Process Mining (ProMi)

Winter Term 2015/16

Matthias Weidlich
Before we start...

Course language?

The course will be given in English unless all students unanimously vote for German as the teaching language in the first lecture.
What is your background?

• Where are you from?
• Why did you choose this course?
• What is your knowledge of
  • Business processes
  • Behavioural formalisms
  • Data Mining
Who am I?

My academic story....

Process-driven Architectures

- Process-oriented systems
- Event-processing systems

Contact

- matthias.weidlich@hu-berlin.de
- Office 4.101
Organisation

When: Tuesdays, 9:00 – 13:00
Where: Room 1.307
What:

1\textsuperscript{st} part (Oct, Nov, Dec)
  - Lectures and recitations
  - Formal foundations: models and algorithms
  - Hands-on tutorial

2\textsuperscript{nd} part (Jan, Feb)
  - Seminar style
  - Recent research paper
  - 45min presentation and critical assessment of paper

Grading:
  - Oral exam at the end of semester
  - Requirement for taking the final exam: seminar style presentation
Material

Slides will be available before the lecture

Further literature

What is process mining?
Process Mining Techniques

Discovery
• Evidence based creation of process models
• Use information observed during operation inside an organisation
• Usage of logs that hint at what is going on in an organisation

Conformance checking
• Detect discrepancies between process model and observed information
• Analyse deviations

Enhancement
• Extend a process model with observed information
• Example: a model is annotated with performance data
Isn’t that just data mining?

Supervised learning
- Data labelled with response variable
- Explain response variable based on predictor variable
- Classification (categorical) and regression (numerical)

Unsupervised learning
- Unlabelled data
- Clustering and pattern discovery

BUT: learned models are comparatively simple and not aware of processes
Spotlight: Discovery
Spotlight: Hands-On Tutorial
Content

I. The Context of Process Mining
   1) Process-oriented information systems
   2) Event logs and their creation

II. Process discovery
   1) Mining algorithms (Alpha, ILP, Inductive)
   2) Quality dimensions of discovered models

III. Conformance Checking
   1) Replay-based techniques
   2) Relational techniques
   3) Trace alignment

IV. Enhancement
   1) Time prediction
   2) Decision mining

V. Presentations on cutting-edge approaches
I. The Context of Process Mining

Process-Oriented Information Systems
The Context
Relevance of Business Processes

Business processes are everywhere

- Products and services are provided by activities
- Execution of activities requires coordination
- Success of this coordination influences costs, time, and quality of products and services

"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]

"a set of logically related tasks performed to achieve a defined business outcome for a particular customer or market"
[Davenport 1992]
Scenario: Insurance Claim Handling

Record claim
Check coverage
Request proof of loss
Do field check
Take decision
Inform claimant
Compensation payment
Archive claim
Scenario: Online Reselling

Submit order
Check credit history
Charge credit card
Check availability
Plan shipments
Aggregate shipments
Last mile delivery
Record customer status
The Context
Process-oriented Information System

Process-oriented Information System (POIS)

• "a generic software system that is driven by explicit process representations to coordinate the enactment of business processes”  
  [Weske 2007]

Process-orchestration

• "a system acts as a central agent that controls the execution of the process activities, very similar to a conductor centrally controlling the musicians in an orchestra”
Process-based Integration

Integration logic is encoded in process model
Workflow engine executes the integration process
System activities vs. human activities

Service Oriented Architectures

Goals

• Reusability of well-defined functionality: services
• Create new applications based on existing services

Requirements

• Service descriptions are available and precise
• Identification, specification, and realization of business relevant functionality (*service carving*)
• Implementation of services (*service enabling*)

Web services

• Self-contained, self-describing, modular applications that can be published, located, and invoked across the web
• Perform functions, anything from simple requests to complicated business processes
• Once deployed, they can be discovered and invoked
Service Composition Process

