Universiteit Leiden

Business Intelligence & Process Modelling

Frank Takes

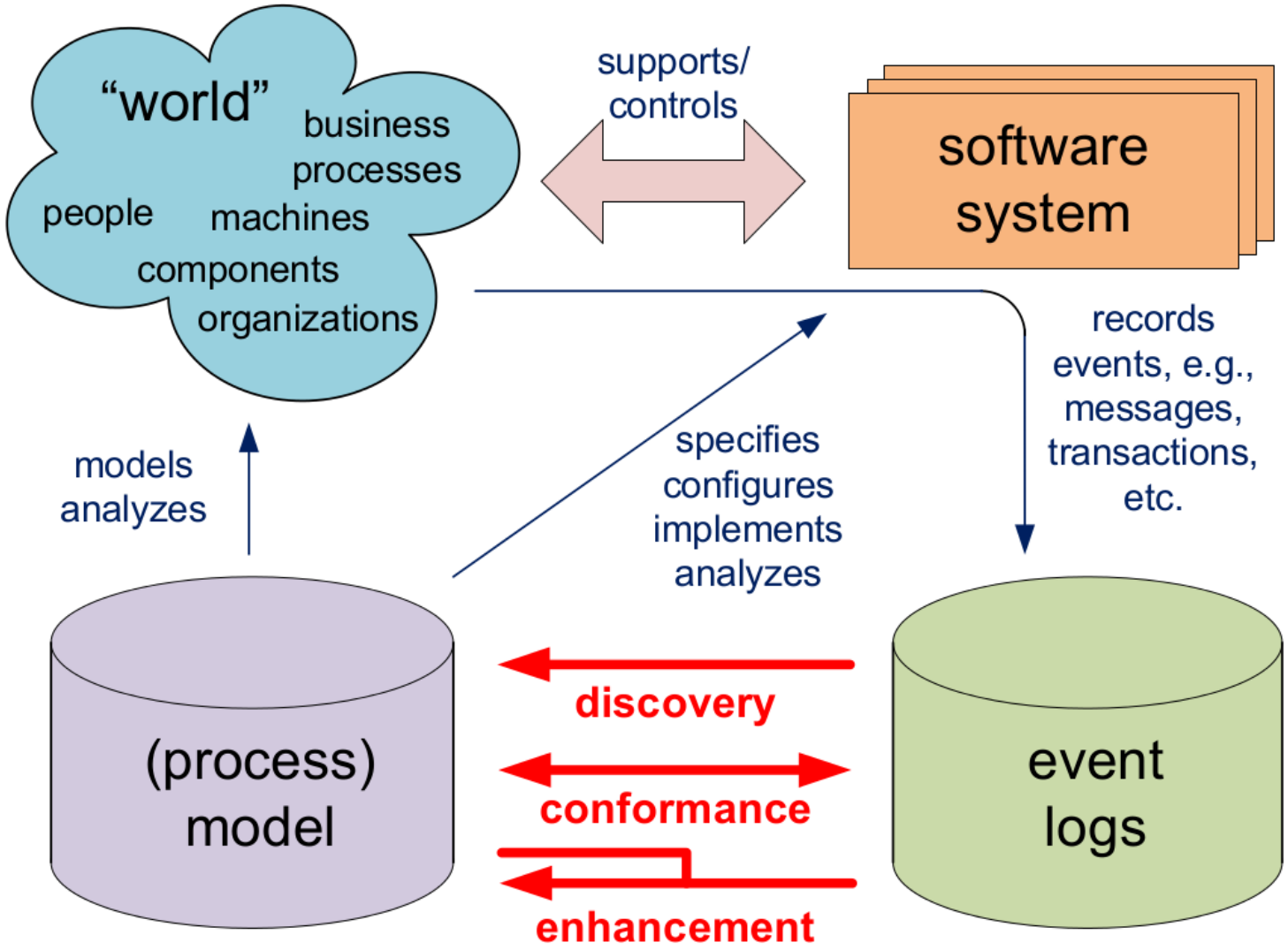
Universiteit Leiden

Lecture 9 — Process Modelling & BPMN & Tooling

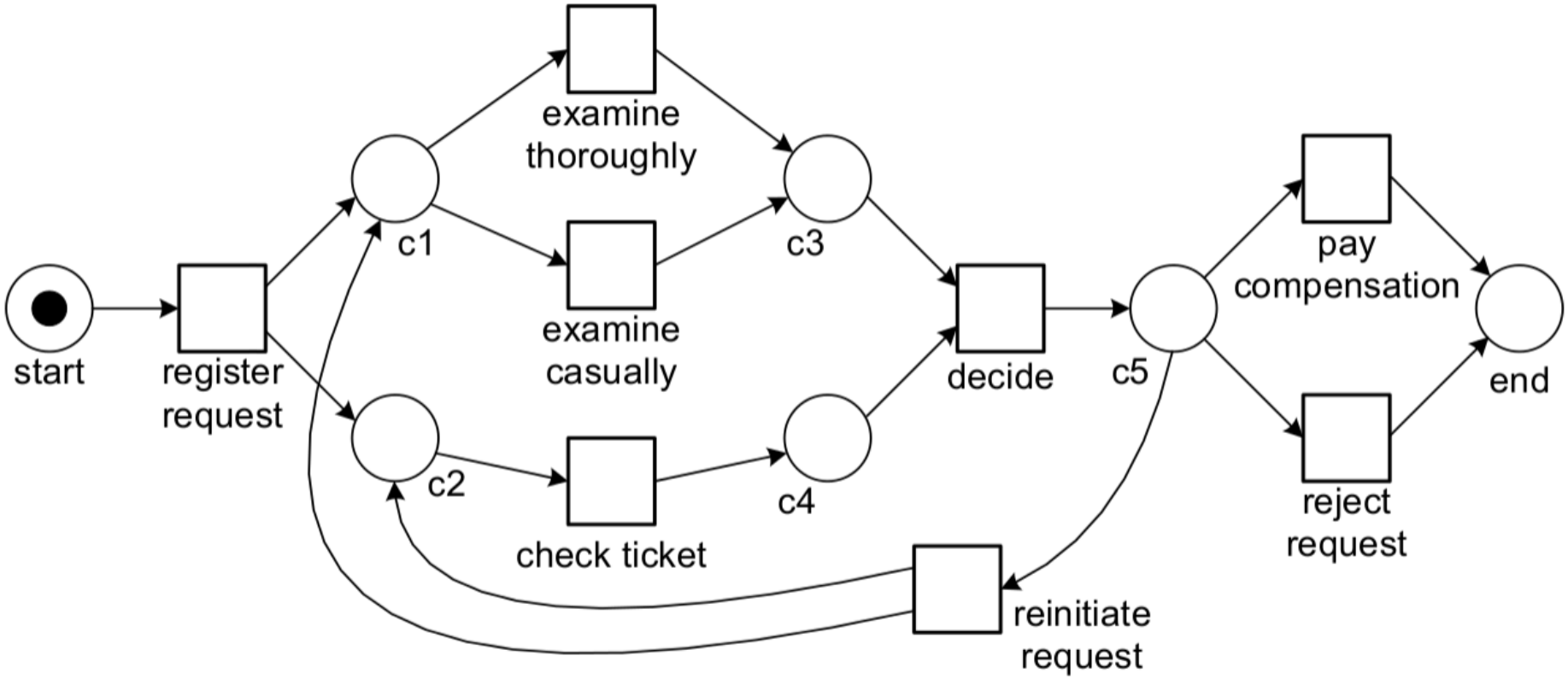
Recap

- **Business Intelligence**: anything that aims at providing actionable information that can be used to support business decision making
  - Business Intelligence
  - Visual Analytics
  - Descriptive Analytics
  - Predictive Analytics
- **Process Modelling** (April and May)

Process Mining (recap)



Petri Nets (recap)



Simplified event log (recap)

Case ID	Trace
1	$\langle a, b, d, e, h \rangle$
2	$\langle a, d, c, e, g \rangle$
3	$\langle a, c, d, e, f, b, d, e, g \rangle$
4	$\langle a, d, b, e, h \rangle$
5	$\langle a, c, d, e, f, d, c, e, f, c, d, e, h \rangle$
6	$\langle a, c, d, e, g \rangle$

