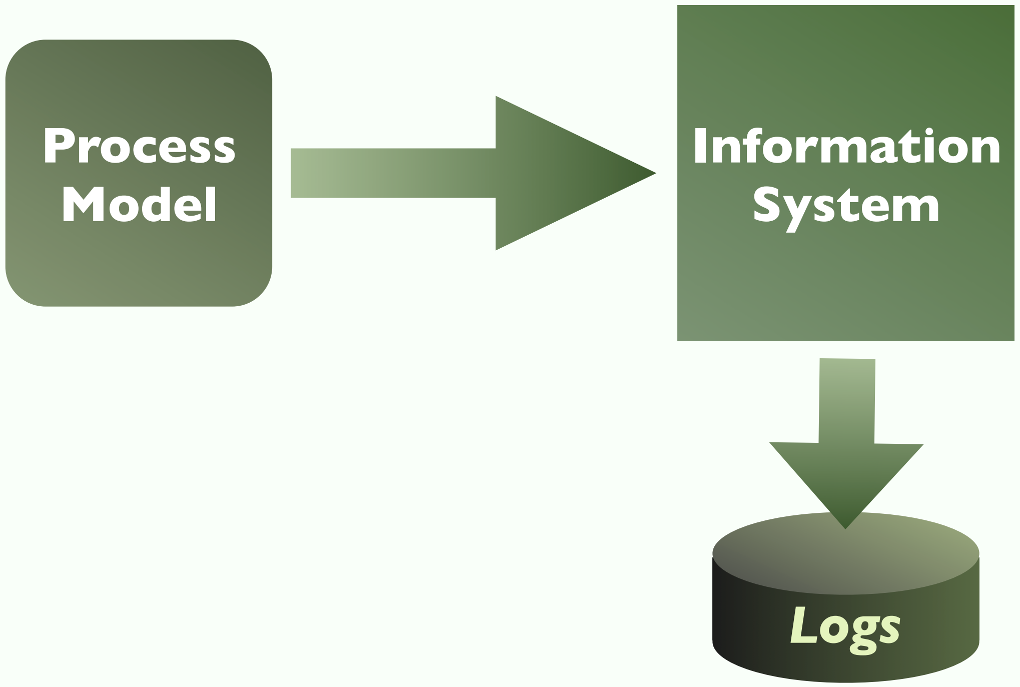
Process Mining: Using CPN Tools to Create Test Logs for Mining Algorithms

Ana Karla Alves de Medeiros and Christian W. Günther

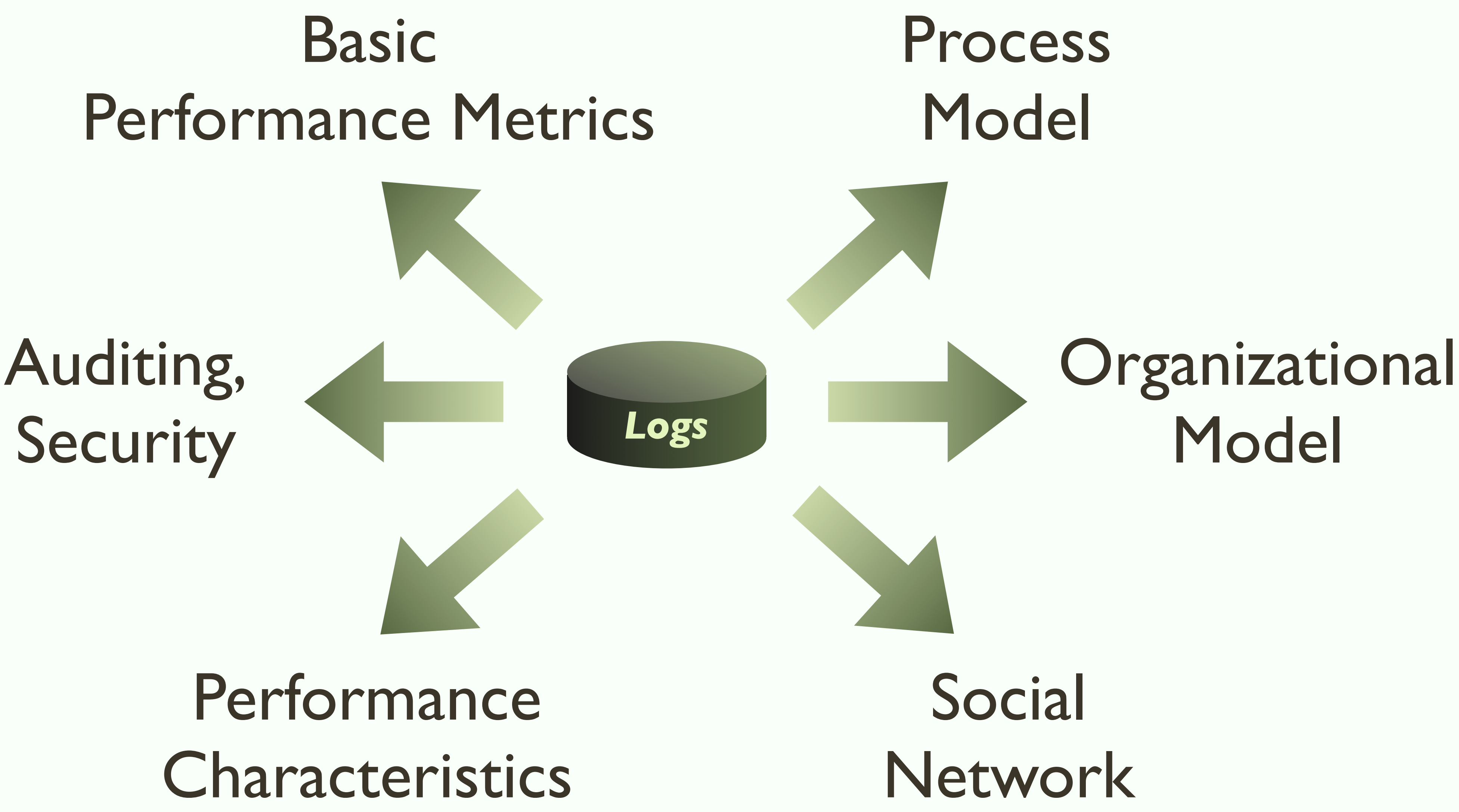
Outline

- Introduction Process Mining
- The need for synthetic logs
- MXML format
- Using the Logging Extensions
- Conclusion

