Part III – Beyond control-flow mining

- Organizational mining
- Discovery of social nets
- Extension algorithms
Outline

- Part I – Introduction to Process Mining
  - Context, motivation and goal
  - General characteristics of the analyzed processes and logs
  - Classification of Process Mining approaches

- Part II – Workflow discovery
  - Induction of basic Control Flow graphs
  - Other techniques (α-algorithm, Heuristic Miner, Fuzzy mining)

- Part III – Beyond control-flow mining
  - Organizational mining
  - Social net discovery
  - Extension algorithms

- Part IV – Evaluation and validation of discovered models
  - Conformance Check
  - Log-based property verification

- Part V – Clustering-based Process Mining
  - Discovery of hierarchical process models
  - Discovery of process taxonomies
  - Outlier detection
Organizational mining techniques
Organizational Mining Algorithms

- **Objective:**
  - Discover the organizational model (i.e., roles, departments, etc.) without prior knowledge about the structure of the organization
  - Aid in understanding and improving social and organizational structures

- **Two types of algorithms**
  - **Organizational Model**
    - Mining of roles and teams in organizations
    - ProM Plug-in: Organizational Miner
  - **Social Networks**
    - Discovery of relationships among originators
    - ProM Plug-ins: Social Network Miner and Analyze Social Network
Organizational Miner

- Main idea: Which originators are executing which tasks
  - Default mining
  - Doing Similar Tasks
- Methods to mine **roles**
  - Working together
- Methods to mine **teams**
  - Working together
Organizational Miner

- Main idea: Which performers are executing which tasks
- Methods to mine roles
  - Default mining
  - Doing Similar Tasks
- Methods to mine teams
  - Working together
Organizational Miner

- Main idea: Which performers are executing which tasks

- Methods to mine **roles**
  - Default mining
  - *Doing Similar Tasks*

- Methods to mine **teams**
  - Working together
Default Mining

Doing Similar Tasks
Main idea: Which performers are executing which tasks

Methods to mine roles
- Default mining
- Doing Similar Tasks

Methods to mine teams
- Working together
Organizational Miner

Why is the notion of process instances necessary to mine teams but unnecessary to mine roles?
Outline

- **Part I – Introduction to Process Mining**
  - Context, motivation and goal
  - General characteristics of the analyzed processes and logs
  - Classification of Process Mining approaches

- **Part II – Workflow discovery**
  - Induction of basic Control Flow graphs
  - Other techniques (α-algorithm, Heuristic Miner, Fuzzy mining)

- **Part III – Beyond control-flow mining**
  - Organizational mining
  - Social net discovery
  - Extension algorithms

- **Part IV – Evaluation and validation of discovered models**
  - Conformance Check
  - Log-based property verification

- **Part V – Clustering-based Process Mining**
  - Discovery of hierarchical process models
  - Discovery of process taxonomies
  - Outlier detection
Social Network Miner

- **Aim:**
  - Monitor how individual process instances are routed between originators

- **Metrics**
  - Handover of work
  - Subcontracting
  - Reassignment
  - Working together
  - Similar task
Social Network Miner

- **Aim:** Monitor how individual process instances are routed between originators

- **Metrics**
  - *Handover of work*
  - Subcontracting
  - Reassignment
  - Working together
  - Similar task

---

John  
Mary
Social Network Miner

- **Aim:** Monitor how individual process instances are routed between originators

- **Metrics**
  - Handover of work
  - *Subcontracting*
  - Reassignment
  - Working together
  - Similar task
Social Network Miner

- **Aim:** Monitor how individual process instances are routed between originators

- **Metrics**
  - Handover of work
  - Subcontracting
  - **Reassignment**
  - Working together
  - Similar task
Social Network Miner

- **Aim**: Monitor how individual process instances are routed between originators

- **Metrics**
  - Handover of work
  - Subcontracting
  - Reassignment
  - Working together
  - Similar task

Based on ordering relations derived from a log!
Social Network Miner: Example

---

**Actors:** John, Alex, Lucia, Peter, Mary

**Output Log #2**

**Clique Calculation**

- Minimum Set Size: 2
- Input dataset: C: \ Program Files

**Minimum 2 Cliques**

- 1: Lucia, Peter, Mary
- 2: John, Mary

**Actor-by-Actor Clique Co-Membership Matrix**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

**Hierarchical Clustering of Equivalence Matrix**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</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>0.833</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>0.667</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>
</tr>
<tr>
<td>0.500</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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**Example Network Visualization**

- Nodes: John, Alex, Lucia, Peter, Mary
- Edges indicate co-membership and strength

---

**Table: Actor Co-Membership**

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>Alex</th>
<th>Lucia</th>
<th>Peter</th>
<th>Mary</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Alex</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lucia</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Peter</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Mary</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
Plugin *Analyze Social Network*

- Better graphical view for the results of the Social Network Miner
- Includes different metrics to measure centrality of nodes
- Example: subcontracting
Which testers have never subcontracted work?