Table : Simplified event log of a support desk handling customer compensations (a = register request, b = examine thoroughly, c = examine casually, d = check ticket, e = decide, f = reinitiate request, g = pay compensation, h = reject request)

Simplified event log (recap)

Case ID	Trace
1	$\langle a, b, d, e, h \rangle$
2	$\langle a, d, c, e, g \rangle$
3	$\langle a, c, d, e, f, b, d, e, g \rangle$
4	$\langle a, d, b, e, h \rangle$
5	$\langle a, c, d, e, f, d, c, e, f, c, d, e, h \rangle$
6	$\langle a, c, d, e, g \rangle$

Table : Simplified event log of a support desk handling customer compensations (a = register request, b = examine thoroughly, c = examine casually, d = check ticket, e = decide, f = reinitiate request, g = pay compensation, h = reject request)

In short: $\{\langle a, b, d, e, h \rangle, \langle a, d, c, e, g \rangle, \langle a, c, d, e, f, b, d, e, g \rangle, \langle a, d, b, e, h \rangle, \langle a, c, d, e, f, d, c, e, f, c, d, e, h \rangle, \langle a, c, d, e, g \rangle\}$

Labeled Petri Nets (recap)

- Petri net $N = (P, T, F, A, \ell)$
- P is a finite set of **places**
- T is a finite set of **transitions**
- $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of directed **arcs** called the **flow relation**
- A is a set of activity **labels**
- $\ell : T \to A$ is a **labeling function**

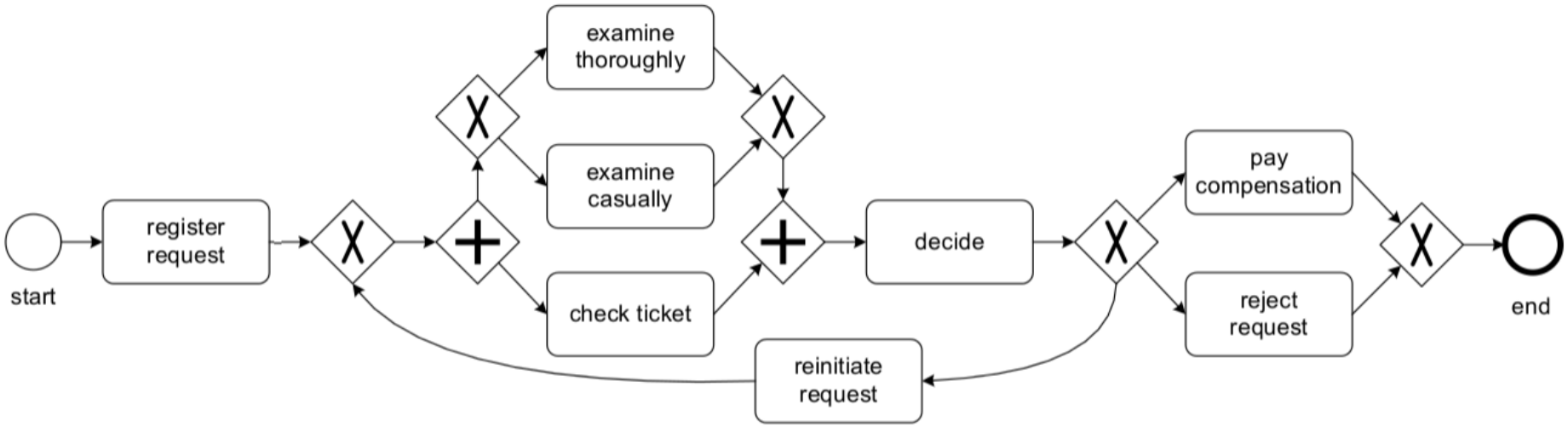
Universiteit Leiden

BPMN

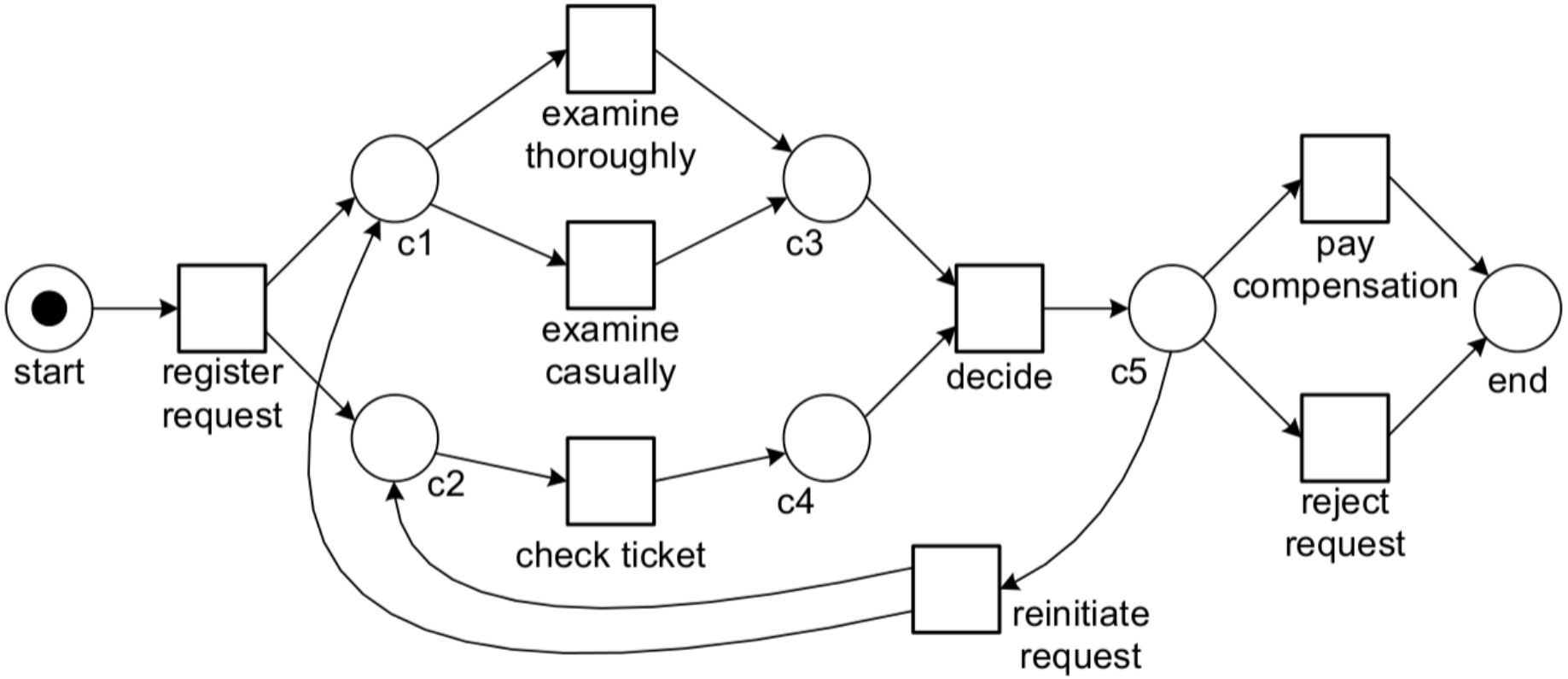
BPMN

- **Business Process Modelling Notation (BPMN)**
- S. White, *Business Process Modelling Notation v1.0*, Business Process Management Initiative, 2004.
- Maintained by BPMI and later the OMG consortium
- Now at version 2.0.2 (2013)
- 532 pages of specification:
  `http://www.omg.org/cgi-bin/doc?formal/2013-12-09.pdf`