Process Mining - *The general idea*



Process Mining - *The general idea*



Basic Performance Metrics

Process Model

Auditing, Security

Logs

Organizational Model

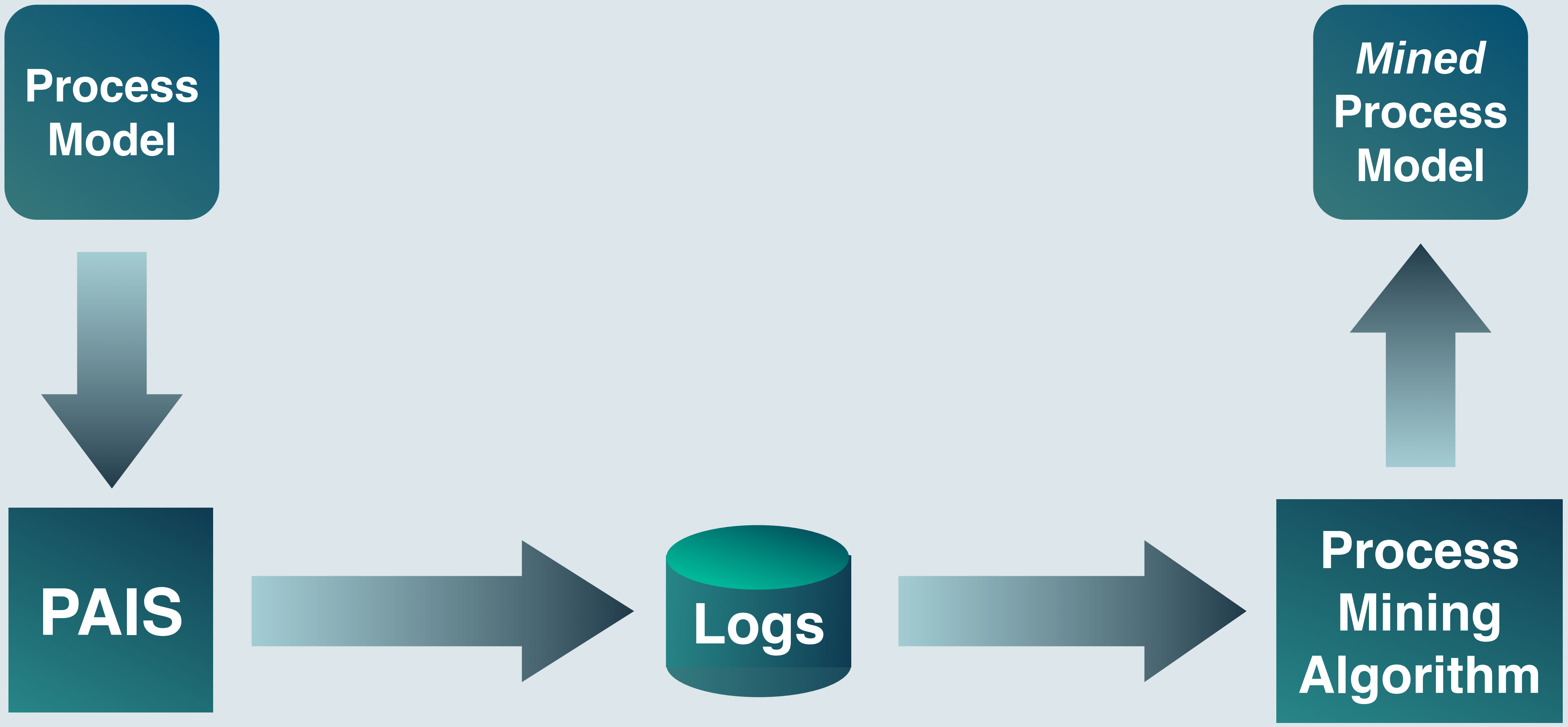
Performance Characteristics

Social Network

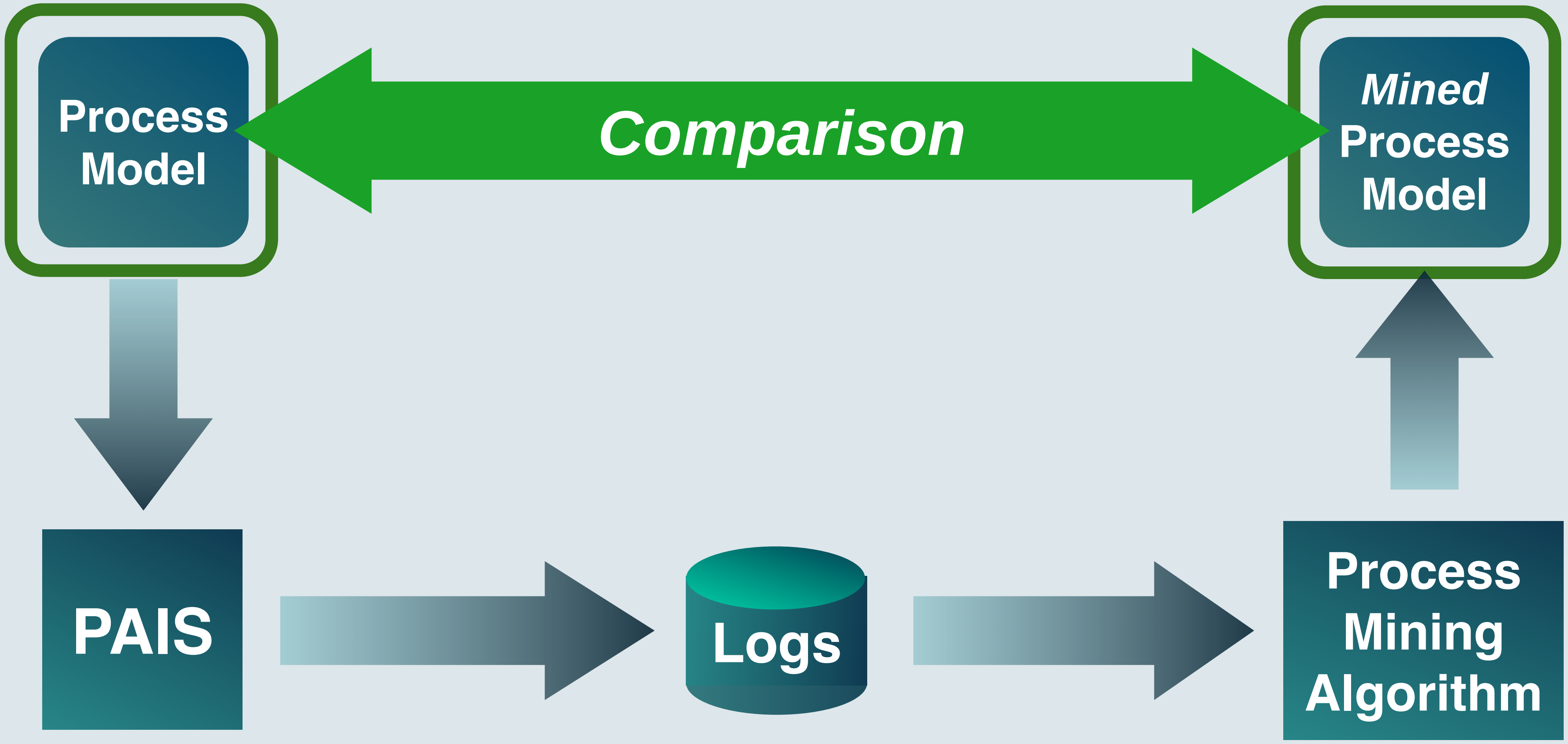
Process Mining - *...what is it good for?*

- How is a specific process ***really*** executed?
  - Discover ***discrepancies*** process model ↔ reality
  - Discover typical ***patterns*** in real-life operation
- Results can be used to:
  - ***Align*** organization and processes in a better way
  - Improve ***efficiency***, ***effectivity*** and ***quality***
  - ***Redesign*** processes based on solid information

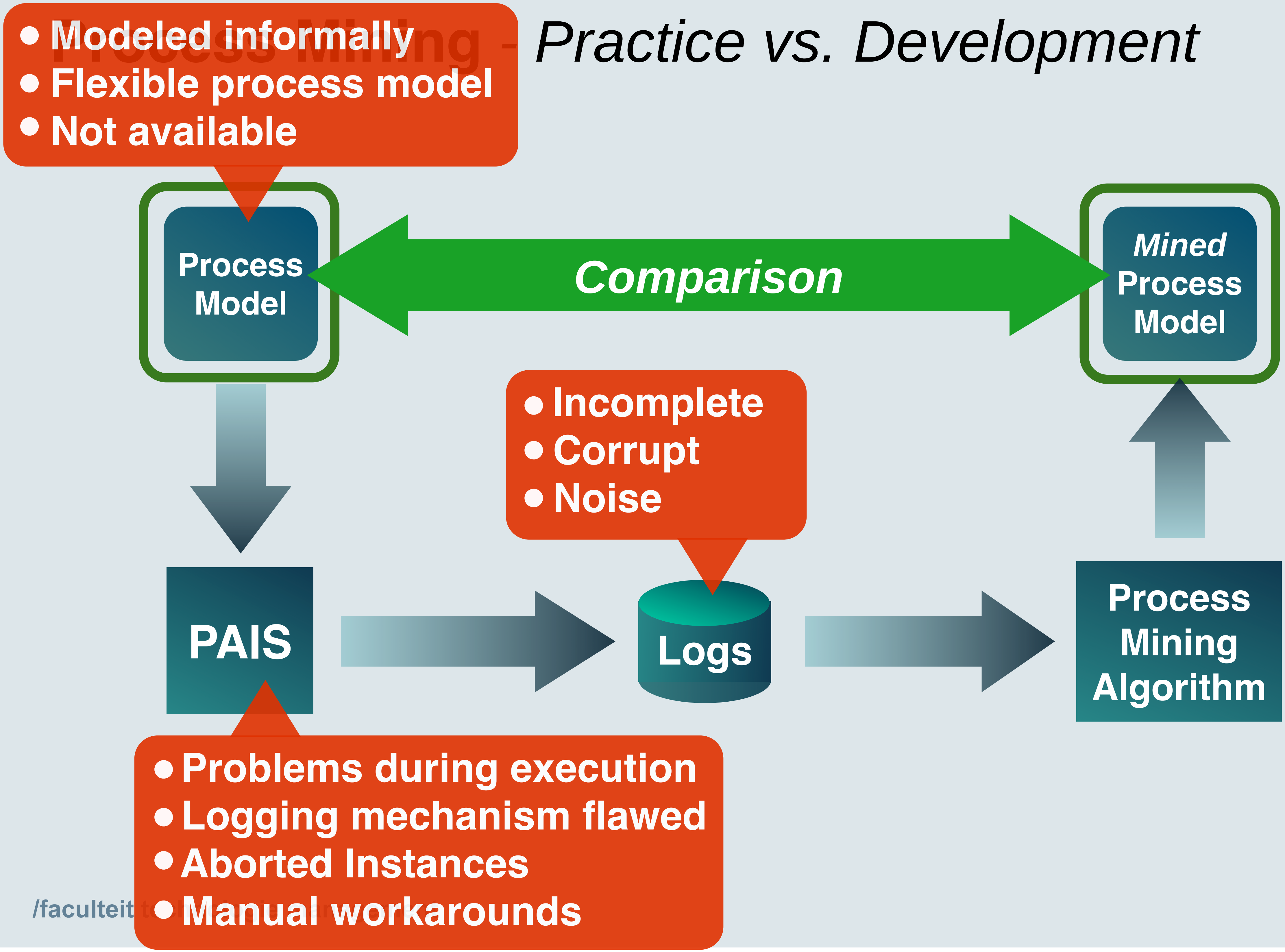
Process Mining - *Practice vs. Development*