Receive Purchase Order → Initiate Price Calculation → Complete Price Calculation → Complete Production Scheduling → Decide on Shipper → Arrange Logistics → Complete Production Scheduling → Invoice Processing

Receive Purchase Order → Initiate Production Scheduling

Initiate Production Scheduling

Invoke requestShipping

Arrange Logistics

Invoice Processing

Invoice Processing → reply Invoice

shifting

purchasing
Beyond System Workflows

Human Interaction Workflows

- User interaction during process execution
- Combination of manual and fully automated activities
- Active control of process by interaction with process participants

Human workflow systems typically also include:

- Modelling and integration of process participants (roles, capabilities)
- Provisioning of specific interfaces (work lists)
Human Interactions

Navigation form

Worklist form
Example of a Human Interaction Workflow

The Context
The Data Hype

"DATA IS THE NEW OIL."

From the beginning of recorded time until 2003, we created 5 exabytes of data. In 2011 the same amount was created every two days. By 2013, it's expected that the time will shrink to 50 minutes.

247 billion emails are sent every day.

80% of all humans own a mobile phone.

60% of all humans (5.4 billion people) are active Internet users. In 2010, 183,000 text messages were sent every second.

29

The Data Hype

Coined in 2006 by Clive Humby, a British data commercialization entrepreneur, this now famous phrase was embraced by the World Economic Forum in 2011 report, which considered data to be an economic asset, like oil.
Data Sensing

30 BILLION
Sensor enabled objects connected to networks by 2020

212 BILLION
Total number of available sensor enabled objects by 2020

212B is 28x the total population of the world

Data source: IDC
Data Sensing

“Things” refer to any physical object with a device that has its own IP address and can connect & send/receive data via a network.
Impact on Everyday Life
Events

Event – *happening of interest*

- Have timestamp: occurrence time, arrival time, ...
- Carry data
  - Typical relational model based on attributes
  - Payload is modelled as key-value pairs

Event type – type for events of similar structure and semantics

- Events are instances of event type
- Defines the set of attributes of the events
Event logs

Process context

- Activity – what has been executed?
- Time – when has it been executed?
- Case – for which process instance has it been executed?

<table>
<thead>
<tr>
<th>Trans. ID</th>
<th>Activity</th>
<th>Start Time</th>
<th>End Time</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>8287</td>
<td>Enter customer data</td>
<td>08:34:15</td>
<td>08:37:44</td>
<td>User jsmith</td>
</tr>
<tr>
<td>8287</td>
<td>Check credit</td>
<td>08:37:52</td>
<td>08:38:05</td>
<td>Equifax service call</td>
</tr>
<tr>
<td>1399</td>
<td>Enter customer data</td>
<td>08:37:59</td>
<td>08:44:40</td>
<td>User sjones</td>
</tr>
<tr>
<td>8287</td>
<td>Enter order</td>
<td>08:38:09</td>
<td>08:38:39</td>
<td>ERP system call</td>
</tr>
<tr>
<td>1399</td>
<td>Check credit</td>
<td>08:44:58</td>
<td>08:45:06</td>
<td>Equifax service call</td>
</tr>
<tr>
<td>4283</td>
<td>Enter order</td>
<td>08:45:01</td>
<td>08:45:35</td>
<td>ERP system call</td>
</tr>
<tr>
<td>1399</td>
<td>Enter order</td>
<td>08:45:18</td>
<td>08:45:38</td>
<td>ERP system call</td>
</tr>
</tbody>
</table>
The Context

- “world” supports/controls
- process-aware information system
- models analyzes
- specifies configures implements analyzes
- discovery conformance extension
- records events, e.g., messages, transactions, etc.
- event logs

process models
BPM Lifecycle and Models

1. Process identification
2. Process architecture
3. Process discovery
   - Conformance and performance insights
   - As-is process model
4. Process monitoring and controlling
   - Executable process model
5. Process implementation
   - To-be process model
6. Process analysis
   - Insights on weaknesses and their impact
7. Process redesign
Purposes of Modelling

Large variety of modelling purposes

• Business purposes
• Information systems purposes

Business purposes

• Documentation, guidelines, work instructions
• Process redesign, from as-is to to-be
• Staff planning, often using statistical annotations
• Quality certification
Purposes of Modelling cont.

