Classification Techniques for Process Analysis

Hind Chfouka[1], Andrea Corradini[1] and Roberto Guanciale[2]

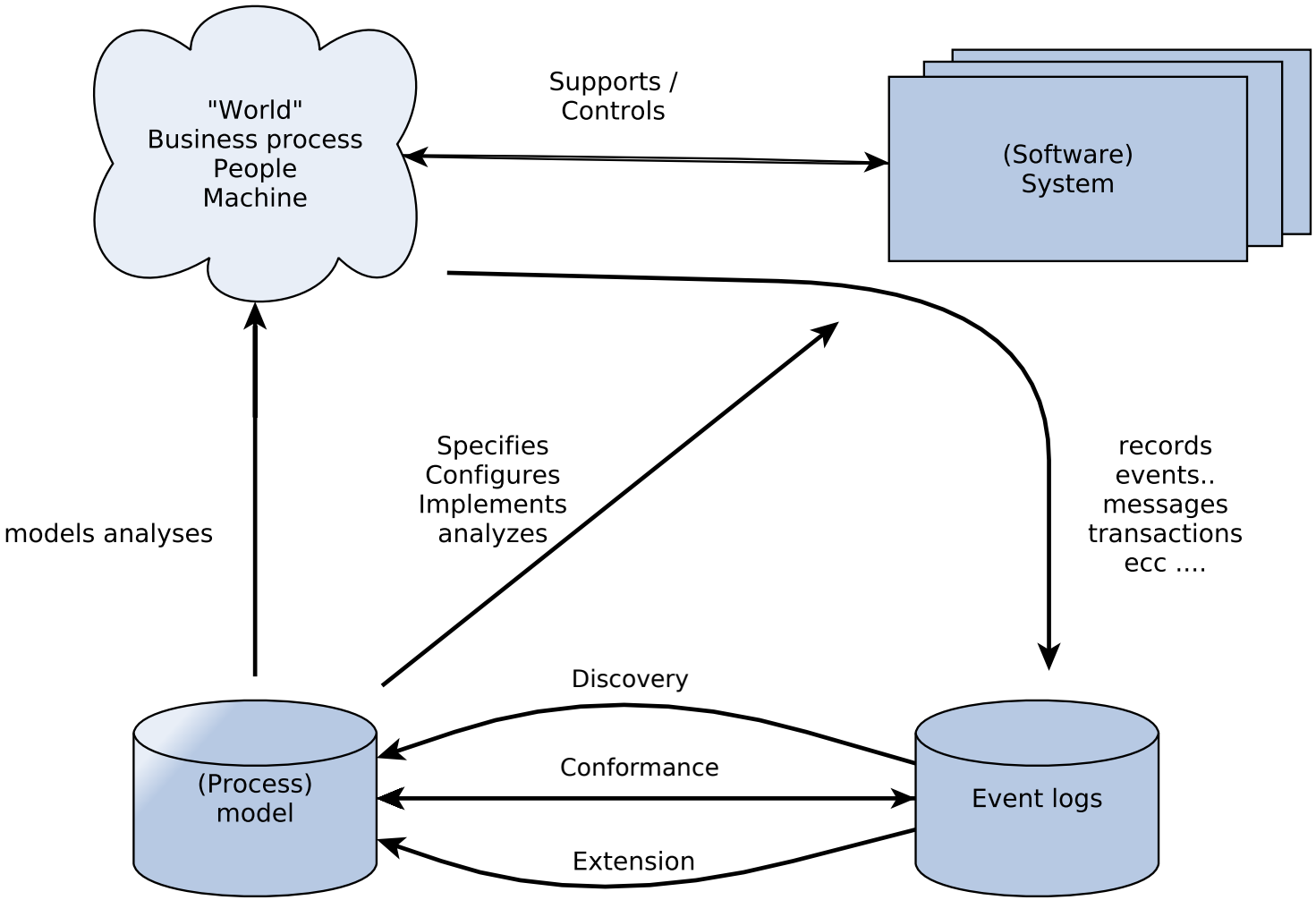
[1]Department of Computer Science, University of Pisa, Italy