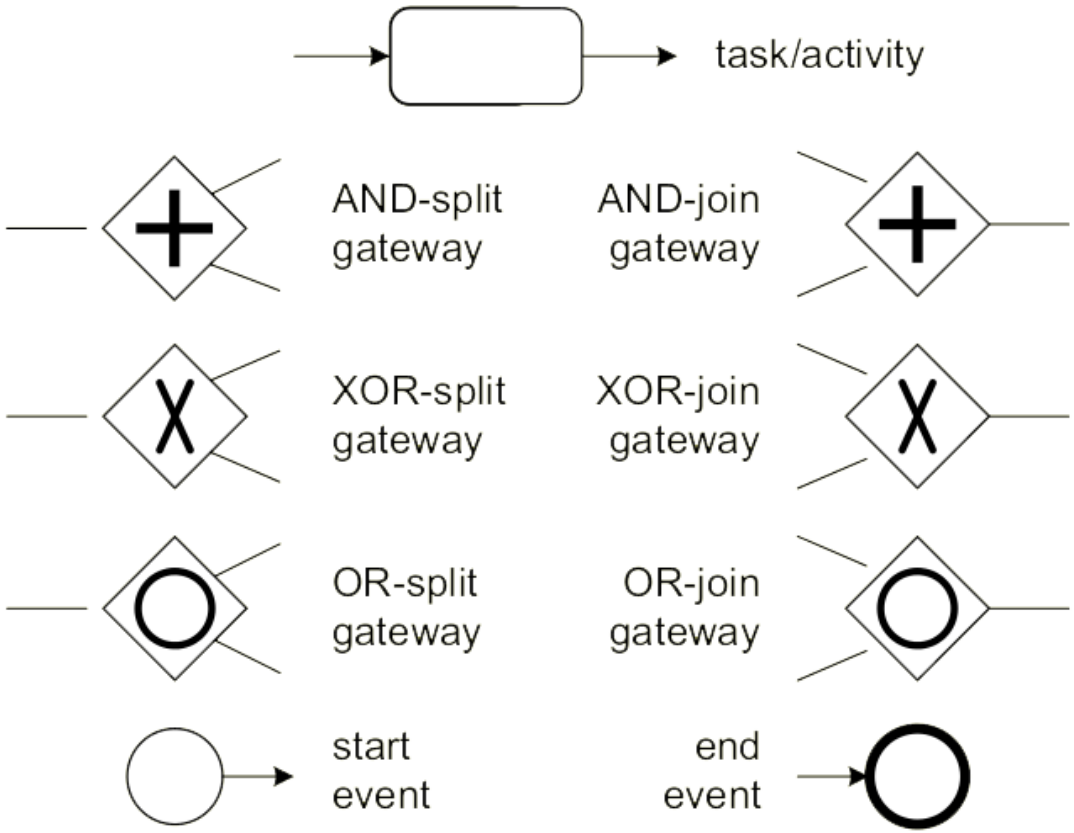
BPMN Example



Corresponding Petri net



Simple BPMN gates



BPMN Example

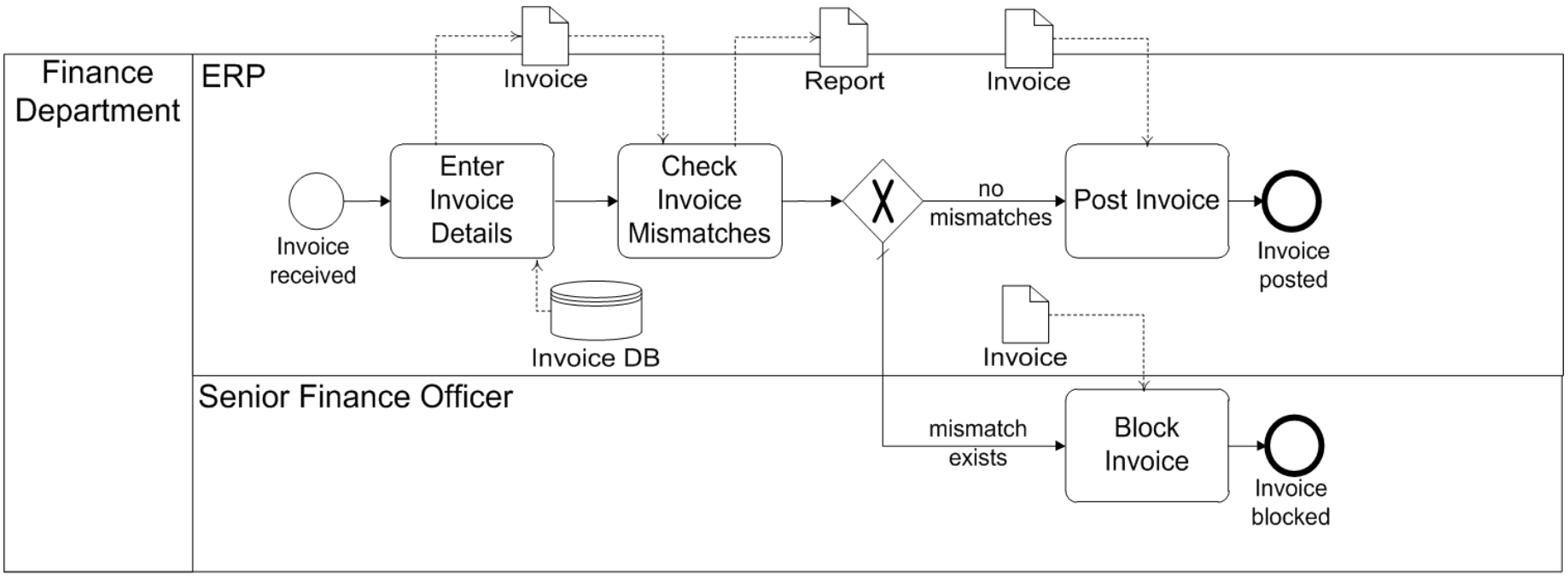
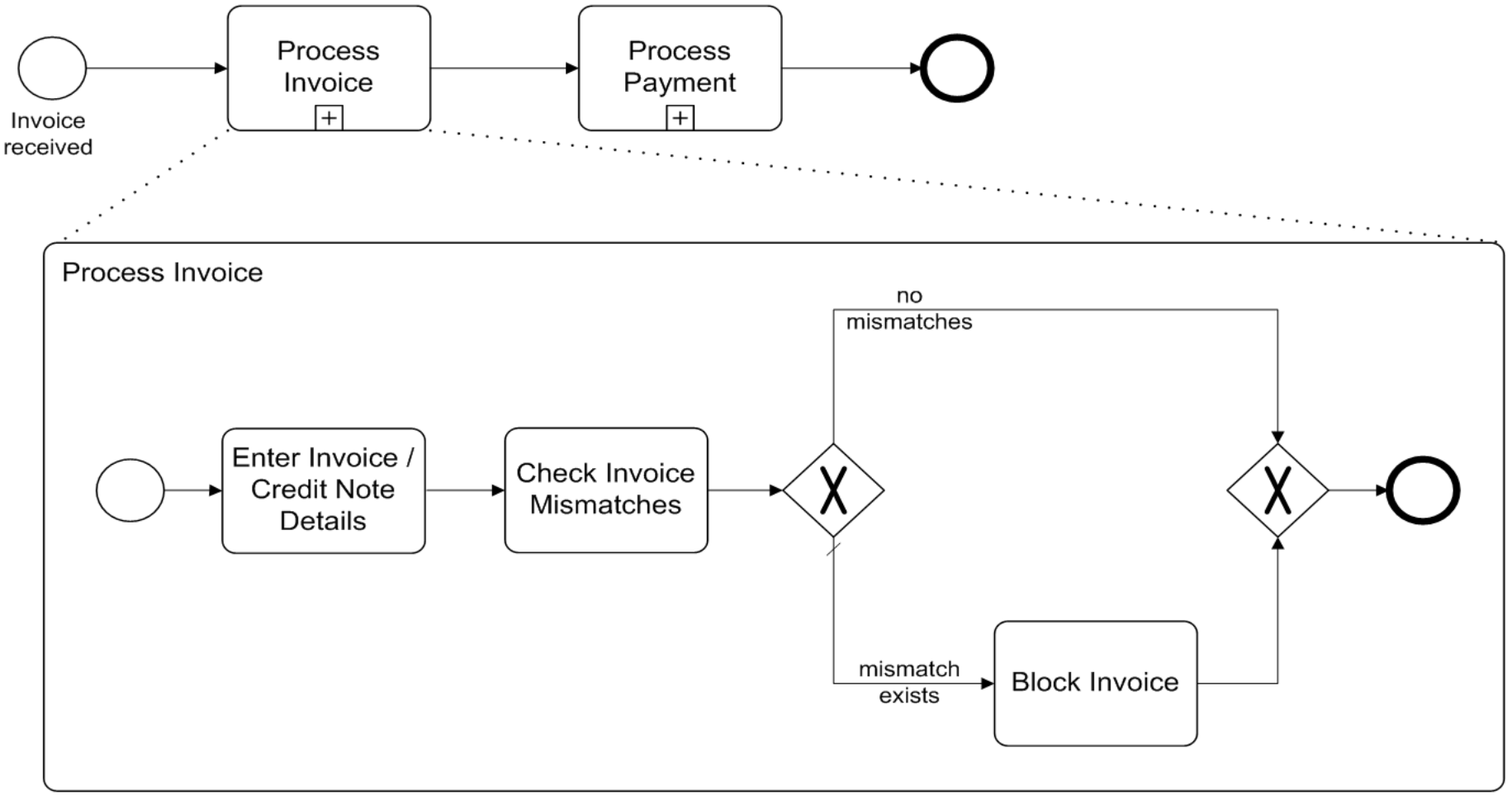