Process Mining - *Practice vs. Development*



Process Mining - *Practice vs. Development*

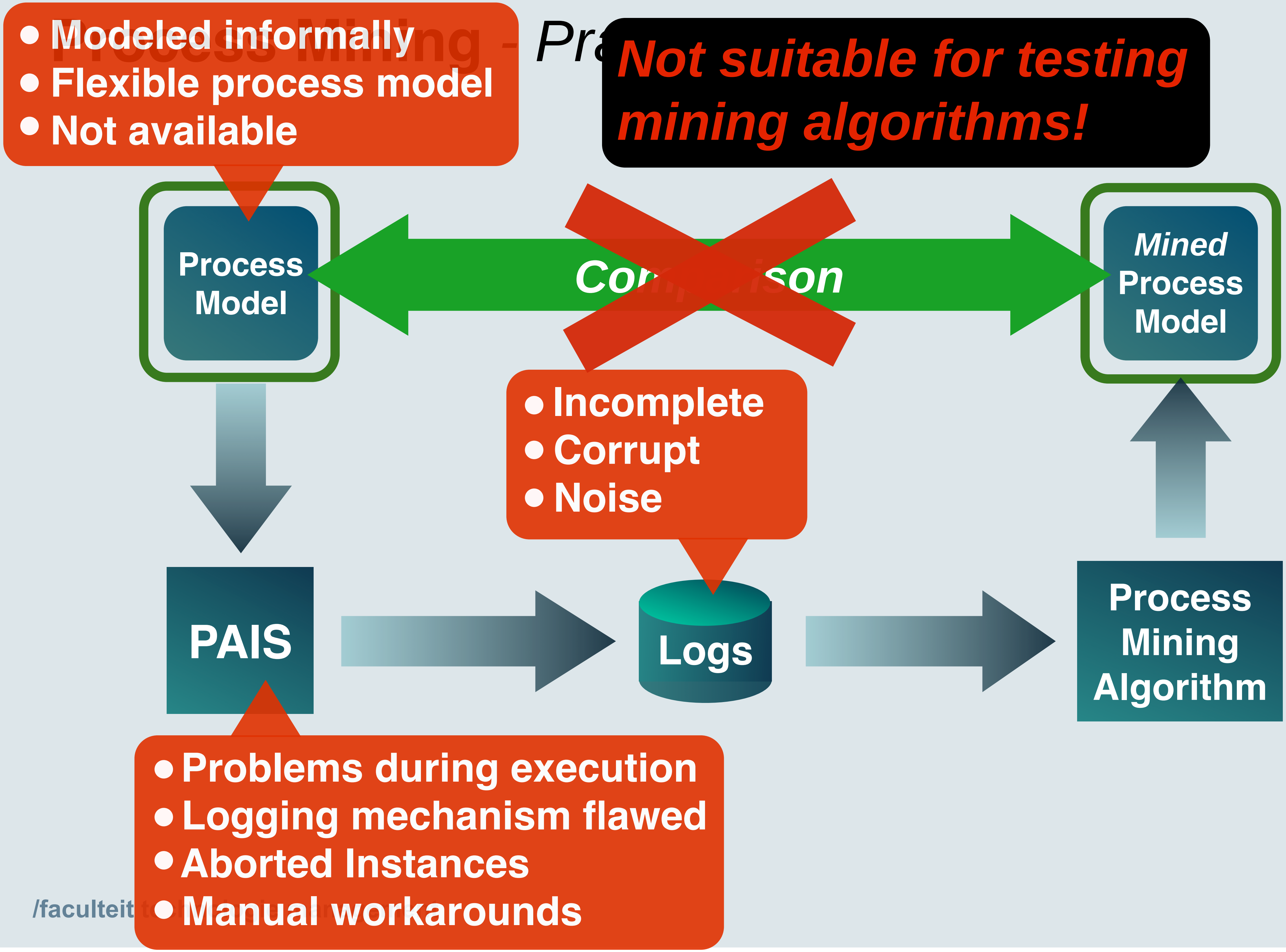


- Modeled informally
- Flexible process model
- Not available

- Incomplete
- Corrupt
- Noise

- Problems during execution
- Logging mechanism flawed
- Aborted Instances
- Manual workarounds

Process Mining - Pra



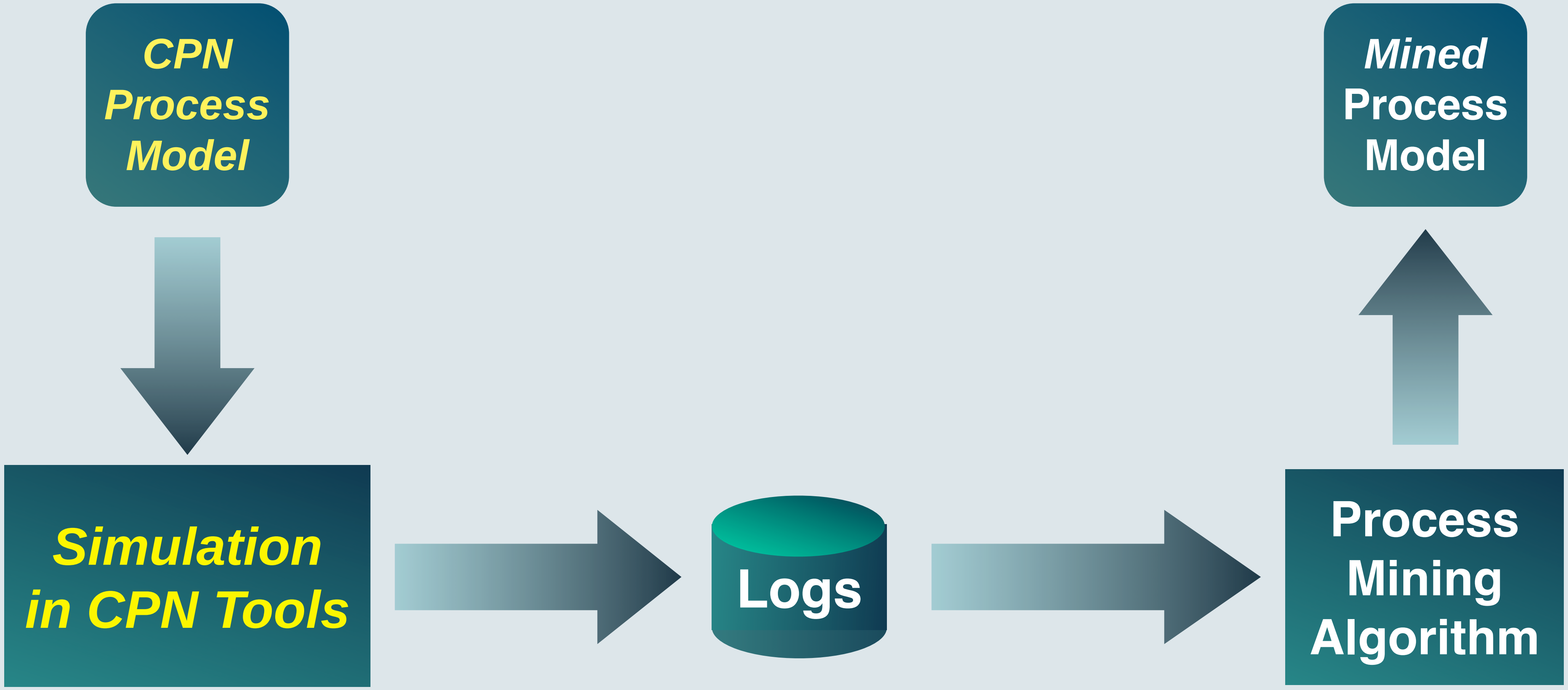
- Modeled informally
- Flexible process model
- Not available

Not suitable for testing mining algorithms!