[2]KTH Royal Institute of Technology, Stockholm, Sweden

December 6 2013, Turin

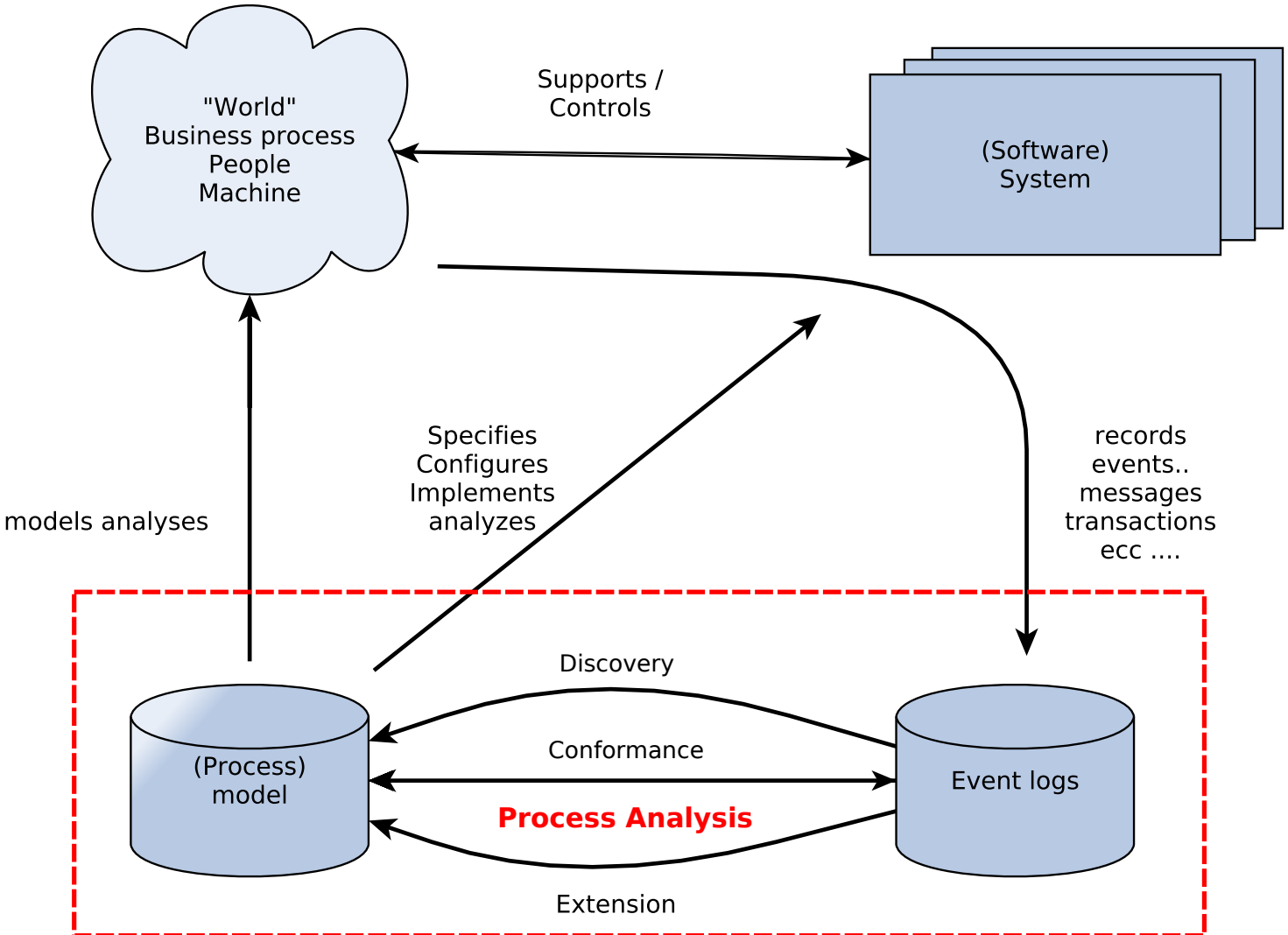
AI meets Business Processes Workshop
XIII Conference of the Italian Association for Artificial Intelligence