Figure : Regular vs. dotted arcs

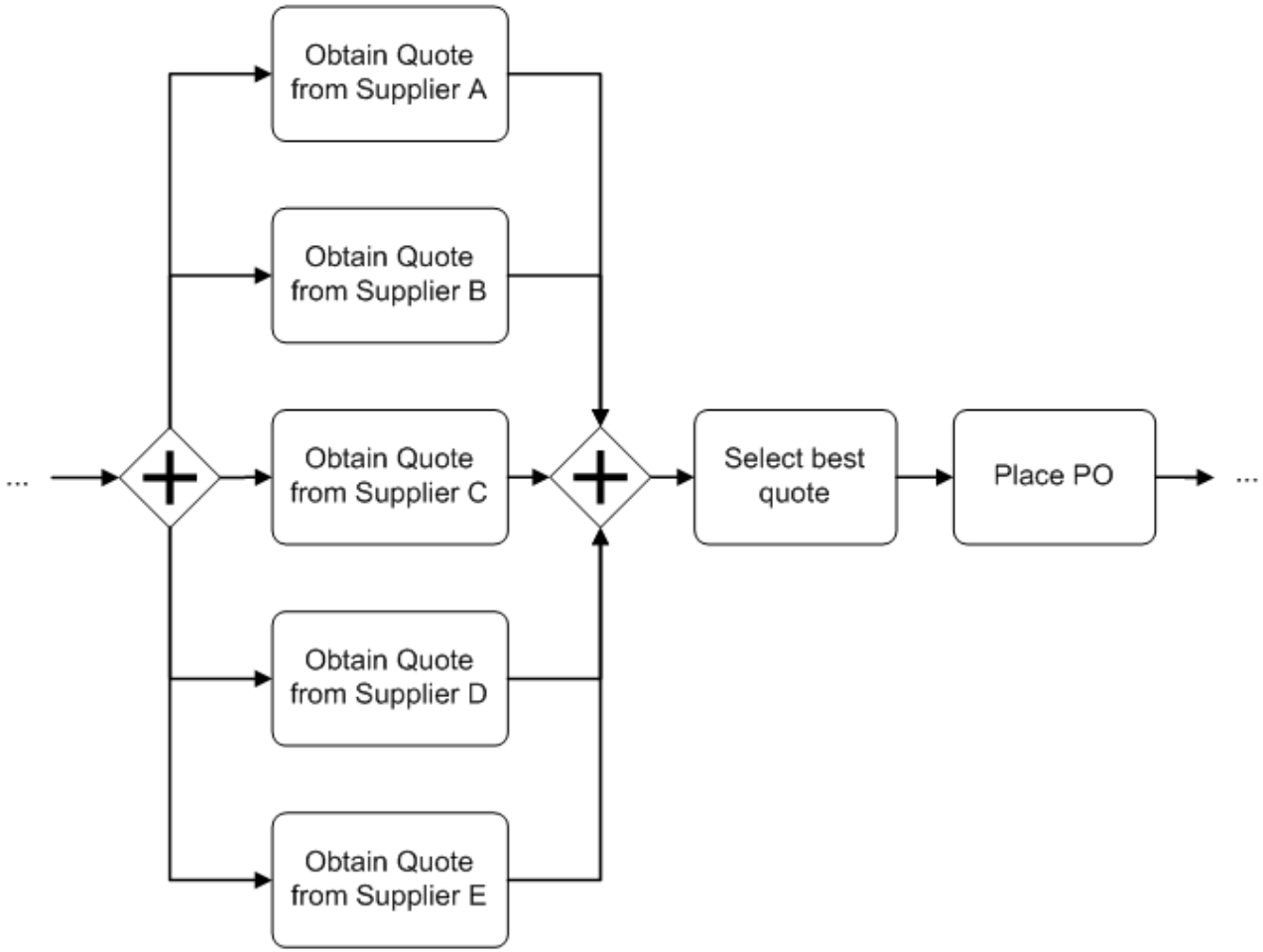
Feature: subprocesses

- Complex processes can be hard to understand
- In large schemas, errors are easily overlooked
- Solution: **decomposition** (encapsulation)
- Sub-processes in an activity
- Sub-process can be re-used later on

