation model
Recommend

• Possible recommendations:
  − next activity,
  − suitable resource, or
  − routing decision.

• A recommendation is always given with respect to a specific goal.

• Example goals are:
  − minimize the remaining flow time,
  − minimize the total costs,
  − maximize the fraction of cases handled within 4 weeks,
  − maximize the fraction of cases that is accepted, and
  − minimize resource usage.
Relation between prediction and recommendation

current state

possible next state

prediction

a_1

a_2

a_k

...
applications of process mining
Next to the 10 different process mining activities, there are different perspectives:

- control-flow,
- data/rules,
- resources/organization,
- ...

Also in terms of application domains the scope of process mining is quite broad.
Process spectrum: From Lasagna to Spaghetti processes
Functional areas

- **Lasagna processes** are typically encountered in production, finance/accounting, procurement, logistics, resource management, and sales/CRM.
- **Spaghetti processes** are typically encountered in product development, service, resource management, and sales/CRM.
We applied ProM in >100 organizations

- Municipalities (e.g., Alkmaar, Heusden, Harderwijk, etc.)
- Government agencies (e.g., Rijkswaterstaat, Centraal Justitieel Incasso Bureau, Justice department)
- Insurance related agencies (e.g., UWV)
- Banks (e.g., ING Bank, Rabobank)
- Hospitals (e.g., AMC hospital, Isala, Catharina hospital)
- Multinationals (e.g., DSM, Deloitte)
- High-tech system manufacturers and their customers (e.g., Philips Healthcare, ASML, Ricoh, Thales)
- Media companies (e.g., Winkwaves)
- ...
How can process mining help?

- Detect bottlenecks
- Detect deviations
- Performance measurement
- Suggest improvements
- Decision support (e.g., recommendation and prediction)

- Provide mirror
- Highlight important problems
- Avoid ICT failures
- Avoid management by PowerPoint
- From “politics” to “analytics”
Lasagna processes
Example of a Lasagna process: WMO process of a Dutch municipality

Each line corresponds to one of the 528 requests that were handled in the period from 4-1-2009 until 28-2-2010. In total there are 5498 events represented as dots. The mean time needed to handled a case is approximately 25 days.
WMO process
(Wet Maatschappelijke Ondersteuning)

- WMO refers to the social support act that came into force in The Netherlands on January 1st, 2007.
- The aim of this act is to assist people with disabilities and impairments. Under the act, local authorities are required to give support to those who need it, e.g., household help, providing wheelchairs and scootmobiles, and adaptations to homes.
- There are different processes for the different kinds of help. We focus on the process for handling requests for household help.
- In a period of about one year, 528 requests for household WMO support were received.
- These 528 requests generated 5498 events.
C-net discovered using heuristic miner (1/3)
C-net discovered using heuristic miner (2/3)
C-net discovered using heuristic miner (3/3)
Conformance check WMO process (1/3)
Conformance check WMO process (2/3)
The fitness of the discovered process is 0.99521667. Of the 528 cases, 496 cases fit perfectly whereas for 32 cases there are missing or remaining tokens.
Bottleneck analysis WMO process (1/3)
Bottleneck analysis WMO process (2/3)
Bottleneck analysis WMO process (3/3)

Flow time of approx. 25 days with a standard deviation of approx. 28
Use cases for process mining

- **Goal:**
  - Improve KPIs related to **time**
  - Improve KPIs related to **costs**
  - Improve KPIs related to **quality**

- **Action:**
  - **Redesign** (improve process)
  - **Adjust** (improve control)
  - **Intervene** (handle problem in ad-hoc manner)
  - **Support** (detect, predict, recommend)
L* life-cycle model

Stage 0: plan and justify

Stage 1: extract
- historic data
- handmade models
- objectives (KPIs)
- questions

Stage 2: create control-flow model and connect event log
- event log
- control-flow model

Stage 3: create integrated process model
- current data
- process model
- event log
- interpret

Stage 4: operational support
- predict
- recommend
- detect
- diagnose

For a Lasagna process all stages are applicable (in principle).
Two Lasagna processes

RWS ("Rijkswaterstaat") process

WOZ ("Waardering Onroerende Zaken") process
RWS Process

• The Dutch national public works department, called “Rijkswaterstaat” (RWS), has twelve provincial offices. We analyzed the handling of invoices in one of these offices.