Context: Process Mining



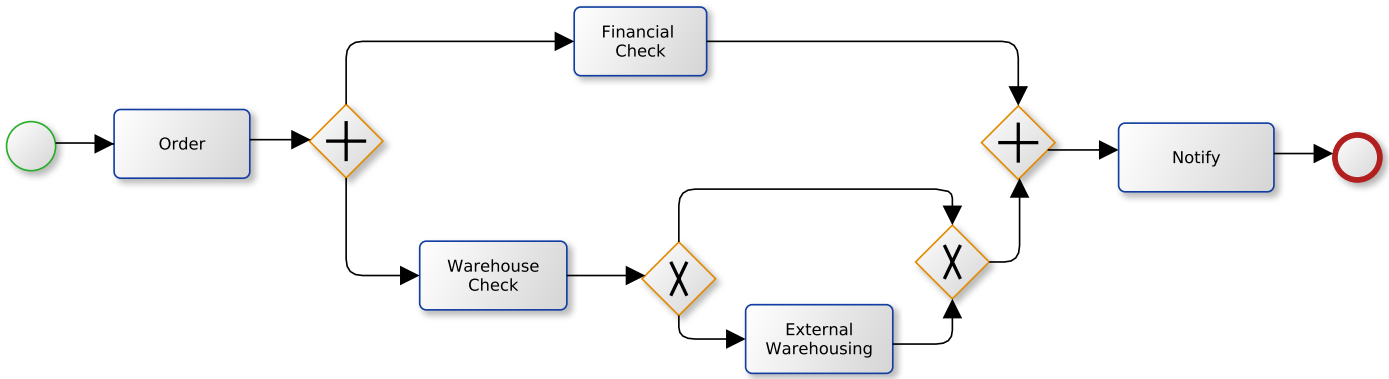
Focus: Process Analysis

We focus on: Process Analysis



An example: Sale Business Process

BPMN



Petri net

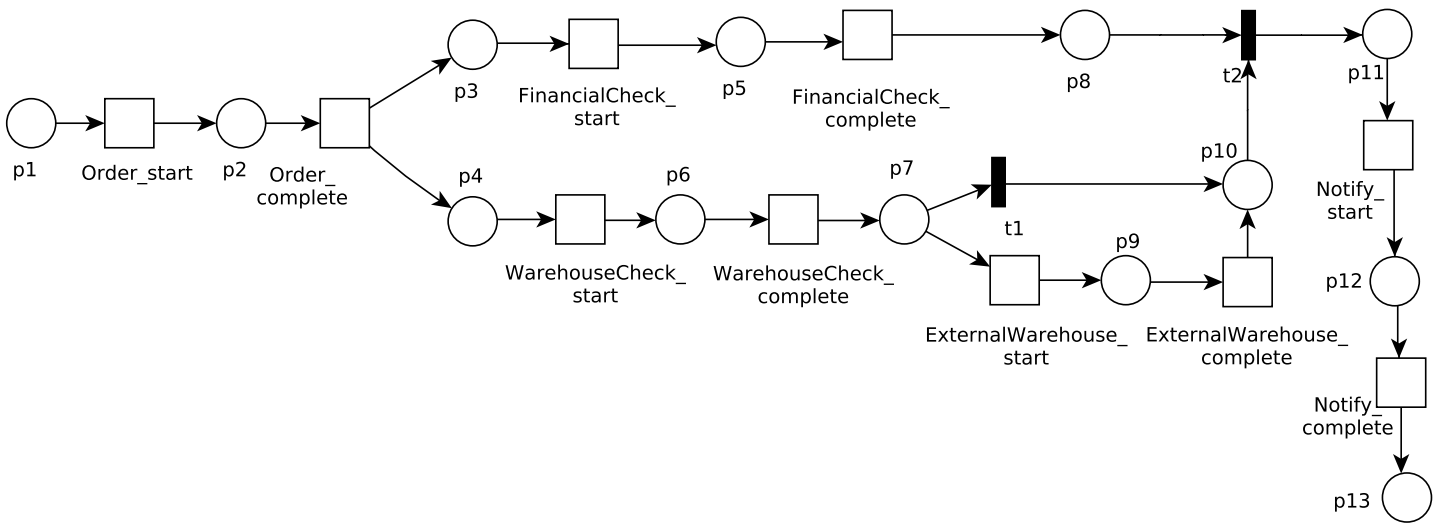
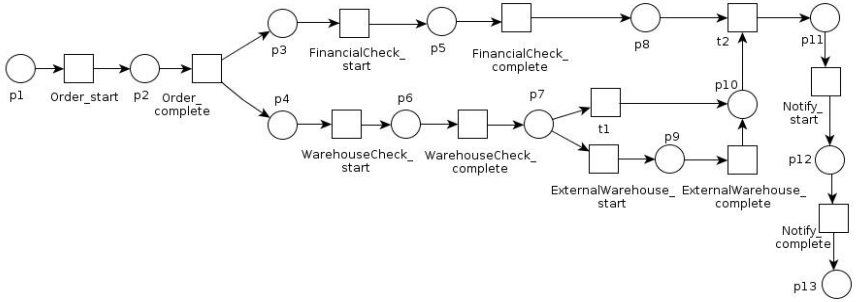