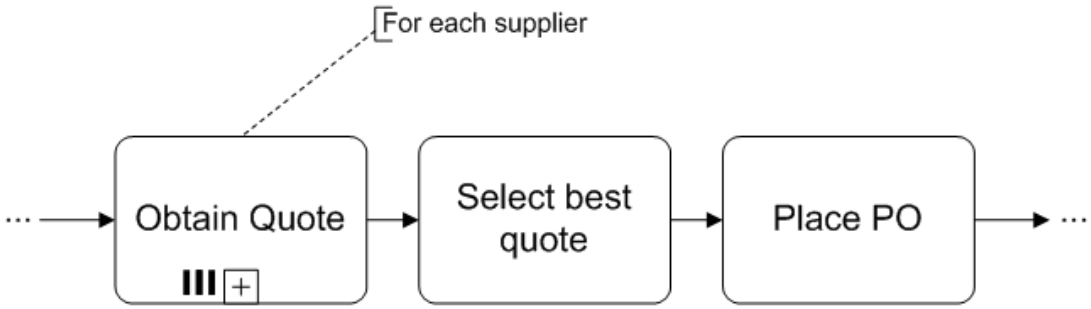
Feature: subprocesses



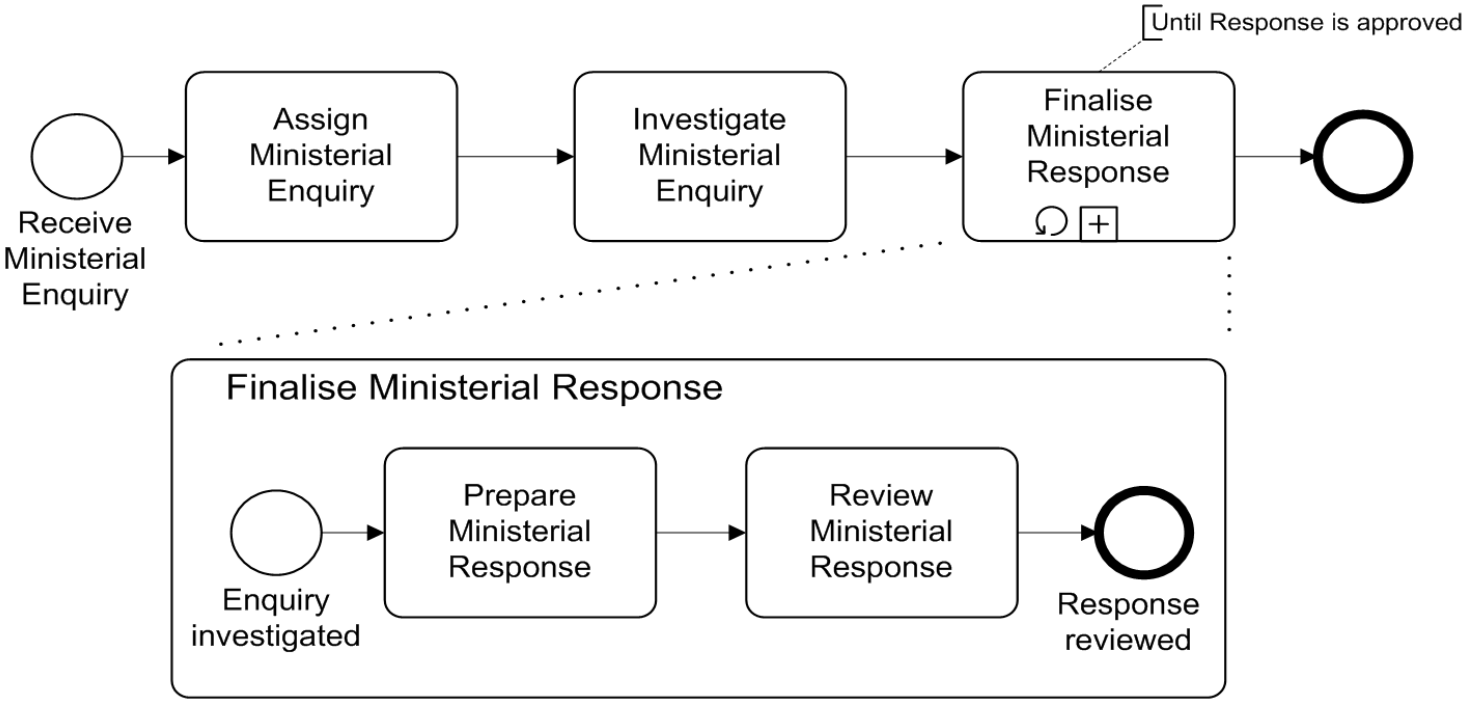
Feature: multiple instances



Feature: multiple instances



Feature: loops



Featured BPMN gates

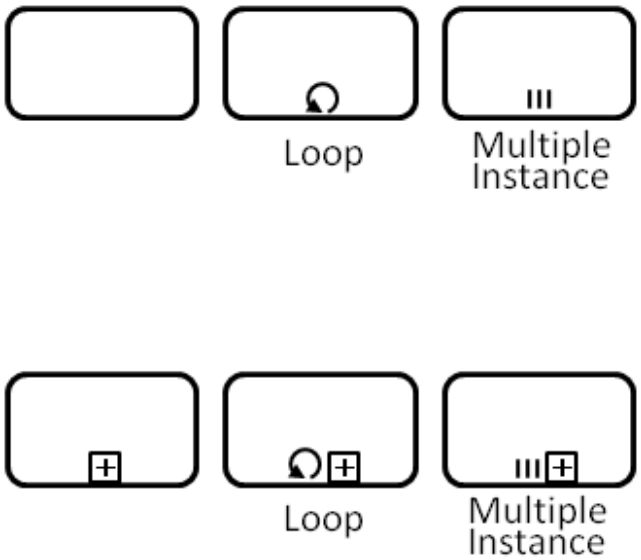


Figure : Loops and multiple instances, plain and as a subprocess

Example exam questions (so far)

- Given an informal process description, draw a Petri net or BPMN diagram
- Given a Petri net, draw the reachability graph
- Given a Petri net, draw the BPMN diagram (or vice versa)
- Play an event log on a Petri net or an BPMN diagram
- Given a Petri net, determine if it adheres to certain properties (e.g., deadlock-free, safe, etc.)

Universiteit Leiden

Event logs & Visualization

Model-based analysis

- Limitations of model-based analysis:
  - Verification and performance analysis heavily rely on the availability of high quality models
  - When the models and reality have little in common, model-based analysis does not make much sense
  - There is often a lack of alignment between hand-made models and reality
- **Process mining** aims to address these problems by establishing a direct connection between the models and actual low-level event data about the process.
- **Process discovery** techniques allow for viewing the same reality from different angles and at different levels of abstraction.

Event logs

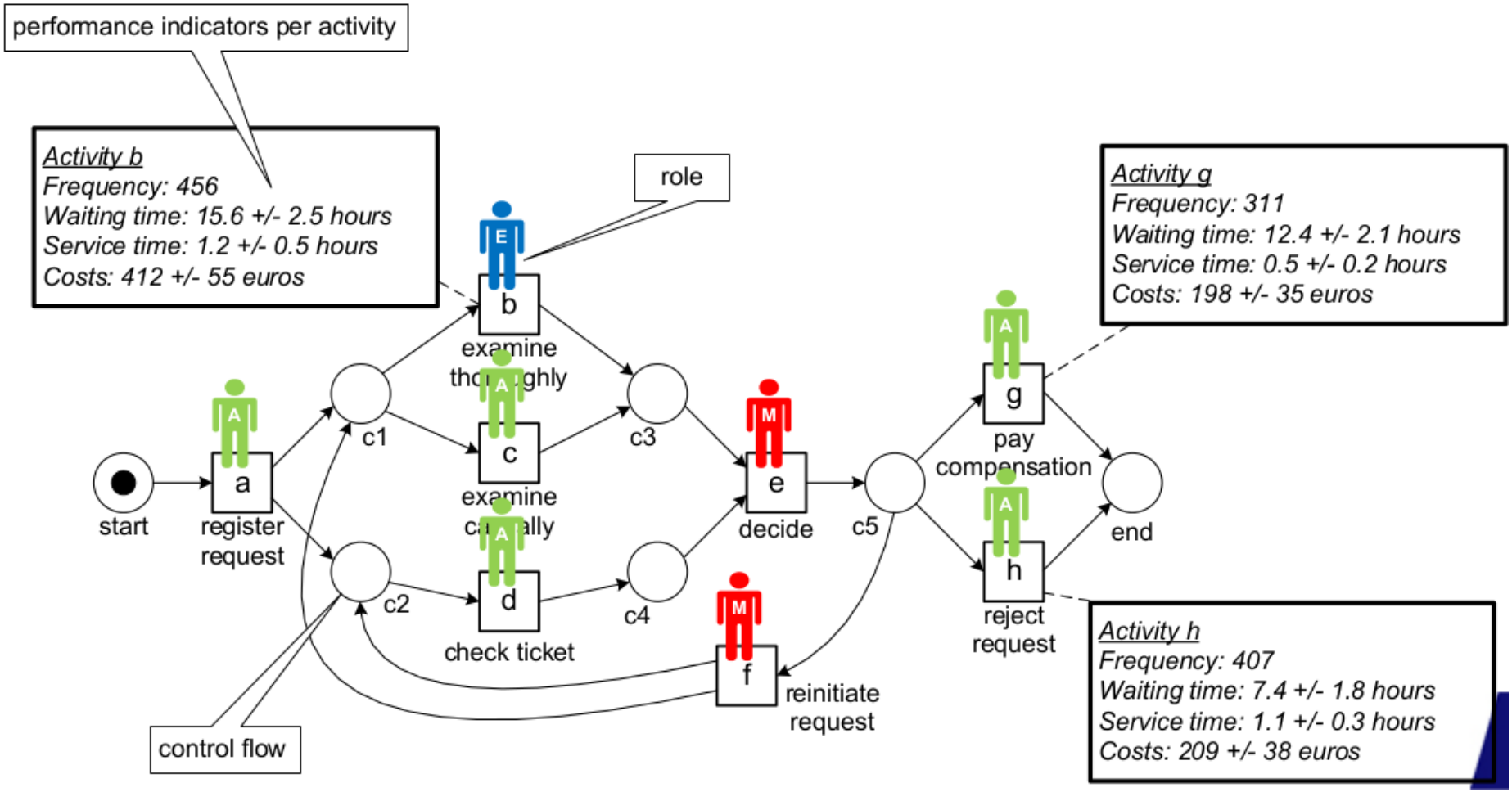
Case ID	Event ID	dd-mm-yyyy:hh.mm	Activity	Resource	Costs
1	35654423	30-12-2010:11.02	register request	Pete	50
1	35654424	31-12-2010:10.06	examine thoroughly	Sue	400
1	35654425	05-01-2011:15.12	check ticket	Mike	100
1	35654426	06-01-2011:11.18	decide	Sara	200
1	35654427	07-01-2011:14.24	reject request	Pete	200
2	35654483	30-12-2010:11.32	register request	Mike	50
2	35654485	30-12-2010:12.12	check ticket	Mike	100
2	35654487	30-12-2010:14.16	examine casually	Sean	400
2	35654488	05-01-2011:11.22	decide	Sara	200
2	35654489	08-01-2011:12.05	pay compensation	Ellen	200
3	35654521	30-12-2010:14.32	register request	Pete	50
3	35654522	30-12-2010:15.06	examine casually	Mike	400
3	35654524	30-12-2010:16.34	check ticket	Ellen	100
3	35654525	06-01-2011:09.18	decide	Sara	200
3	35654526	06-01-2011:12.18	reinitiate request	Sara	200
3	35654527	06-01-2011:13.06	examine thoroughly	Sean	400
3	35654530	08-01-2011:11.43	check ticket	Pete	100
3	35654531	09-01-2011:09.55	decide	Sara	200
3	35654533	15-01-2011:10.45	pay compensation	Ellen	200
4	35654641	06-01-2011:15.02	register request	Pete	50
4	35654643	07-01-2011:12.06	check ticket	Mike	100
4	35654644	08-01-2011:14.43	examine thoroughly	Sean	400
4	35654645	09-01-2011:12.02	decide	Sara	200
4	35654647	12-01-2011:15.44	reject request	Ellen	200
...					

