Overview

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
Process mining spectrum

The process mining spectrum involves analyzing event logs to support and control software systems. It involves the discovery of models, conformance, and enhancement of business processes in the context of people, machines, components, and organizations.
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
Business process provenance

information system(s)

provenance

people
machines
business processes
“world”
organizations
documents

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.
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.

- **A de facto model** is descriptive and its goal is not to steer or control reality. Instead, de facto models aim to capture reality.
Cartography

- **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

- **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

- **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:
online process mining using "pre mortem" event data

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)
<table>
<thead>
<tr>
<th>case id</th>
<th>trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>\langle a_{\text{start}}^{12}, a_{\text{complete}}^{19}, b_{\text{start}}^{25}, d_{\text{start}}^{26}, b_{\text{complete}}^{32}, d_{\text{complete}}^{33}, e_{\text{start}}^{35}, e_{\text{complete}}^{40}, h_{\text{start}}^{50}, h_{\text{complete}}^{54} \rangle</td>
</tr>
<tr>
<td>2</td>
<td>\langle a_{\text{start}}^{17}, a_{\text{complete}}^{23}, d_{\text{start}}^{28}, c_{\text{start}}^{30}, d_{\text{complete}}^{32}, c_{\text{complete}}^{38}, e_{\text{start}}^{50}, e_{\text{complete}}^{59}, g_{\text{start}}^{70}, g_{\text{complete}}^{73} \rangle</td>
</tr>
<tr>
<td>3</td>
<td>\langle a_{\text{start}}^{25}, a_{\text{complete}}^{30}, c_{\text{start}}^{32}, c_{\text{complete}}^{35}, d_{\text{start}}^{35}, d_{\text{complete}}^{40}, e_{\text{start}}^{45}, e_{\text{complete}}^{50}, f_{\text{start}}^{50}, f_{\text{complete}}^{55}, b_{\text{start}}^{60}, b_{\text{complete}}^{62}, b_{\text{complete}}^{65}, a_{\text{complete}}^{67}, e_{\text{start}}^{80}, e_{\text{complete}}^{87}, g_{\text{start}}^{90}, g_{\text{complete}}^{98} \rangle</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Transition system (with start/complete)
Operational support: Detect

- Enterprise information system
- Operational support system
- Normative model
- Event log

Arrow flows indicate the process:
- Partial trace
- Violation detected!
Example

\[ \langle a_{\text{start}}, a_{\text{complete}}, b_{\text{start}}, d_{\text{start}} \rangle \]

\begin{center}
\begin{tabular}{|c|c|}
\hline
\textbf{case id} & \textbf{trace} \\
\hline
1 & \langle a_{\text{start}}, a_{\text{complete}}, b_{\text{start}}, d_{\text{start}} \rangle \\
2 & \langle a_{\text{start}}, a_{\text{complete}}, b_{\text{start}}, d_{\text{start}} \rangle \\
3 & \langle a_{\text{start}}, a_{\text{complete}}, b_{\text{start}}, d_{\text{start}} \rangle \\
\hline
\end{tabular}
\end{center}

alert!!!!
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.
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
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

Recommendation:
- 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.

• Examples of 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
Process mining spectrum

people
machines
business processes
organizations
documents

“world”

information system(s)
provenance

event logs
"pre mortem"
current data
historic data
"post mortem"

models

de jure models
de facto models

control-flow
data/rules
resources/organization

control-flow
data/rules
resources/organization

navigation

auditing
cartography

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