Figure: Sale Business Process

Process Analysis



T1	Order #1 start 9:30:50	Order #1 complete 10:15:00	WarhouseCheck #1 start 10:35:25	FinancialCheck #1 start 10:40:20	FinancialCheck #1 complete 12:00:20
T2	Order #1 start 9:30:50	Order #1 complete 10:15:00	WarhouseCheck #1 start 10:35:25	FinancialCheck #1 start 10:40:20	WarhouseCheck #1 complete 12:40:20
T3	Order #1 start 9:30:50	Order #1 complete 10:15:00	WarhouseCheck #1 start 10:35:25	FinancialCheck #1 start 10:40:20	WarhouseCheck #1 complete 12:40:20

Process Model
Ideal Behavior

Event logs
Real Behavior

Process Analysis

Conformance Checking

Performance Checking

Log Replay Algorithm

Assumptions

- Event $e = (a, t, atts)$: the event log building block
- Trace: a finite sequence of events $T[1], ..., T[n]$ ordered by timestamp. A trace represents a process instance
- Event log: a finite sequence of traces
- Each event of a trace can be mapped into a transition of the Petri net model

Algorithm

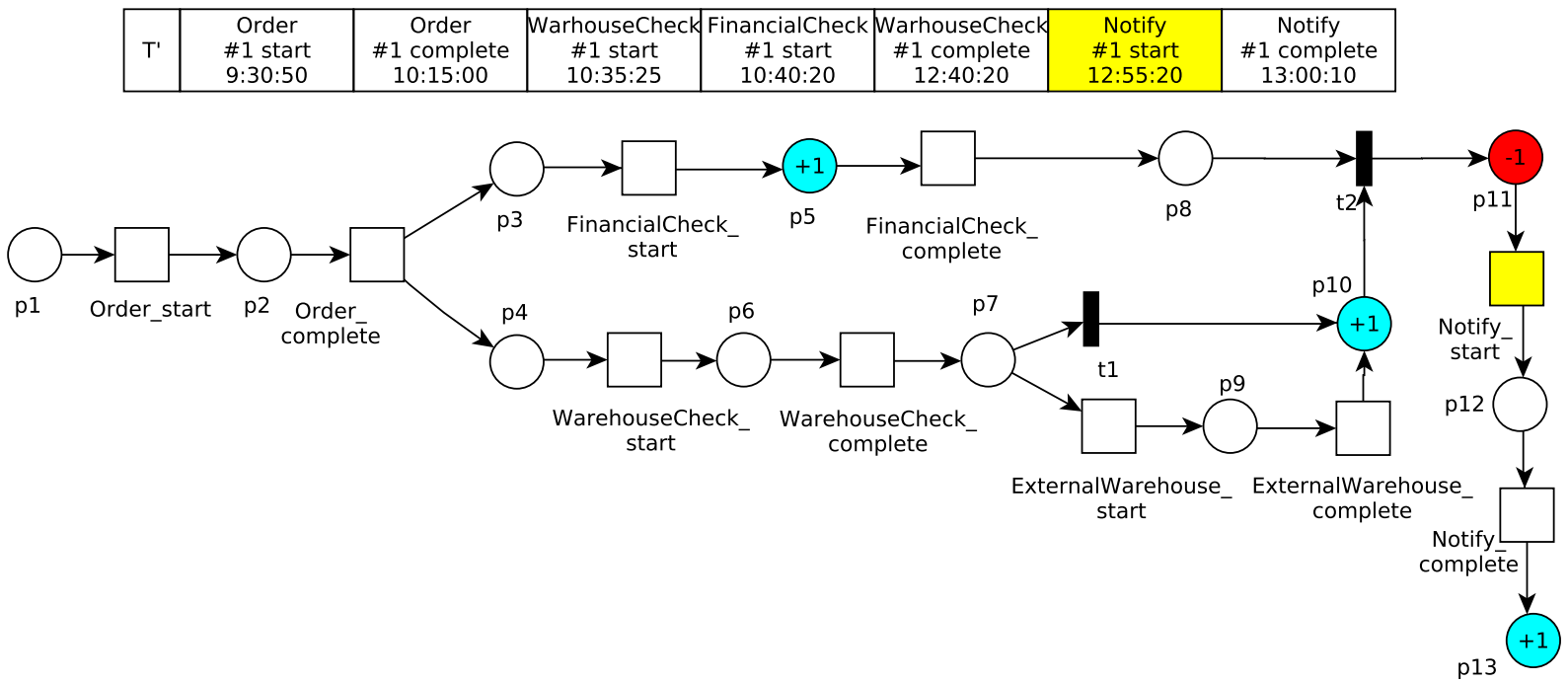
- Log replay: executes traces of an event log in a non-blocking way
  1. Starts with a token in the start place
  2. Extracts the top event of the log
  3. Fires the corresponding transition in the current marking of the net
     - If the transition is not enabled creates the missing tokens artificially
- Log replay results are used in conformance and performance checking