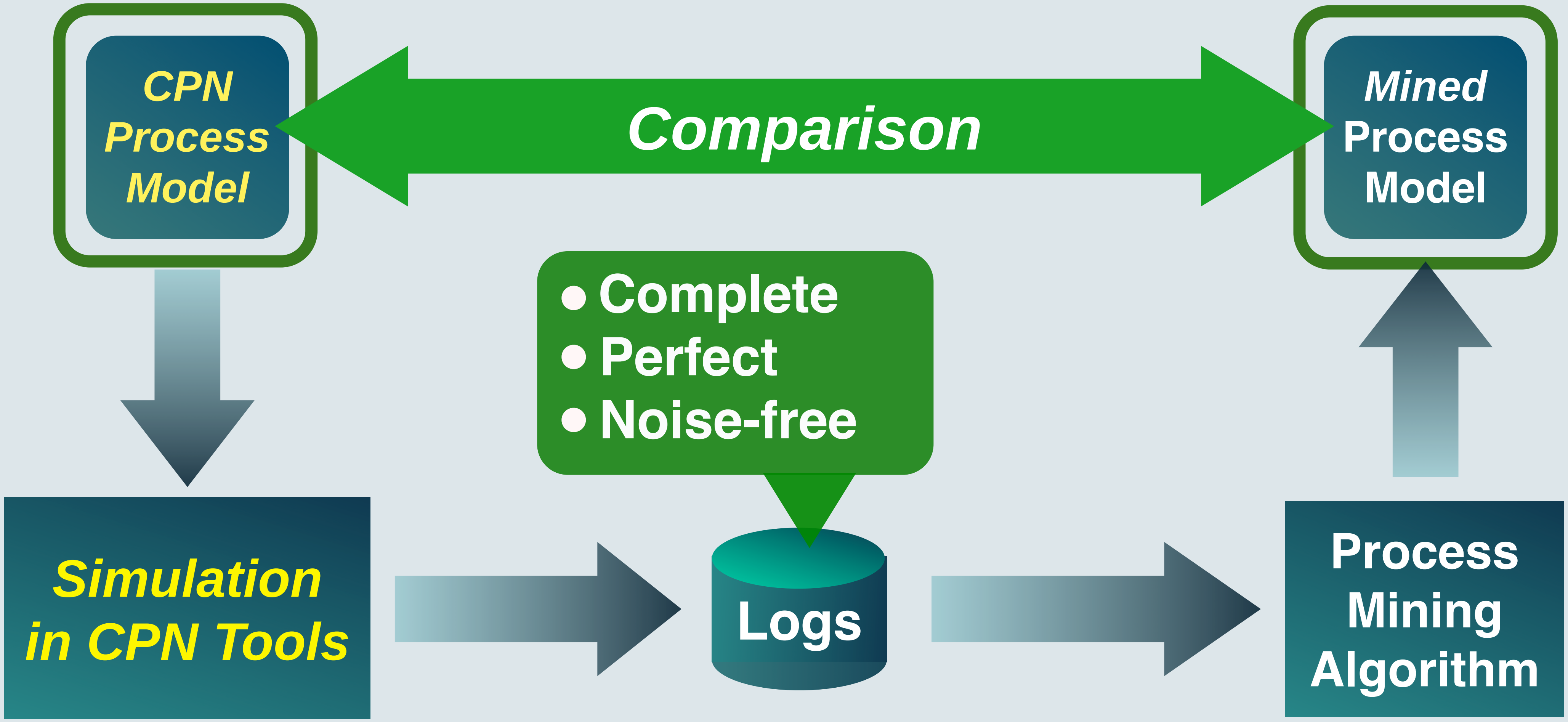
- Incomplete
- Corrupt
- Noise

- Problems during execution
- Logging mechanism flawed
- Aborted Instances
- Manual workarounds