Information systems purposes

• Enterprise Resource Planning (ERP) system selection
  • ERP systems provide business functionality
  • System selection based on delta-analysis of own processes and implemented process

• Software development
  • Process models as requirement documents

• Process implementation
  • Workflow system supports execution of cases
  • Different degrees of automation of activities
Process models answer questions

- What is done?
- Who is responsible?
- What are the decisions taken?
- How long does it take to finish the process?
- Who is affected by a change in the process?
Mapping Business Processes

What is mapped to a process model?

• Activities
  Building blocks that describe elementary pieces of work

• Routing conditions
  Describe temporal and logical constraints on the execution of activities

• Inputs, Outputs
  Informational or physical artefacts processed by activities

• Events
  How time, messages, exception influence the execution

• Resources
  Persons, organisational units, systems that execute activities
Abstraction

Abstraction is information loss

- **Projection**
  Remove information considered irrelevant

- **Classification**
  Aggregate related information
  Different types of classification
Abstraction Overview

Levels of Abstraction:
- Meta-model
  - process \(1 \rightarrow n\) activity
  - process landscape

Levels of Granularity:
- Level 1:
  - accounting
  - production
  - sales
- Level 2:
  - make pin
  - request credit
  - hire staff
  - book trip
- Level 3:
  - pin made by Adam Smith, 1776
  - pin made by Peter Smith, 1776
  - pin made by John Smith, 1776
  - credit request by Hammer & Champy, 1990
  - credit request by Davenport, 1990
  - credit request by Kettinger, 1997
  - hiring of Taylor, 1911
  - hiring of Ford, 1925
  - hiring of Rosemann, 2003
Perspectives

Perspectives of process modelling

• Control flow
• Organisational structures
• Data structures
• IT landscapes

I. The Context of Process Mining

Process-Oriented Information Systems

Behavioural Formalisms
Overview

Zoo of process modelling languages
  • Different expressiveness
  • Different syntax, semantics, notation

Yet, precise definition of syntax and semantics is a prerequisite for any work on process mining
  • Structure and meaning of model needs to be unambiguous

Next: recap
  • Traces
  • Labelled transition systems
  • Petri nets
Trace Semantics

Interpret process as a set of traces
- Valid sequences of activity executions
- Notion of a single (or multiple) “start(s)” of the process
- No notion of “end” of the process
- Also called sequential runs
- Set of traces may be infinite

Example:
- \(<A, B, C, D, G, H>\)
- \(<A, B>\)
- \(<A, B, C, D, F, C, D, E, F, C>\)
Background: LTS

Consider not only the occurrence of activities, but the states of a process

- Finite representation of infinite sets of traces
- Enables one to consider the “moment of choice”

Labelled Transition System (LTS) is a tuple \((S, s_0, L, R)\)

- A set \(S\) of states
- An initial state \(s_0 \in S\)
- A set \(L\) of labels
- A labelled transition relation \(R \subseteq S \times L \times S\)
By Example
The Question of Equivalence

When are two processes equivalent?

• Process models describe behaviour
• Hence, we focus on behaviour equivalence
• Other techniques for assessing equivalence of model structure, business semantics of activities

When are two process models behaviour equivalent?