Conformance Analysis with log replay

Conformance checking = check if a trace is compliant with the Petri net model

Based on the log replay results...

- missing tokens are generated to mimic an event with a corresponding transition not enabled: this indicates a non-compliance to the model



ML for Process Analysis

Event logs are huge and rich of data: this encourages use of ML techniques

Several approaches exploiting Machine Learning techniques for the Business Process understanding:

- To extract the process model
- To find rules associated with a decision point
- To extract implicit information from the data process instances with Data Mining tools
- ....

Our idea

Exploiting ML techniques to discover how the process instance data may influence its conformance.

Classification for Conformance Checking

Conformance checking problem can be seen as a Classification problem:

Why?

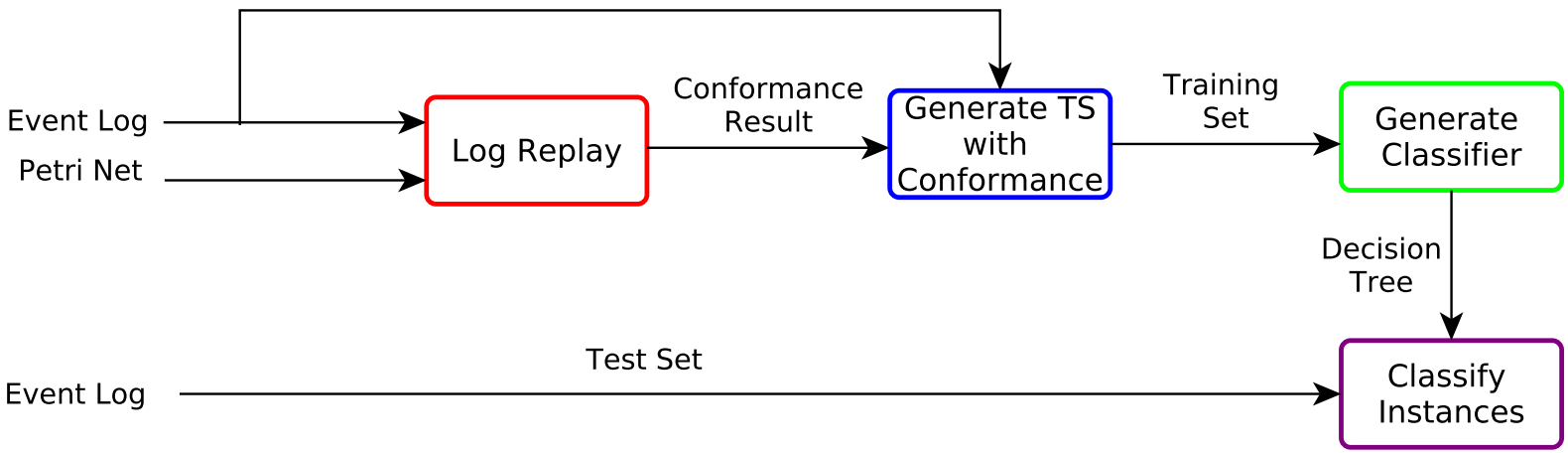
- To find out patterns of data in correspondence of which conformance errors occur
- To Predict conformance result at run-time.

How?

- Learning from previous analysis.
- Using an explicit classifier: Decision Tree.