Process Mining - *Practice vs. Development*

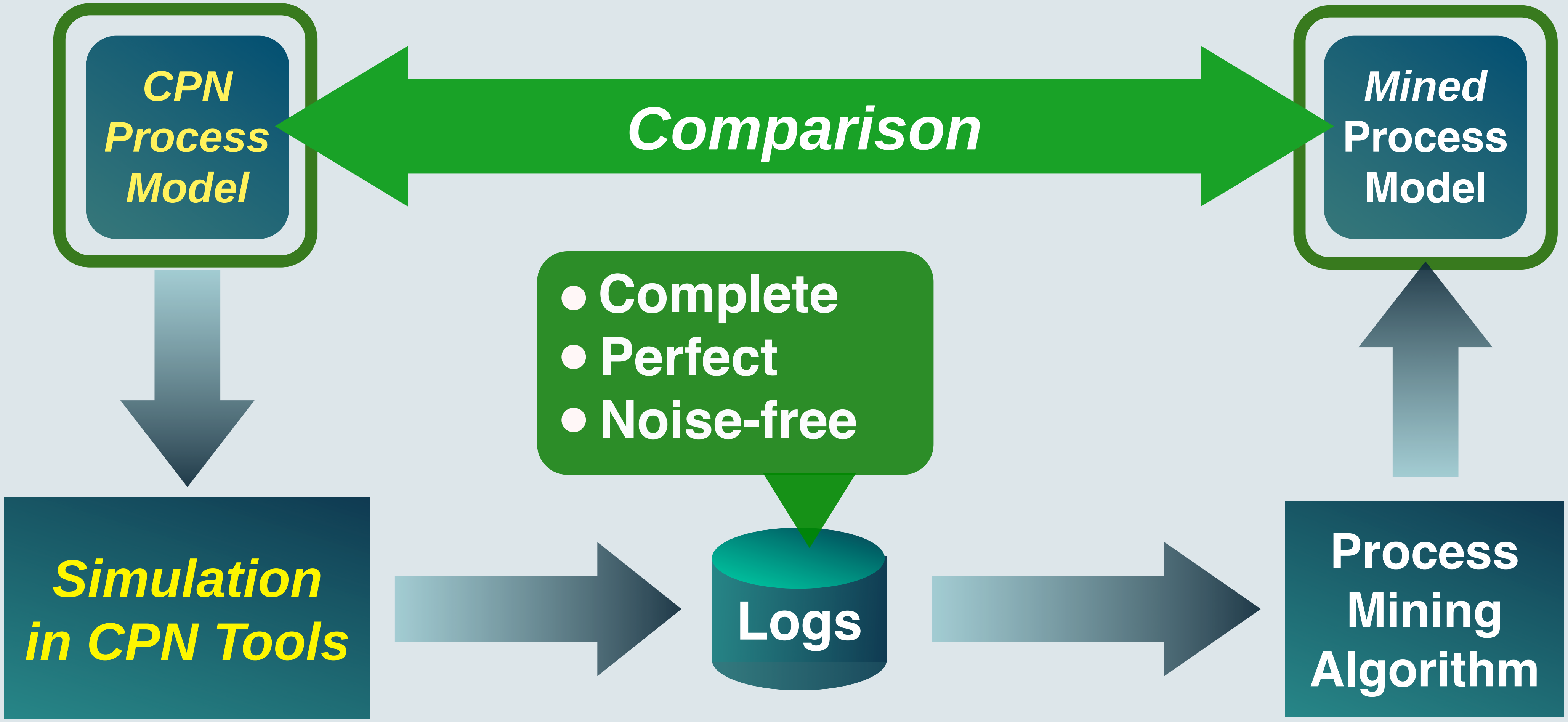


Process Mining - *Practice vs. Development*



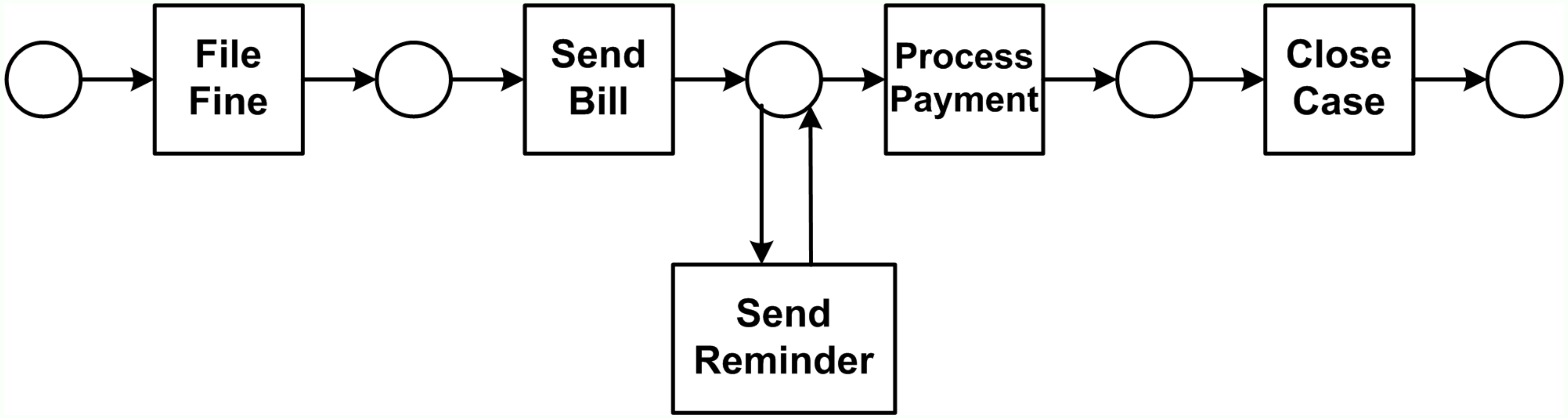
Process Mining -

Entirely controlled environment:
Perfect testbed for
Process Mining Algorithms!

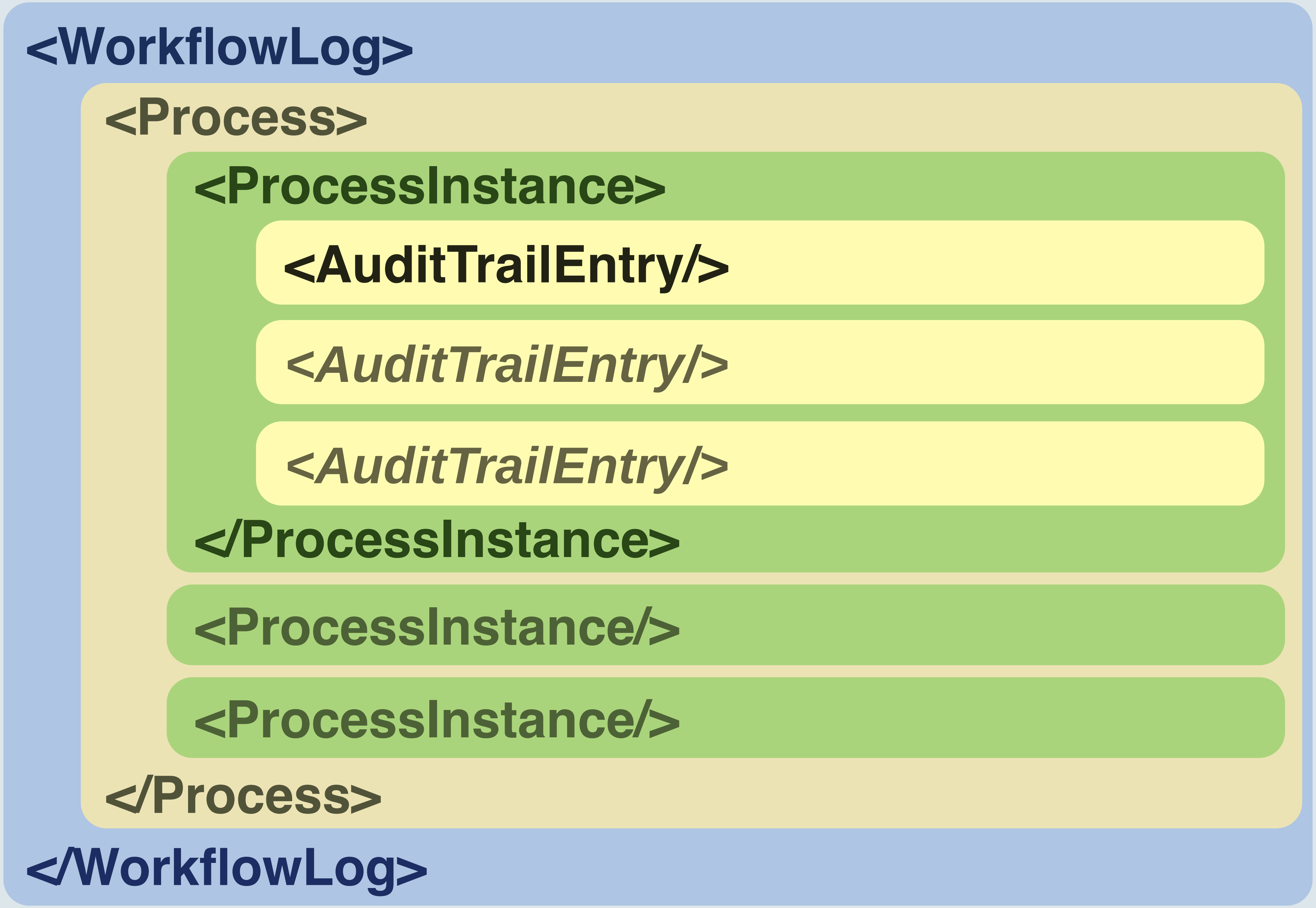


Logging a CPN Simulation

- Example process model: **Fine handling**



The MXML Format *- WF-Log, Process, Proc. Instance*



```
<WorkflowLog>
    <Process>
        <ProcessInstance>
            <AuditTrailEntry/>
            <AuditTrailEntry/>
            <AuditTrailEntry/>
        </ProcessInstance>
        <ProcessInstance/>
        <ProcessInstance/>
    </Process>
</WorkflowLog>
```