Table : Event logs of a helpdesk handling customer compensations

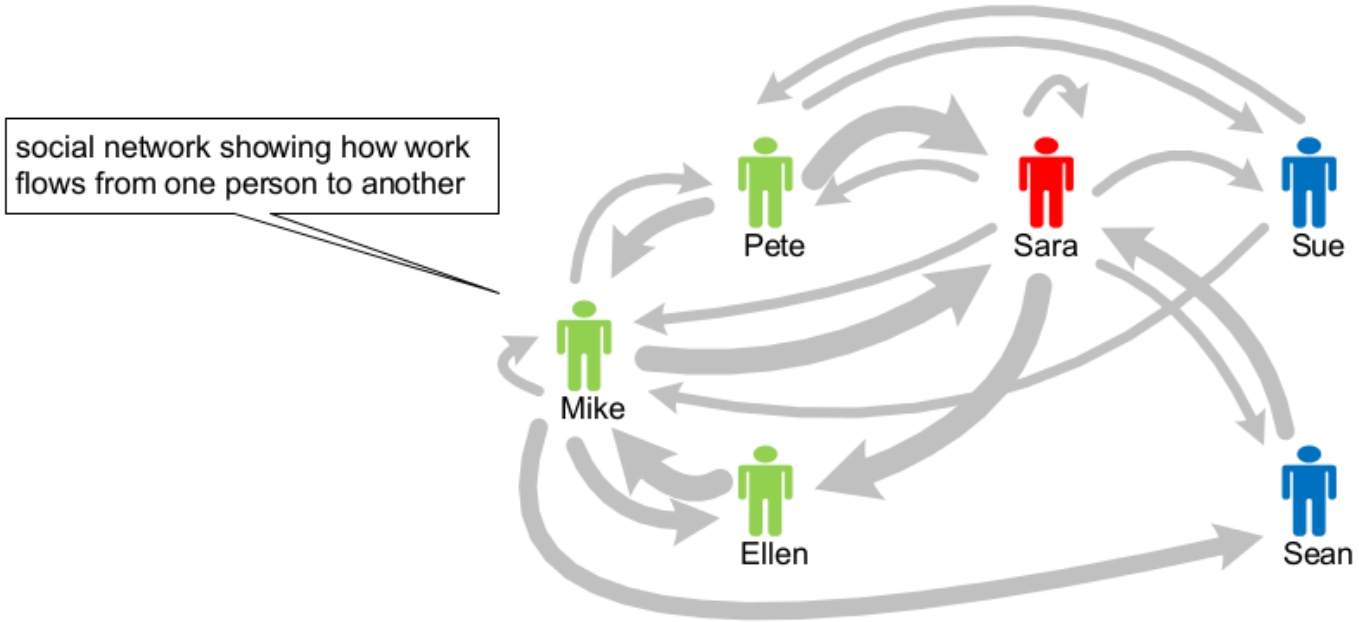
Event logs

- A process consists of **cases**
- A case consists of **events** such that each event relates to precisely one case
- Events within a case are **ordered**.
- Events can have **attributes**, e.g.: activity, time, costs and resource

Event log attributes



Resource-based Social Network



Extracting event logs

- From data to event log (ETL-like process)
- Not just a syntactical issue
- Different views are possible
- Important:
  - Selecting the right instance notion
  - Ordering of events
  - Selection of events
- Tool support: e.g., XESame