• The office employs about 1,000 civil servants and is primarily responsible for the construction and maintenance of the road and water infrastructure in its province.

• To perform its functions, the RWS office subcontracts various parties such as road construction companies, cleaning companies, and environmental bureaus. Also, it purchases services and products to support its construction, maintenance, and administrative activities.
C-net discovered using heuristic miner
Social network constructed based on handovers of work

Each of the 271 nodes corresponds to a civil servant. Two civil servants are connected if one executed an activity causally following an activity executed by the other civil servant.
Social network consisting of civil servants that executed more than 2000 activities in a 9 month period.

The darker arcs indicate the strongest relationships in the social network. Nodes having the same color belong to the same clique.
WOZ process

- Event log containing information about 745 objections against the so-called WOZ ("Waardering Onroerende Zaken") valuation.
- Dutch municipalities need to estimate the value of houses and apartments. The WOZ value is used as a basis for determining the real-estate property tax.
- The higher the WOZ value, the more tax the owner needs to pay. Therefore, there are many objections (i.e., appeals) of citizens that assert that the WOZ value is too high.
- "WOZ process" discovered for another municipality (i.e., different from the one for which we analyzed the WMO process).
The log contains events related to 745 objections against the so-called WOZ valuation. These 745 objections generated 9583 events. There are 13 activities. For 12 of these activities both start and complete events are recorded. Hence, the WF-net has 25 transitions.
Conformance checker:
(fitness is 0.98876214)
Performance analysis

bottleneck detection: places are colored based on average durations

time required to move from one activity to another

information on total flow time

Performance information of the selected transitions:

<table>
<thead>
<tr>
<th>Frequency: 416 cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time in between (days)</td>
</tr>
<tr>
<td>avg 202.73</td>
</tr>
<tr>
<td>min 126.89</td>
</tr>
<tr>
<td>max 245.98</td>
</tr>
<tr>
<td>stdev 19.74</td>
</tr>
<tr>
<td>fast 25.00% 177.2</td>
</tr>
</tbody>
</table>

Arrival rate:
2,85 cases per day

Throughput time (days):

<table>
<thead>
<tr>
<th></th>
<th>Throughput time (days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>177.99</td>
</tr>
<tr>
<td>min</td>
<td>3.78</td>
</tr>
<tr>
<td>max</td>
<td>251.9</td>
</tr>
<tr>
<td>stdev</td>
<td>52.87</td>
</tr>
<tr>
<td>fast 25...</td>
<td>98.98</td>
</tr>
<tr>
<td>slow 2...</td>
<td>230.76</td>
</tr>
<tr>
<td>normal</td>
<td>191.11</td>
</tr>
</tbody>
</table>
## Resource-activity matrix (four groups discovered)