The MXML Format *- The Structure of an Audit Trail Entry*

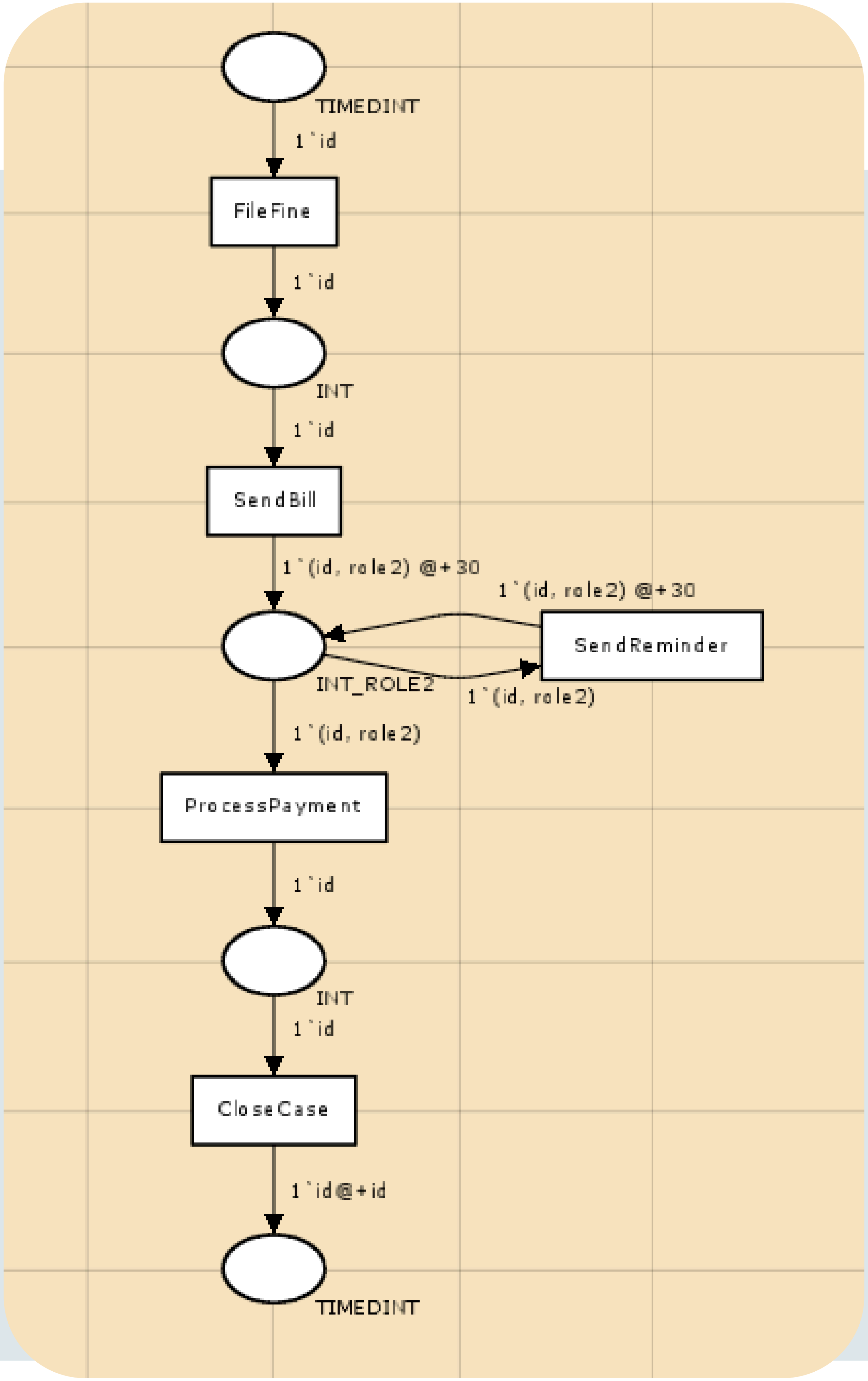
```
<AuditTrailEntry>
    <WorkflowModelElement/> Task A </Wf.M.E.>
    <EventType> complete </EventType>
    <TimeStamp> 2005-10-26T12:37:33... </TimeStamp>
    <Originator> John Doe </Originator>
    <Data>
      <Attribute name=”x”> 1 </Attribute>
      <Attribute name=”y”> whatever </Attribute>
    </Data>
</AuditTrailEntry>
```

MXML Logging Extensions *for CPN Tools*

- Define **2 global constants**
- Import **ML file** containing logging functions
- Call function ***createCaseFile*** to initialize new process instance
- Call function ***addATE*** to create new audit trail entry

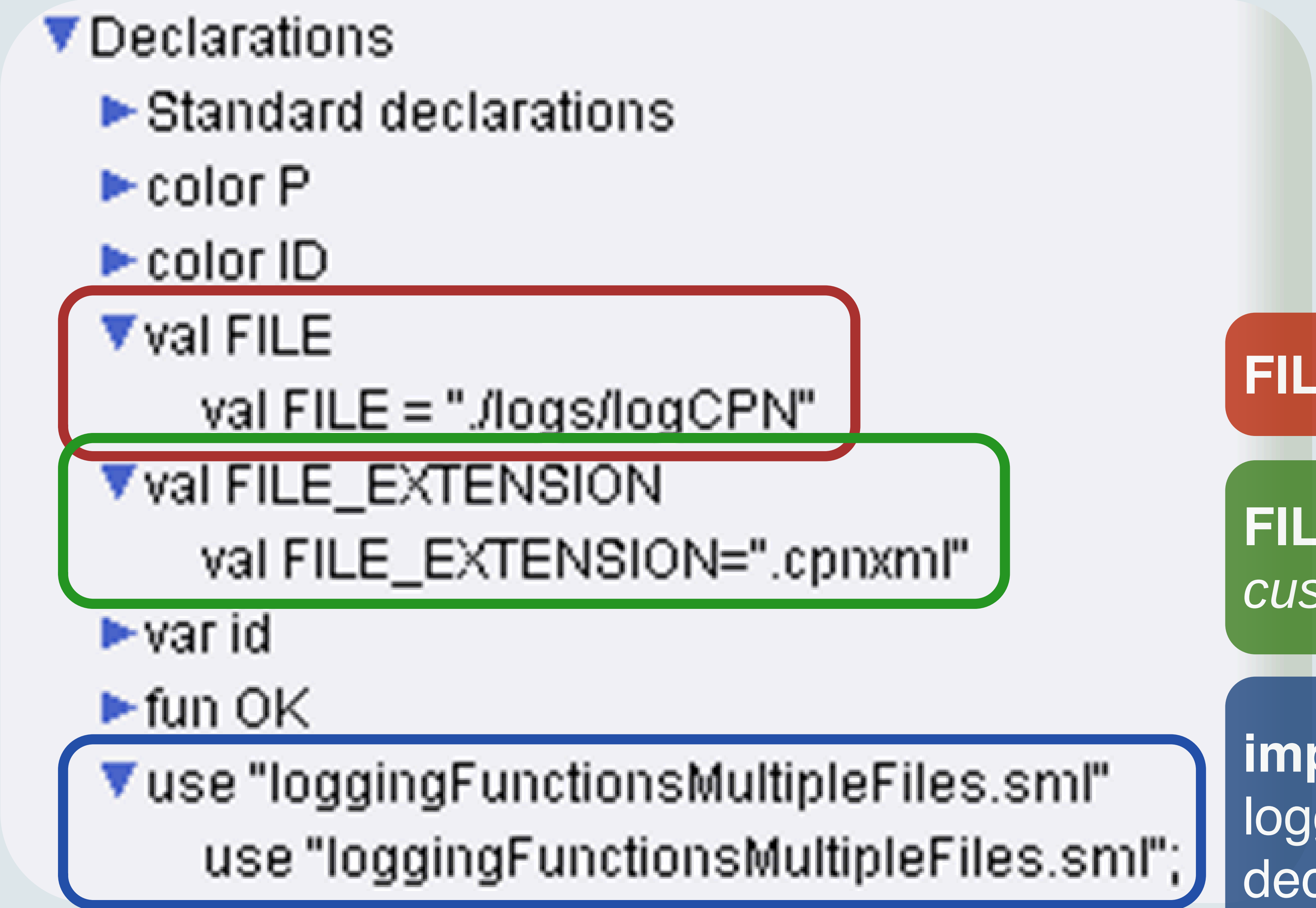
HOWTO create an MXML log from a CPN Model:

- If not already done: ***model the process in CPN Tools***



MXML Logging Extensions - *Control Variables*

- *Three declarations are necessary:*



```
▼ Declarations
    ▶ Standard declarations
    ▶ color P
    ▶ color ID
    ▼ val FILE
        val FILE = "./logs/logCPN"
    ▼ val FILE_EXTENSION
        val FILE_EXTENSION=".cpnxml"
    ▶ var id
    ▶ fun OK
    ▼ use "loggingFunctionsMultipleFiles.sml"
        use "loggingFunctionsMultipleFiles.sml";
```