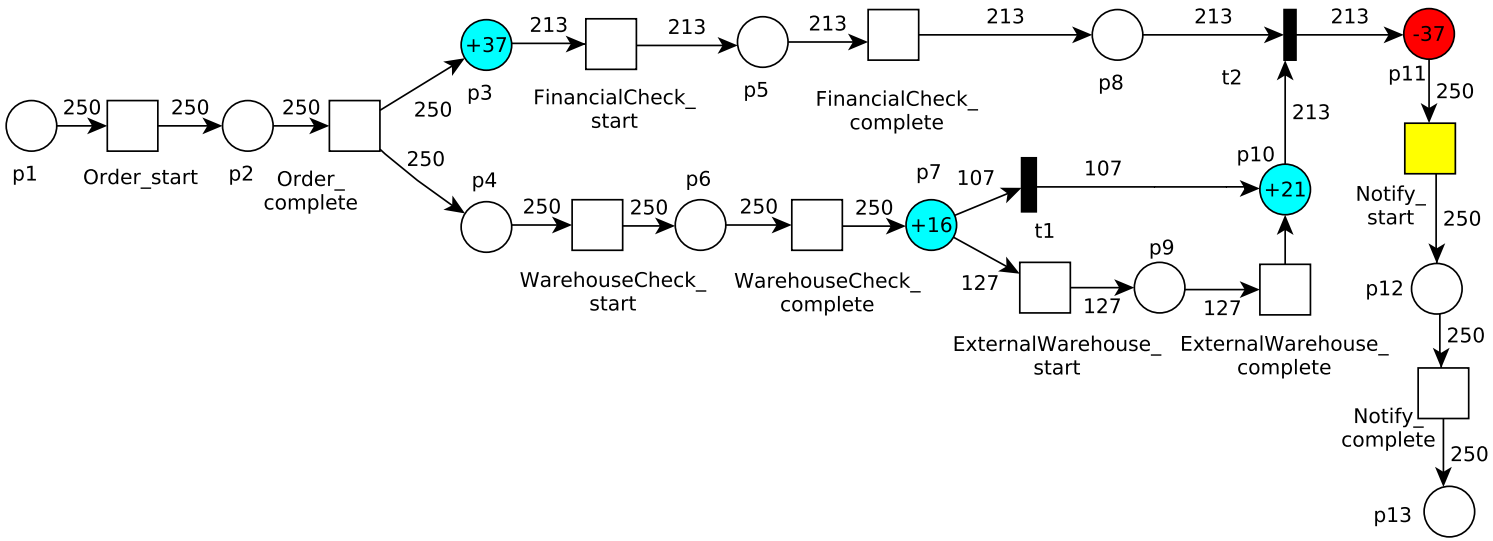
The approach

- Step 1: collecting a dataset based on event logs
- Step 2: dataset preprocessing including feature selection
- Step 3: building a decision tree model using ML algorithm
- Step 4: using the classifier to predict conformance result



Example: classification for the sale process

A Petri net summarizing the results of the log replay execution on an event log L:



37 instances of the process are not compliant with the sale policy since they did not execute the financial check activity.

Example: classification for the sale process (cont.)

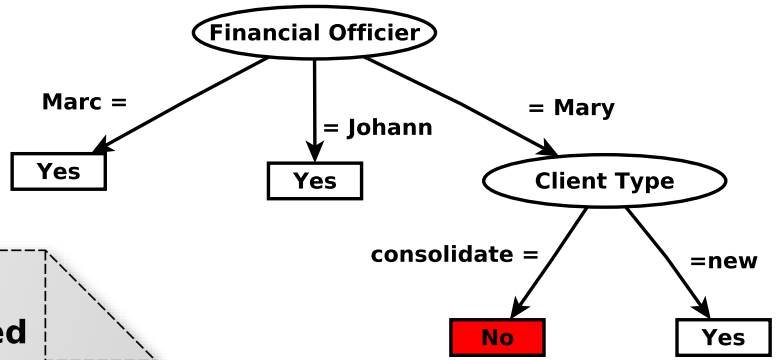
Log replay conformance results and process data extracted from the event log L enable:

- the construction of a dataset.
- the mining of a decision tree.

Data set for the sale business process: we focus on activities actors

Order Identifier	Client Identifier	Client Type	Sales Manager	Financial Officer	Warehouseman	Supplying Responsible	OrderResult	Conformance
1	17	consolidate	Marco	Mary	Alex	Gianni	positive	no
2	15	new	Anna	Johann	Roberto	Mario	positive	yes
...	...	....	...	...	...	...	...	...