• Question is answered for certain behavioural model
• Variety of interpretations of behaviour equivalence
Trace Equivalence

Idea: the set of traces is equal

- Let $T_1$ and $T_2$ be sets of traces of models $S_1$ and $S_2$
- $S_1$ and $S_2$ are trace equivalent, $S_1 \approx_{tr} S_2$, iff $T_1 = T_2$

Some of the examples are due to Ed Brinksma/Jan Tretmans
Trace Equivalence cont.

\[
T = \{ \epsilon, a, ab, ac \}
\]
Complete Trace Equivalence

Idea: the set of complete traces is equal

- Let $T_{c1}$ and $T_{c2}$ be sets of complete traces of models $S_1$ and $S_2$
- $S_1$ and $S_2$ are complete trace equivalent, $S_1 \approx_{ctr} S_2$, iff $T_{c1} = T_{c2}$
Bisimulation

- Idea: either model can simulate the other model, i.e., it can mirror the state transitions
- Let $W = (U,u,L,V)$ and $W' = (U',u',L',V')$ be LTS
- $W$ and $W'$ are bisimilar, $W \approx_b W'$, iff there exists a binary relation $R$ over $U \times U'$ such that
  - $(u,u') \in R$
  - If $(x,x') \in R$ and $(x,l,y) \in V$, then there exists a tuple $(x',l,y') \in V'$ such that $(y,y') \in R$
  - If $(x,x') \in R$ and $(x',l,y') \in V'$, then there exists a tuple $(x,l,y) \in V$ such that $(y,y') \in R$
Bisimulation cont.
Background: Petri nets

Idea

• Formal description of concurrent systems
• Formal model and graphical representation are equivalent

Background

• Foundations developed by Carl Adam Petri, 1962
• Variety of variants and extensions
• Here: modelling and analysis of business processes
The Essence

- A Petri net is a directed graph consisting of places, transitions, and arcs between them
- Petri nets are often referred to as bipartite graphs (not really correct, though)
  - An arc is defined from a place to a transition, or from a transition to a place
- Notation by example:

Syntax

**Definition 4.1** A *Petri net* is a tuple \((P, T, F)\) with

- a finite set \(P\) of places
- a finite set \(T\) of transitions such that \(T \cap P = \emptyset\)
- a flow relation \(F \subseteq (P \times T) \cup (T \times P)\)
- A place \(p \in P\) is an input place of a transition \(t \in T\), if and only if there exists a directed arc from \(p\) to \(t\), i.e., if and only if \((p, t) \in F\). The set of input places for a transition \(t\) is denoted \(\bullet t\).
- A place \(p\) is an output place of a transition \(t\), if and only if there exists a directed arc from \(t\) to \(p\), i.e., if and only if \((t, p) \in F\). The set of output places for a transition \(t\) is denoted \(t \bullet\).
- \(p\bullet\) and \(\bullet p\) denote the set of transitions that share \(p\) as input places or output place, respectively.
Example

\[ P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}, \]
\[ T = \{t_1, t_2, t_3, t_4, t_5\}, \]
\[ F = \{(p_1, t_1), (t_1, p_2), (p_2, t_2), (t_2, p_3), (t_2, p_4), (p_3, t_3), (p_4, t_4), (t_3, p_5), (t_4, p_6), (p_5, t_5), (p_6, t_5), (t_5, p_7)\}. \]
Semantics

- Dynamic behaviour is represented by tokens in the Petri net
- State of a Petri net (the *marking*) is described as a distribution of tokens over places

**Definition 4.2** The *marking* (or *state*) of a Petri \((P, T, F)\) net is defined by a function \(M : P \rightarrow \mathbb{N}\) mapping the set of places onto the natural numbers, where \(\mathbb{N}\) is the set of natural numbers including zero.

Enabling, Firing, Reachability

**Definition 4.3** Let \((P,T,F)\) be a Petri net and \(M\) a marking. The firing of a transition is represented by a state change of the Petri net.