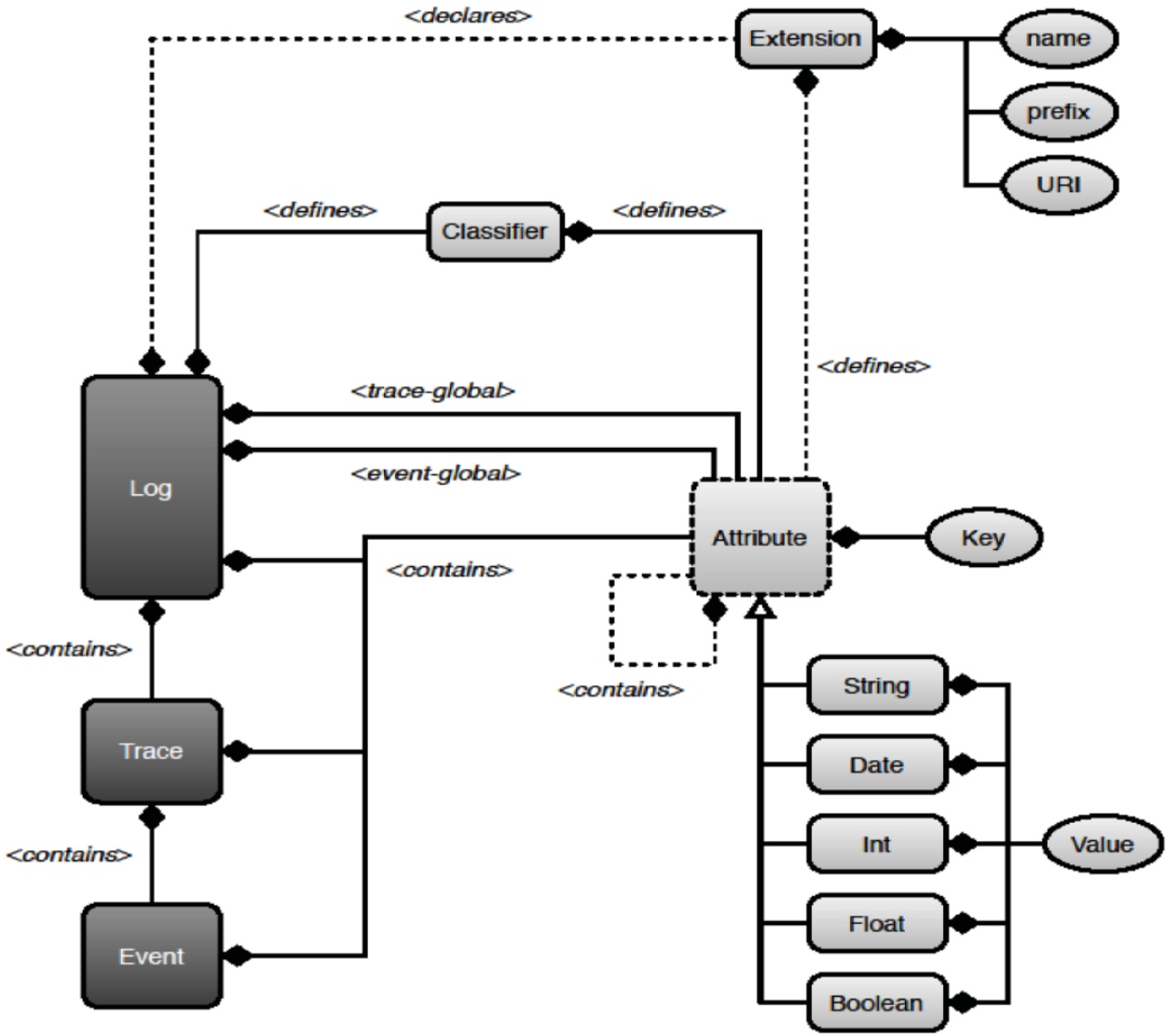
Challenges when extracting event logs

- **Correlation**: events in an event log are grouped per case. Challenging, as it requires event correlation, i.e., events need to be related to each other
- **Timestamps**: events need to be ordered per case. Typical problems: only dates, different clocks, delayed logging
- **Snapshots**: cases may have a lifetime extending beyond the recorded period, e.g., a case was started before the beginning of the event log
- **Scoping**: how to decide which tables to incorporate?

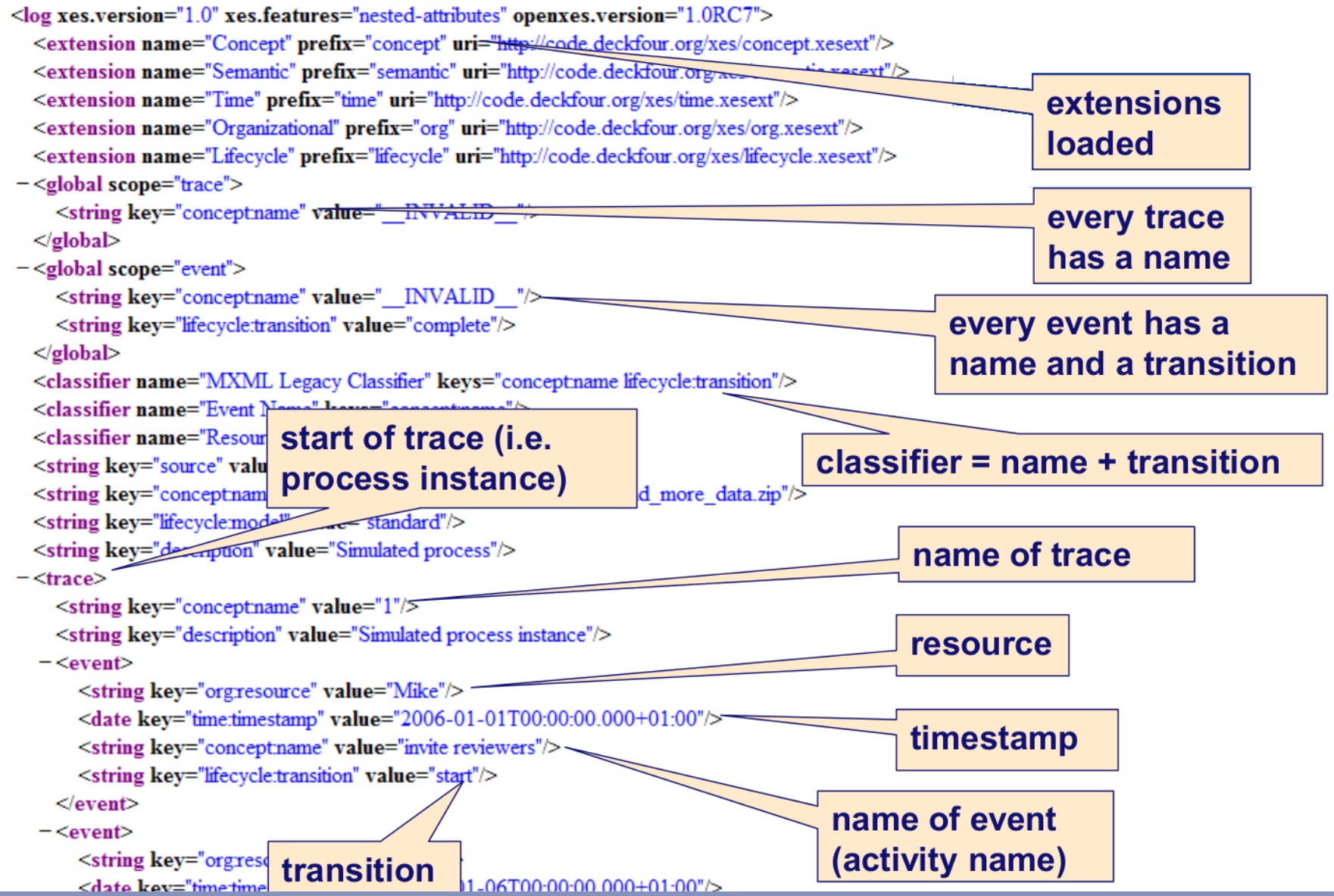
XES

- XES (eXtensible Event Stream)
- `http://www.xes-standard.org`
- Adopted by the IEEE Task Force on Process Mining
- Predecessors: MXML and SA-MXML.
- The format is supported by tools such as ProM