The resulting decision tree:



Rule

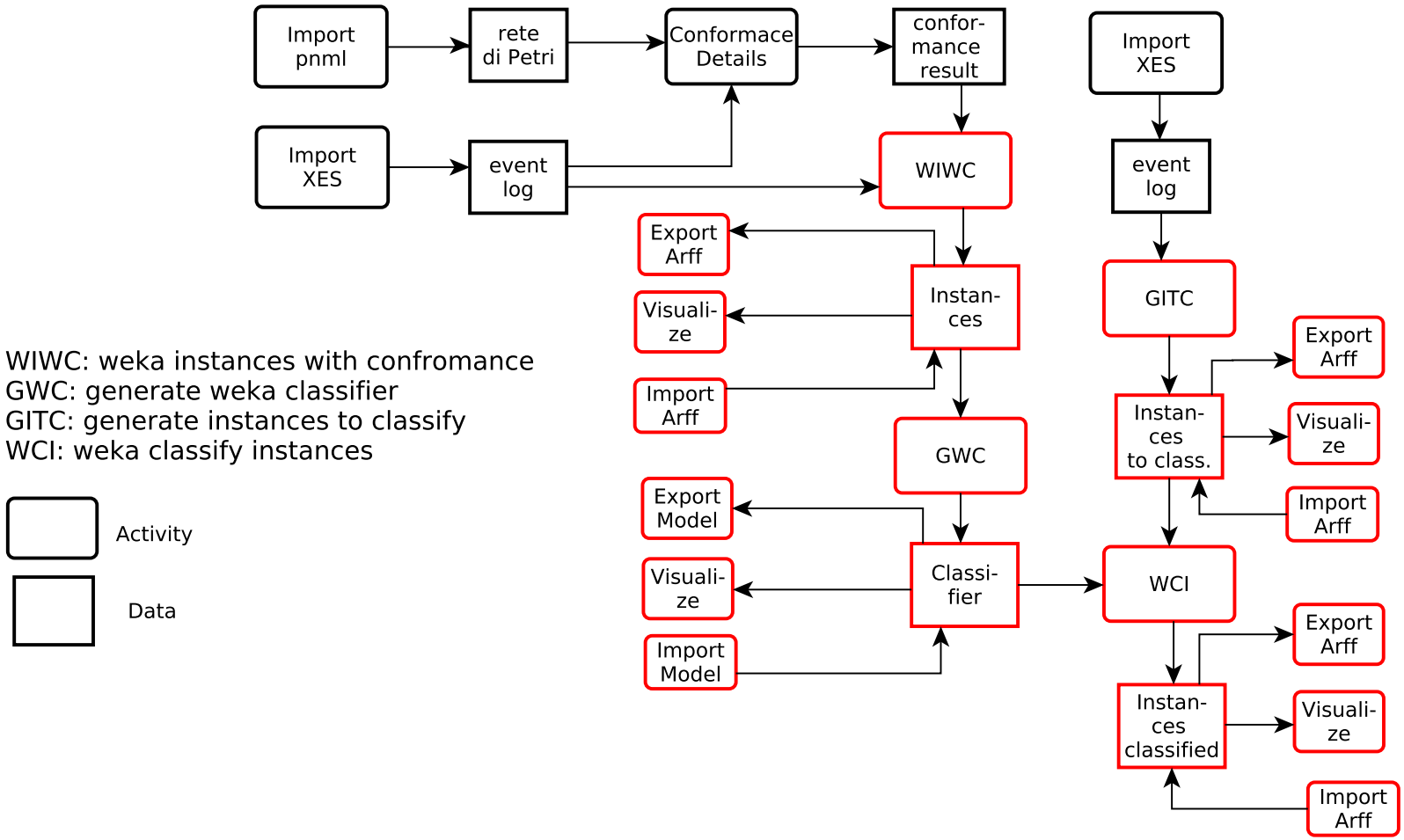
Orders issued by consolidated clients and handled by Mary may not respect the standard sale procedure.

Framework ProM6

ProM

- ProM is an extensible framework that supports a large variety of process mining and analysis techniques.
- It is a modular software implemented in Java and distributed under GNU Public License (GPL).
- ProM is a project of Process Mining Group of Eindhoven Technical University, Netherlands.

Framework for the analysis



Conclusion and future works

Conclusions:

- Preliminary research aimed at applying ML techniques in the Process Analysis.
- Extension to performance checking.
- Experimentation done only with synthetic data.

Future Work:

- Experimentations with real events logs.
- Exploration of a new technique of conformance checking based on event log alignment.