Which testers subcontract the most?
Outline

- Part I – Introduction to Process Mining
  - Context, motivation and goal
  - General characteristics of the analyzed processes and logs
  - Classification of Process Mining approaches

- Part II – Workflow discovery
  - Induction of basic Control Flow graphs
  - Other techniques (α-algorithm, Heuristic Miner, Fuzzy mining)

- Part III – Beyond control-flow mining
  - Organizational mining
  - Social net discovery
  - Extension algorithms

- Part IV – Evaluation and validation of discovered models
  - Conformance Check
  - Log-based property verification

- Part V – Clustering-based Process Mining
  - Discovery of hierarchical process models
  - Discovery of process taxonomies
  - Outlier detection in a process mining setting
Extension techniques

Enhance existing models with information discovered from logs

- The Decision Point Analysis plug-in can discover the “business rules” for the moments of choice in a process model
- The Performance Analysis with Petri Nets plug-in provides various KPIs w.r.t. the execution of processes
Decision Point Analysis: Main Idea

- Detection of data dependencies that affect the routing of process instances

**Motivations**
- Make tacit knowledge explicit
- Better understand the process model
Decision Point Analysis: Motivation

(amount > 500) AND (policyType = normal)

(amount <= 500) OR (policyType = premium)

status = approved

status = rejected

(amount = R)

clientID = String

policyType = normal | premium

status = approved | rejected

decision point

data modification

data dependency
Decision Point Analysis: Approach
Decision Point Analysis

1. Read a log + model
2. Identify the decision points in a model
3. Find out which alternative branch has been taken for a given process instance and decision point
4. Discover the rules for each decision point
5. Return the enhanced model with the discovered rules
Decision Point Analysis

1. Read a log + model
2. Identify decision points in a model
3. Find out which alternative branch has been taken for a given process instance and decision point
4. Discover the rules for each decision point
5. Return the enhanced model with the discovered rules

Which elements are the classes and which are the attributes?
### Step 4

#### Training examples for decision point "p0"

<table>
<thead>
<tr>
<th>amount</th>
<th>clientID</th>
<th>policyType</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>C567894938</td>
<td>premium</td>
<td>C</td>
</tr>
<tr>
<td>700</td>
<td>C938609223</td>
<td>normal</td>
<td>B</td>
</tr>
<tr>
<td>550</td>
<td>C135697567</td>
<td>normal</td>
<td>B</td>
</tr>
<tr>
<td>500</td>
<td>C568120443</td>
<td>normal</td>
<td>C</td>
</tr>
<tr>
<td>50</td>
<td>C493823084</td>
<td>normal</td>
<td>C</td>
</tr>
<tr>
<td>200</td>
<td>C945675110</td>
<td>premium</td>
<td>C</td>
</tr>
</tbody>
</table>

#### Discovered decision tree for point "p0"

```
amount <= 500
  | policyType
  | normal
  | premium
  | <= 500
  | > 500
```

- \( C \)
- \( B \)
Decision Point Analysis: Example in ProM
Decision Point Analysis: Example in ProM

Assessment:
- Node: Amount
- Condition: <= 500
- Branch 4.1: Check policy only/complete (10.0/1.0)
- Condition: > 500
- Node: PolicyType
- Condition: = premium
- Branch 4.1: Check policy only/complete (3.0)
- Condition: = normal
- Branch 4.2: Check all/complete (6.0)
Decision Point Analysis
Extension techniques

- Decision Miner
- Performance Analysis
Performance analysis: pattern visualization
Performance Analysis with Petri Nets

- **Motivation**
  - Provide different Key Performance Indicators (KPIs) relating to the execution of processes

- **Main idea**
  - Replay the log in a model and detect
    - Bottlenecks
    - Throughput times
    - Execution times
    - Waiting times
    - Synchronization times
    - Path probabilities etc
Bottlenecks – Throughput Times
Bottlenecks – Synchronization Times
Bottlenecks – Synchronization Times

What are these average synchronization times telling us?

1.3 minutes

20.8 minutes
Bottlenecks – Path Probabilities

What are these path probabilities telling us?
Performance Analysis with Petri Nets