<table>
<thead>
<tr>
<th>user</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
<th>$a_6$</th>
<th>$a_7$</th>
<th>$a_8$</th>
<th>$a_9$</th>
<th>$a_{10}$</th>
<th>$a_{11}$</th>
<th>$a_{12}$</th>
<th>$a_{13}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>user 1</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user 2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>38</td>
<td>0</td>
<td>69</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user 3</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user 4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user 5</td>
<td>117</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>user 6</td>
<td>172</td>
<td>6</td>
<td>14</td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>48</td>
<td>53</td>
</tr>
<tr>
<td>user 7</td>
<td>1</td>
<td>41</td>
<td>8</td>
<td>14</td>
<td>275</td>
<td>8</td>
<td>8</td>
<td>865</td>
<td>55</td>
<td>180</td>
<td>0</td>
<td>128</td>
<td>5</td>
</tr>
<tr>
<td>user 8</td>
<td>2</td>
<td>868</td>
<td>7</td>
<td>6</td>
<td>105</td>
<td>0</td>
<td>0</td>
<td>79</td>
<td>266</td>
<td>441</td>
<td>0</td>
<td>844</td>
<td>3</td>
</tr>
<tr>
<td>user 9</td>
<td>90</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>user 10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>899</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1019</td>
</tr>
<tr>
<td>user 11</td>
<td>336</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>18</td>
<td>23</td>
</tr>
<tr>
<td>user 12</td>
<td>1</td>
<td>645</td>
<td>13</td>
<td>21</td>
<td>419</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>217</td>
<td>281</td>
<td>1</td>
<td>334</td>
<td>9</td>
</tr>
<tr>
<td>user 13</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user 14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user 15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>user 16</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user 17</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user 18</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user 19</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user 20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>258</td>
</tr>
</tbody>
</table>
Spaghetti processes
Remember: How can process mining help?

- Detect bottlenecks
- Detect deviations
- Performance measurement
- Suggest improvements
- Decision support (e.g., recommendation and prediction)

- Provide mirror
- Highlight important problems
- Avoid ICT failures
- Avoid management by PowerPoint
- From “politics” to “analytics”
Example of a Spaghetti process

Spaghetti process describing the diagnosis and treatment of 2765 patients in a Dutch hospital. The process model was constructed based on an event log containing 114,592 events. There are 619 different activities (taking event types into account) executed by 266 different individuals (doctors, nurses, etc.).
Fragment
18 activities of the 619 activities (2.9%)
Another example (event log of Dutch housing agency)

The event log contains 208 cases that generated 5987 events. There are 74 different activities.

note the vertical patterns
L* approach for Spaghetti processes

Stage 0: plan and justify

Stage 1: extract
- historic data
- handmade models
- objectives (KPIs)
- questions

Stage 2: create control-flow model and connect event log
- event log
- control-flow model

Stage 3: create integrated process model
- event log
- process model

Stage 4: operational support
- explore
- discover
- check
- compare
- promote
- enhance
- detect
- predict
- recommend
- interpret
- diagnose
- redesign
- adjust
- intervene

Focus on Stages 0-2, Stage 3 is typically only partially possible, Stage 4 requires more structure.
Any process model can be simplified: filtering

Filtering in ProM 6: select top 80% of activities in event log of housing agency.
Effect of filtering
(event log of Dutch housing agency)
Filtering in ProM 5.2: select activities that appear in more than 5% of the cases in the hospital log.
Trade-off

- split heterogeneous log into smaller more homogeneous logs
- smaller event logs each corresponding to a “simple” process model

entire event log

fewer models, but more complex

more models, but simpler
Fuzzy mining
(event log of Dutch housing agency)

See Disco, Perceptive process mining, Celonis, etc. for similar functionality inspired by ProM's fuzzy miner.
More Spaghetti processes

Processes of ASML, Philips Healthcare, and AMC.
Test process ASML

• **ASML** is the world’s leading manufacturer of chip-making equipment and a key supplier to the chip industry.

• The testing of manufactured **wafer scanners** is an important, but also time-consuming, process.