FILE: *path and file prefix*

FILE_EXTENSION: *custom log file suffix*

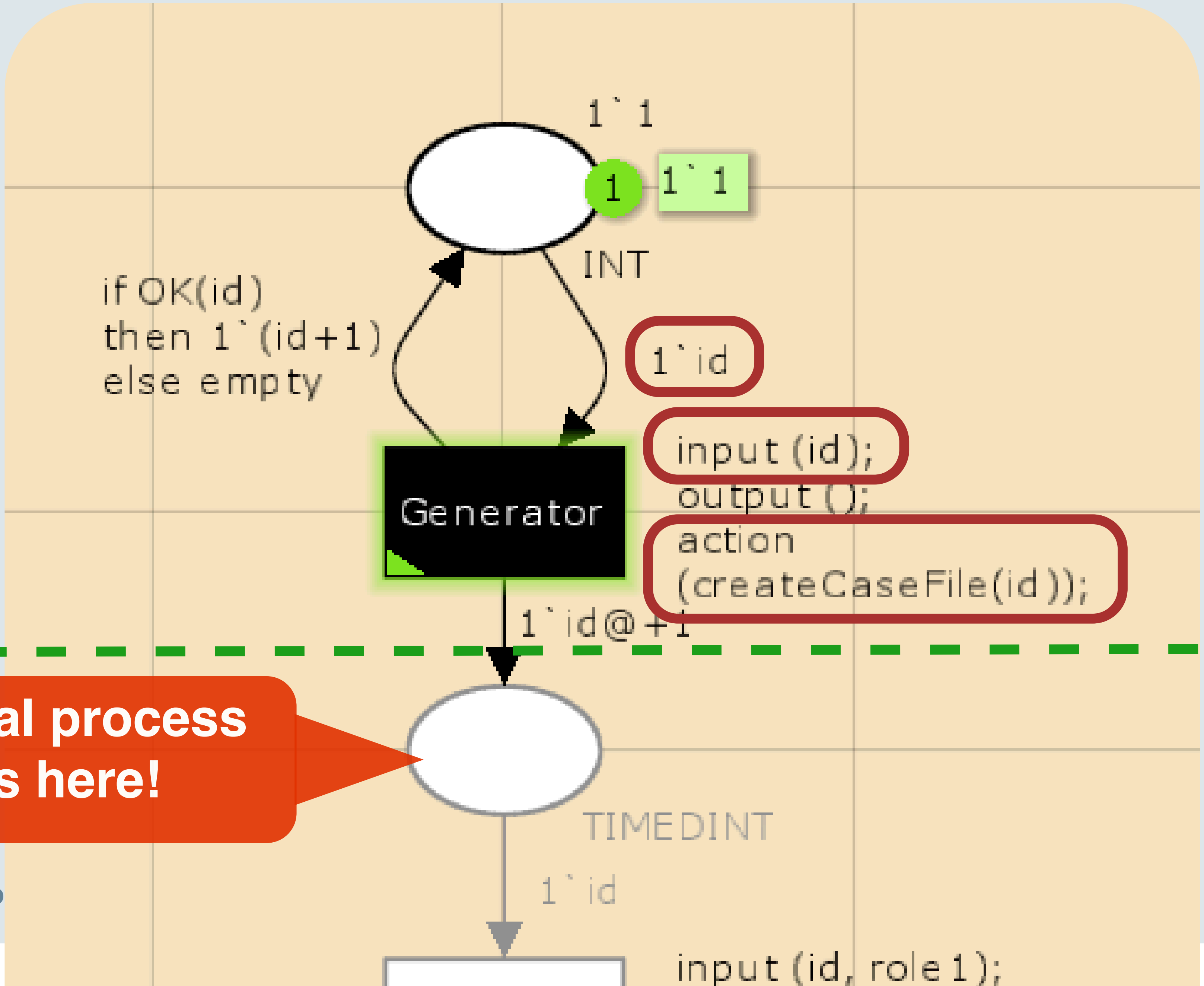
import ML file containing logging functions' declarations

MXML Logging Extensions - *Case File Creation*

- function ***createCaseFile***
  - one parameter **caseId** of type **Integer**
  - creates new file for recording a Process Instance
  - *File:* **<FILE> <caseId> <FILE_EXTENSION>**
- Example Use: **Case Generator**
  - extra transition at start of process
  - creates a defined number of tokens (=cases)
  - initializes log by calling **createCaseFile()**

MXML Logging Extensions - *Case File Creation*

Example application: *Case Generator*



```
input (id);
output ();
action
(createCaseFile(id));
```

```
input (id, role 1);
```

Actual process starts here!

MXML Logging Extensions - *Audit Trail Entries*

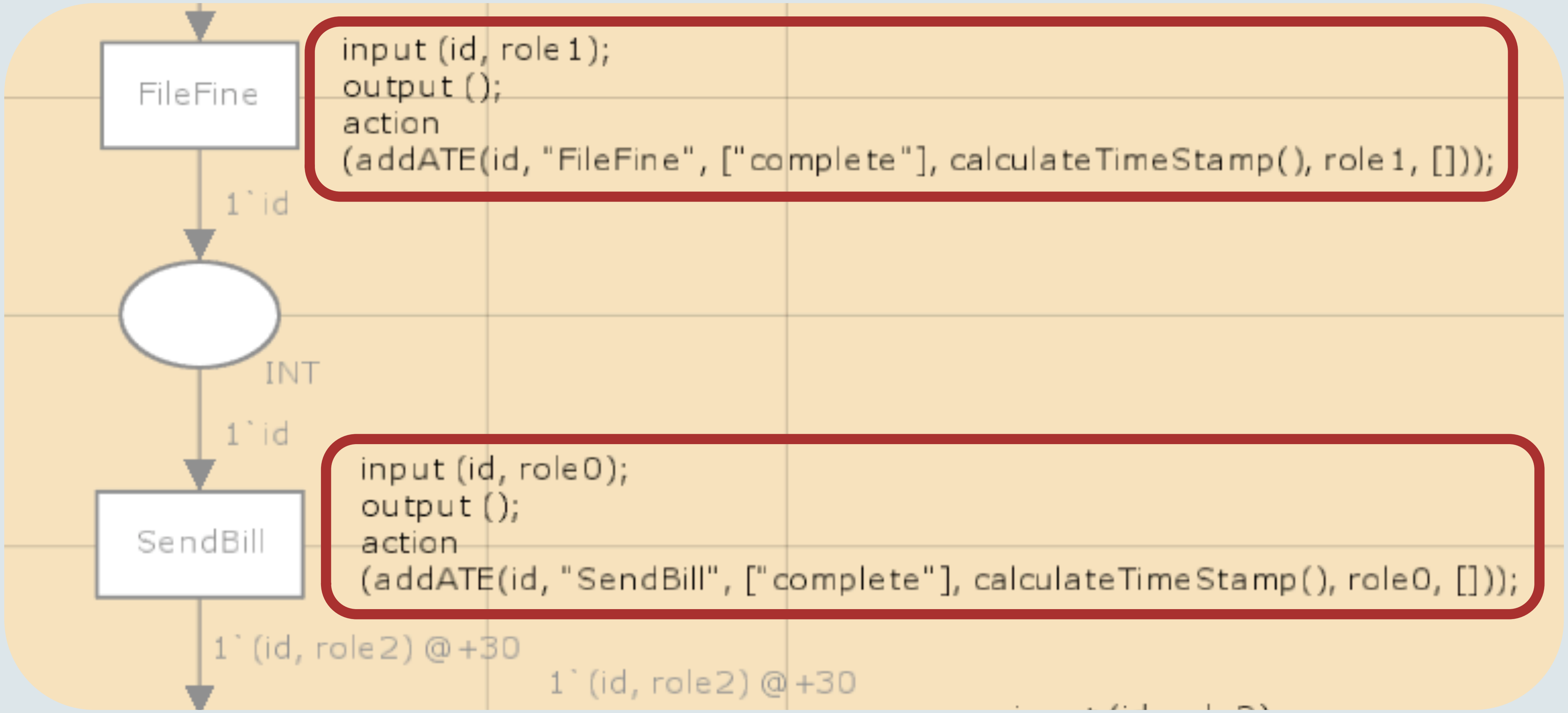
- function ***addATE***
  - ***Parameter list*** (*all MXML fields*)***:***
    - **caseId** (*Integer*): which ***file*** to append to
    - **workflowModelElement** *(String):* task name
    - **EventType** *(String):* complete, schedule, abort, ...
    - **TimeStamp** *(String):* when has the event occurred?
    - **Originator** *(String):* executing resource
    - **Data** *(List of Strings):* [name1, value1, name2, ...]
- Appends **audit trail entry** to case file with ***caseId***

MXML Logging Extensions - *Audit Trail Entries*

- (convenience) function ***calculateTimeStamp***
  - Takes ***no parameters***
  - Creates **MXML-compliant** timestamp from current **logical model time**

MXML Logging Extensions - *Audit Trail Entries*

- Usage of ***addATE***:

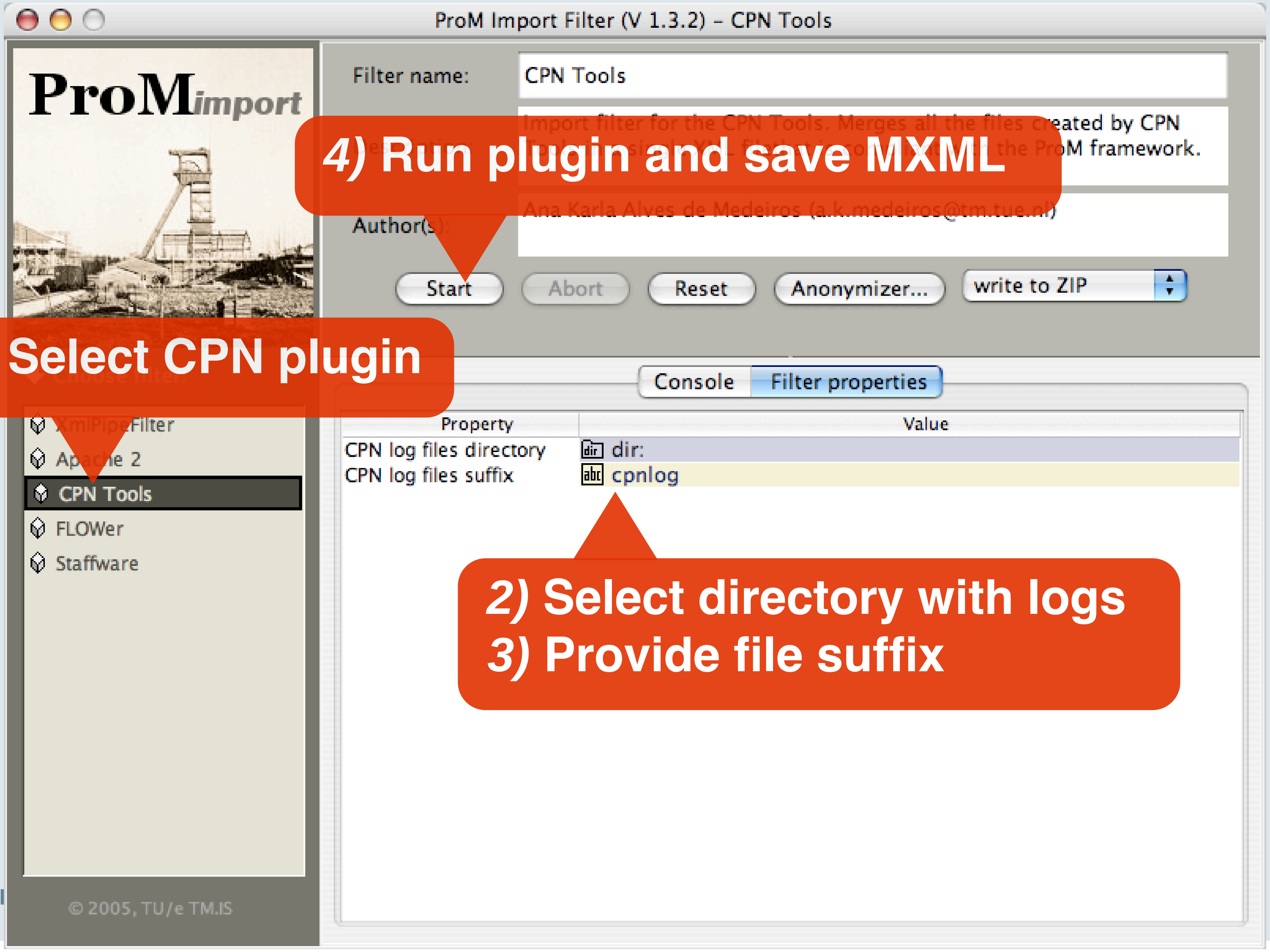


```
input (id, role1);
output ();
action
(addATE(id, "FileFine", ["complete"], calculateTimeStamp(), role1, []));
```

```
input (id, role0);
output ();
action
(addATE(id, "SendBill", ["complete"], calculateTimeStamp(), role0, []));
```

Aggregating Simulation Logs

- **Combines** all case files and writes clean **MXML**



ProM Import Filter (V 1.3.2) – CPN Tools

Filter name: CPN Tools

Description: Import filter for the CPN Tools. Merges all the files created by CPN Tools in single XML file that is compliant with the ProM framework.

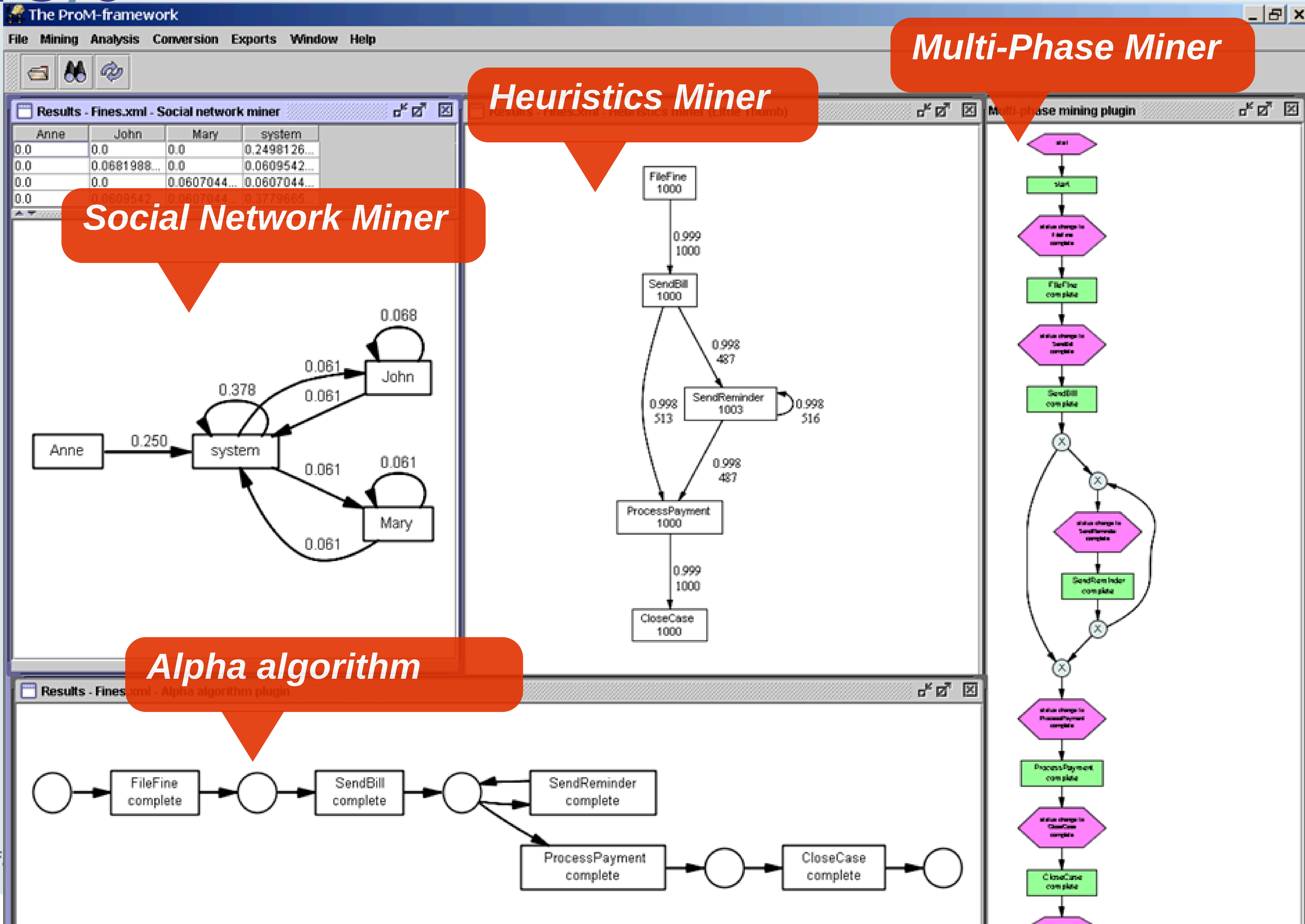
Author(s): Ana Karla Alves de Medeiros (a.k.medeiros@tm.tue.nl)

Property	Value
CPN log files directory	dir:
CPN log files suffix	cpnlog

1) Select CPN plugin

2) Select directory with logs
3) Provide file suffix

4) Run plugin and save MXML



Anne	John	Mary	system
0.0	0.0	0.0	0.2498126...
0.0	0.0681988...	0.0	0.0609542...
0.0	0.0	0.0607044...	0.0607044...
0.0	0.0609542...	0.0607044...	0.3779665...

Multi-Phase Miner

Heuristics Miner

Social Network Miner

Alpha algorithm

Conclusion

- Extending a model with Logging is ***straightforward***
    - Combined with the *ease of modeling in CPN Tools*, ***rapid creation of test logs***
- **Intended Usage:**
    - Synthesis of test logs for benchmarking mining algorithms
    - Creation of a test log repository
    - Testing mining algorithms on advanced models
        - *"...could we mine your models?"*