- A transition \(t\) is **enabled** in a state \(M\) if \(M(p) = 1\) for all input places \(p\) of \(t\).
- The firing of a transition \(t\) in a state \(M\) results in state \(M'\), where

\[
(\forall p \in \bullet t) M'(p) = M(p) - 1 \land (\forall p \in t \bullet) M'(p) = M(p) + 1
\]

- \(M \xrightarrow{t} M'\) indicates that by firing \(t\) the state of the Petri net changes from \(M\) to \(M'\).
- \(M_1 \xrightarrow{*} M_n\) means that there is a sequence of transitions \(t_1, t_2, \ldots t_{n-1}\) such that \(M_i \xrightarrow{t_i} M_{i+1}\), for \(1 \leq i < n\).
- A state \(M'\) is **reachable** from a state \(M\), if and only if \(M \xrightarrow{*} M'\).
Enabling, Firing, Reachability cont.

Firing of a transition is atomic

Multiple transitions may be enabled, but only one fires as part of a state change

The number of tokens in a net may vary if there are transitions for which the number of input places is not equal to the number of output places

The network, the net structure, is static
Non-Determinism

Two transitions are enabled but only one can fire

from: Wil van der Aalst, Petri net refresher
Petri Net System

- A Petri net system is a pair \((N, M)\)
  - \(N = (P,T,F)\) is a Petri net
  - \(M\) is the initial marking
- Let \(((\{p1,p2,p3\}, \{t1,t2,t3\}, \{(p1,t1), (t1,p2), (p2,t2), (t2,p3), (p2,t3),(t3,p3)\}), (1,0,0))\) be a Petri net system
- \(M_1 \xrightarrow{o} M_3\) with \(o=(t1,t2)\) changes the state of the net system from marking \(M_1\) to marking \(M_3\)

\[M_1 : (1,0,0) \quad M_3 : (0,0,1)\]
Reachability

Examples
- (0,1,0,0,1) is reachable from (1,1,0,0,0) by \( o=(t_1,t_2,t_3) \)
- (0,1,1,0,0) is NOT reachable from (1,1,0,0,0) since there is no according firing sequence

Remark
- Multiset notation is often used for markings
- In the example: \([p_1,p_2]\) instead of (1,1,0,0,0)
- Multiple tokens in one place: \([p_1,p_1,p_2]\) or \([p_1^2,p_2]\) instead of (2,1,0,0,0)
Reachability Graph Example

Reachability graph

- Nodes are states, i.e., markings
- Arcs represent state transitions
Workflow Petri Nets

Use Petri nets to model business processes

• Transitions represent activities
• Places represent states of the business process
• Arcs represent control flow dependencies
• Token represent instances (tokens may carry values in CPNs)
• Behaviour of process instance is represented by firing rule
• Behaviour of transitions is influenced by tokens in the preset

In fact: many process modelling languages are rooted in Petri net systems
Definition 4.8 A Petri net $PN = (P, T, F)$ is called \textit{workflow net}, if and only if the following conditions hold:

- There is a distinguished place $i \in P$ (called initial place) that has no incoming edge, i.e., $\bullet i = \emptyset$.
- There is a distinguished place $o \in P$ (called final place) that has no outgoing edge, i.e., $o \bullet = \emptyset$.
- Every place and every transition is located on a path from the initial place to the final place.
Example

Properties

1. **i** is the only initial place:
   It holds, for each \( p \in P \): \( \bullet p \neq \emptyset \) or \( p = i \)

2. **o** is the only final place:
   It holds, for each \( p \in P \): \( p\bullet \neq \emptyset \) or \( p = o \)

3. Let \( N \) be a WF-net.
   If it is extended with a transition \( t^* \), which connects \( o \) with \( i \) (i.e. \( \bullet t^* = \{o\} \) and \( t^*\bullet = \{i\} \)), the resulting net is strongly connected.

**Strongly connected:**
A graph is strongly connected, if and only if for each pair of nodes \((x,y)\) there is a path from \( x \) to \( y \).
Take Away

Process Mining links data analysis with process management

Event data is used to discover process models, assess their conformance, or enhance them with additional details

Behavioural formalisms as the basis of techniques for process mining