• Every wafer scanner is tested in the factory of ASML. When it passes all tests, the wafer scanner is disassembled and shipped to the customer where the system is re-assembled (and tested again).
About the example log

• The event log containing 154,966 events.
• The event log contained information about 24 carefully chosen wafer scanners (same type, same circumstances, and having complete logs).
• The number of events per case (i.e., the length of the executed test sequence) in this event log ranges from 2820 to 16250 events.
• There are 360 different activities, all identified by four letter test codes.
• Each instance of these 360 activities has a start event and complete event.
Discovered process model (just complete events)
Conformance checking

- ASML also had a so-called reference model describing the way that machines should be tested.
- This reference model is at the level of job steps rather than test codes. However, ASML maintains a mapping from the lower level codes to these higher level activities. Comparing the reference model and our discovered model (both at the job step and test code level) revealed interesting differences.
- Moreover, using the ProM’s conformance checker we could show that the average fitness was only 0.375, i.e., less than half of the events can be explained by the model.
- When replaying, we discovered many activities that had occurred but that should not have happened according to the reference model and activities that should have happened but did not.
Philips Healthcare: Allura Xper systems

- **Philips Healthcare** is one of the leading manufacturers of medical devices, offering diagnosing imaging systems, healthcare information technology solutions, patient monitoring systems, and cardiac devices.

- **Philips Remote Services (PRS)** is a system for the active monitoring of systems via the Internet. PRS has been established to deliver remote technical support, monitoring, diagnostics, application assistance, and other added value services.

- We analyzed the event logs of **Allura Xper systems**. These are X-ray systems designed to diagnose and possibly assist in the treatment of all kinds of diseases, like heart or lung diseases, by generating images of the internal body.
Fuzzy miner tailored towards the needs of Philips Healthcare
Group of 627 gynecological oncology patients treated in 2005 and 2006.

The event log contains 24331 events referring to 376 different activities.
Social network
(between different organizational units of the AMC hospital)
Analyzing “Spaghetti Processes”

More difficult, but …

the potential gains are also more substantial.
conclusion and outlook
Process mining: many activities, perspectives, and applications

- **People**, **machines**, **business processes**, **organizations**, **documents**

**“World”**

- **information system(s)**
- **provenance**
- **event logs**
  - “pre mortem”
  - “post mortem”
- **current data**
- **historic data**

**navigation**
- **explore**
- **predict**
- **plan**
- **recommend**
- **control-flow**
- **data/rules**
- **resources/organization**

**models**
- **de jure models**
  - control-flow
  - data/rules
  - resources/organization
- **de facto models**
  - control-flow
  - data/rules
  - resources/organization

**auditing**
- **detect**
- **check**
- **compare**
- **promote**

**cartography**
- **discover**
- **enhance**
- **diagnose**

**Stage 0: plan and justify**
- data understanding
- business understanding

**Stage 1: extract**
- event log

**Stage 2: create control-flow model and connect event log**
- de jure model
- de facto model

**Stage 3: create integrated process model**
- current data
- event log
- process model

**Stage 4: operational support**
- interpret
- recommend
- support
- redesign
- adjust
- intervene

**resources/organization**

**Stage 0:**
- Plan and justify
- Data understanding
- Business understanding

**Stage 1:**
- Extract
  - Event log

**Stage 2:**
- Create control-flow model and connect event log
  - De jure model
  - De facto model

**Stage 3:**
- Create integrated process model
  - Current data
  - Event log
  - Process model

**Stage 4:**
- Operational support
  - Interpret
  - Recommend
  - Support
  - Redesign
  - Adjust
  - Intervene
Outlook

Part I: Preliminaries

- Chapter 2: Process Modeling and Analysis
- Chapter 3: Data Mining

Part II: From Event Logs to Process Models

- Chapter 4: Getting the Data
- Chapter 5: Process Discovery: An Introduction
- Chapter 6: Advanced Process Discovery Techniques

Part III: Beyond Process Discovery

- Chapter 7: Conformance Checking
- Chapter 8: Mining Additional Perspectives
- Chapter 9: Operational Support

Part IV: Putting Process Mining to Work

- Chapter 10: Tool Support
- Chapter 11: Analyzing 'Lasagna Processes'
- Chapter 12: Analyzing 'Spaghetti Processes'

Part V: Reflection

- Chapter 13: Cartography and Navigation
- Chapter 14: Epilogue