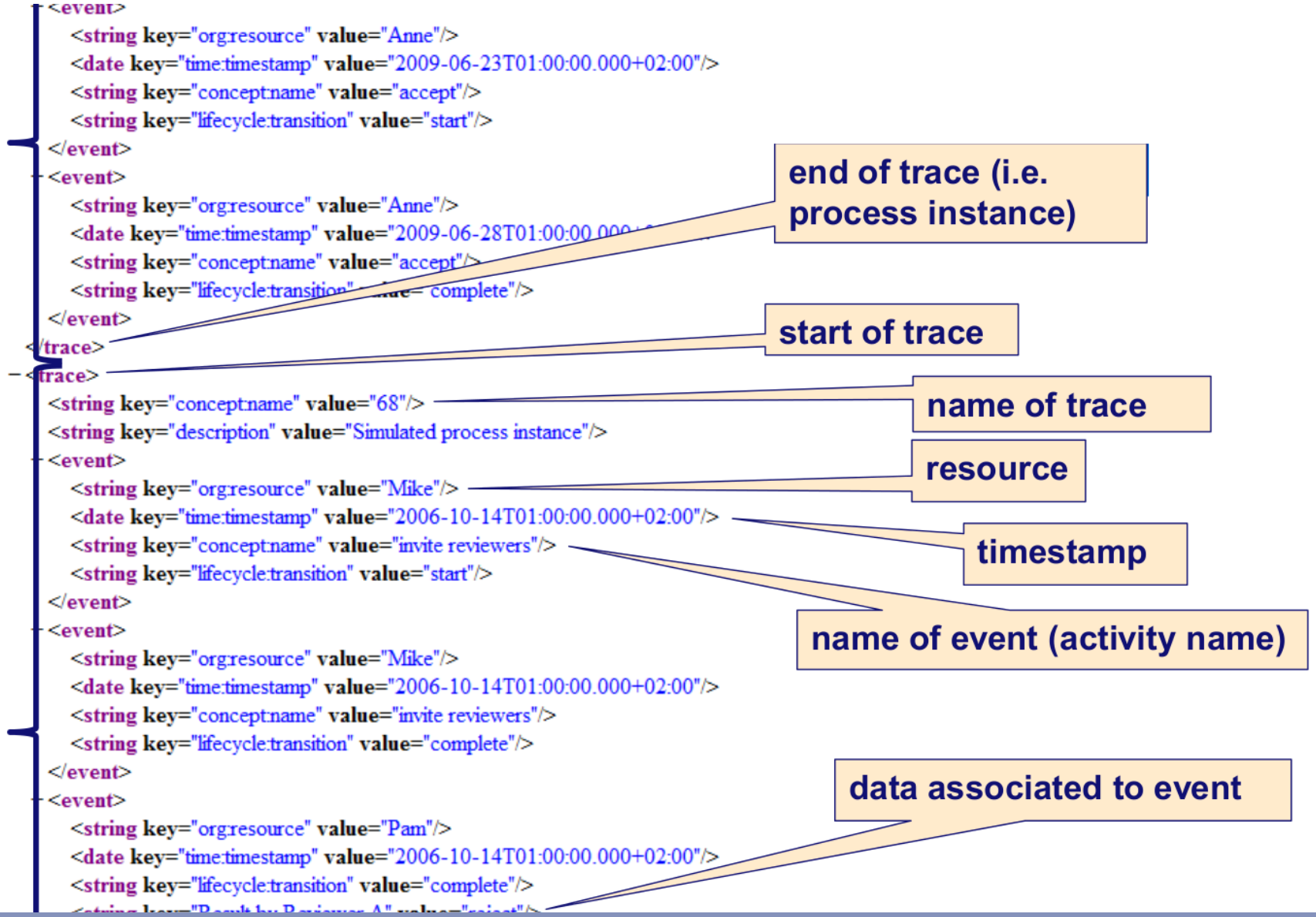
XES specification



XES header


<log xes.version="1.0" xes.features="nested-attributes" openxes.version="1.0RC7">
<extension name="Concept" prefix="concept" uri="http://code.deckfour.org/xes/concept.xesext"/>
<extension name="Semantic" prefix="semantic" uri="http://code.deckfour.org/
<extension name="Time" prefix="time" uri="http://code.deckfour.org/xes/time.xesext"/>
<extension name="Organizational" prefix="org" uri="http://code.deckfour.org/xes/org.xesext"/>
<extension name="Lifecycle" prefix="lifecycle" uri="http://code.deckfour.org/xes/lifecycle.xesext"/>
<global scope="trace">
<string key="concept:name" value="__INVALID__"/>
</global>
<global scope="event">
<string key="concept:name" value="__INVALID__"/>
<string key="lifecycle:transition" value="complete"/>
</global>
<classifier name="MXML Legacy Classifier" keys="concept:name lifecycle:transition"/>
<classifier name="Event
<classifier name="Resour
<string key="source" valu
<string key="concept:nam
d_more_data.zip"/>
<string key="lifecycle:mode
standard"/>
<string key="de
ription" value="Simulated process"/>
<trace>
<string key="concept:name" value="1"/>
<string key="description" value="Simulated process instance"/>
<event>
<string key="org:resource" value="Mike"/>
<date key="time:timestamp" value="2006-01-01T00:00:00.000+01:00"/>
<string key="concept:name" value="invite reviewers"/>
<string key="lifecycle:transition" value="start"/>
</event>
<event>
<string key="org:reso
<date key="time:time
01-06T00:00:00.000+01:00"/>
extensions loaded
every trace has a name
every event has a name and a transition
classifier = name + transition
start of trace (i.e. process instance)
name of trace
resource
timestamp
name of event (activity name)
transition

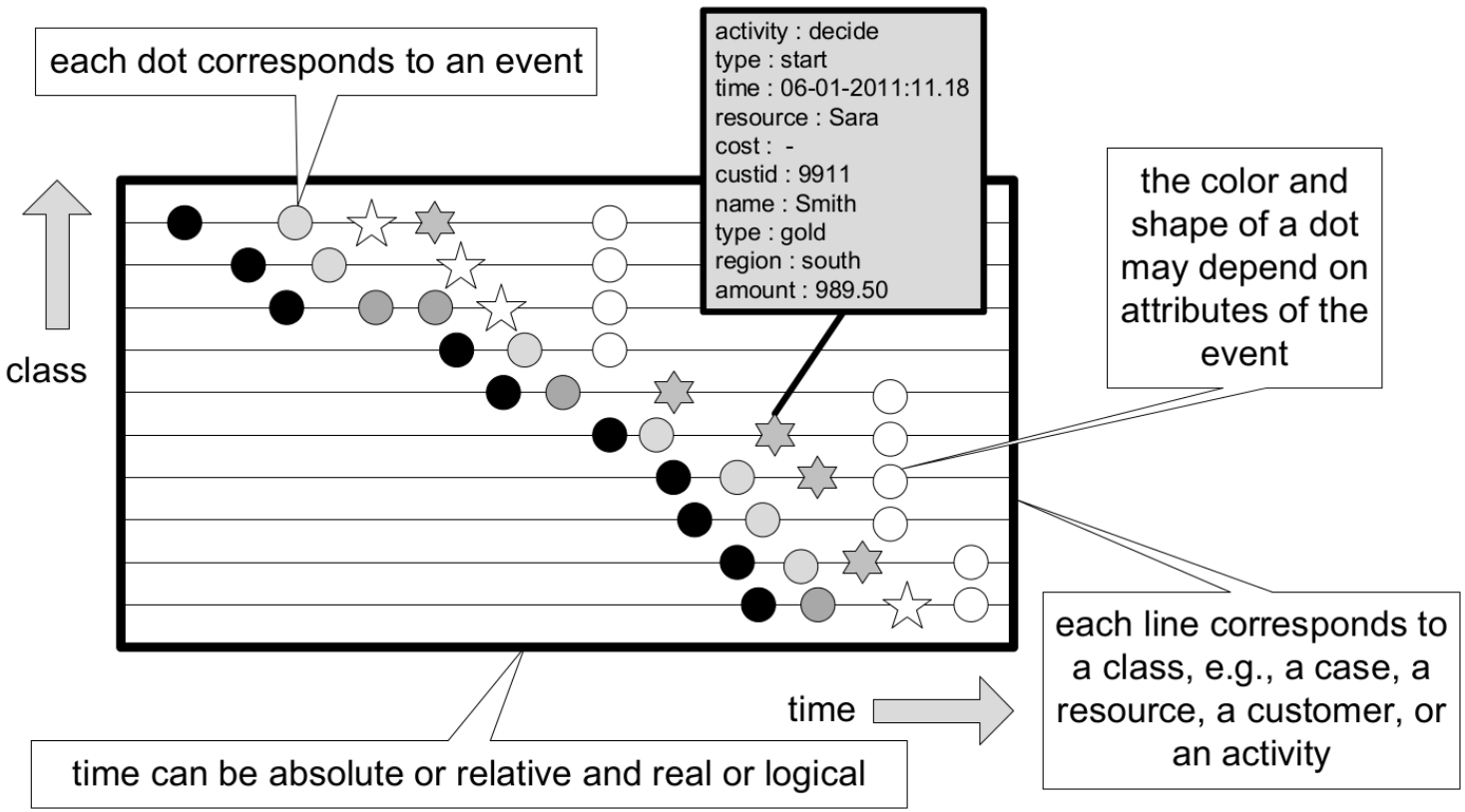
XES events


<event>
<string key="org:resource" value="Anne"/>
<date key="time:timestamp" value="2009-06-23T01:00:00.000+02:00"/>
<string key="concept:name" value="accept"/>
<string key="lifecycle:transition" value="start"/>
</event>
<event>
<string key="org:resource" value="Anne"/>
<date key="time:timestamp" value="2009-06-28T01:00:00.000
<string key="concept:name" value="accept"/>
<string key="lifecycle:transition"
complete"/>
</event>
</trace>
<trace>
<string key="concept:name" value="68"/>
<string key="description" value="Simulated process instance"/>
<event>
<string key="org:resource" value="Mike"/>
<date key="time:timestamp" value="2006-10-14T01:00:00.000+02:00"/>
<string key="concept:name" value="invite reviewers"/>
<string key="lifecycle:transition" value="start"/>
</event>
<event>
<string key="org:resource" value="Mike"/>
<date key="time:timestamp" value="2006-10-14T01:00:00.000+02:00"/>
<string key="concept:name" value="invite reviewers"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="Pam"/>
<date key="time:timestamp" value="2006-10-14T01:00:00.000+02:00"/>
<string key="lifecycle:transition" value="complete"/>
end of trace (i.e. process instance)
start of trace
name of trace
resource
timestamp
name of event (activity name)
data associated to event

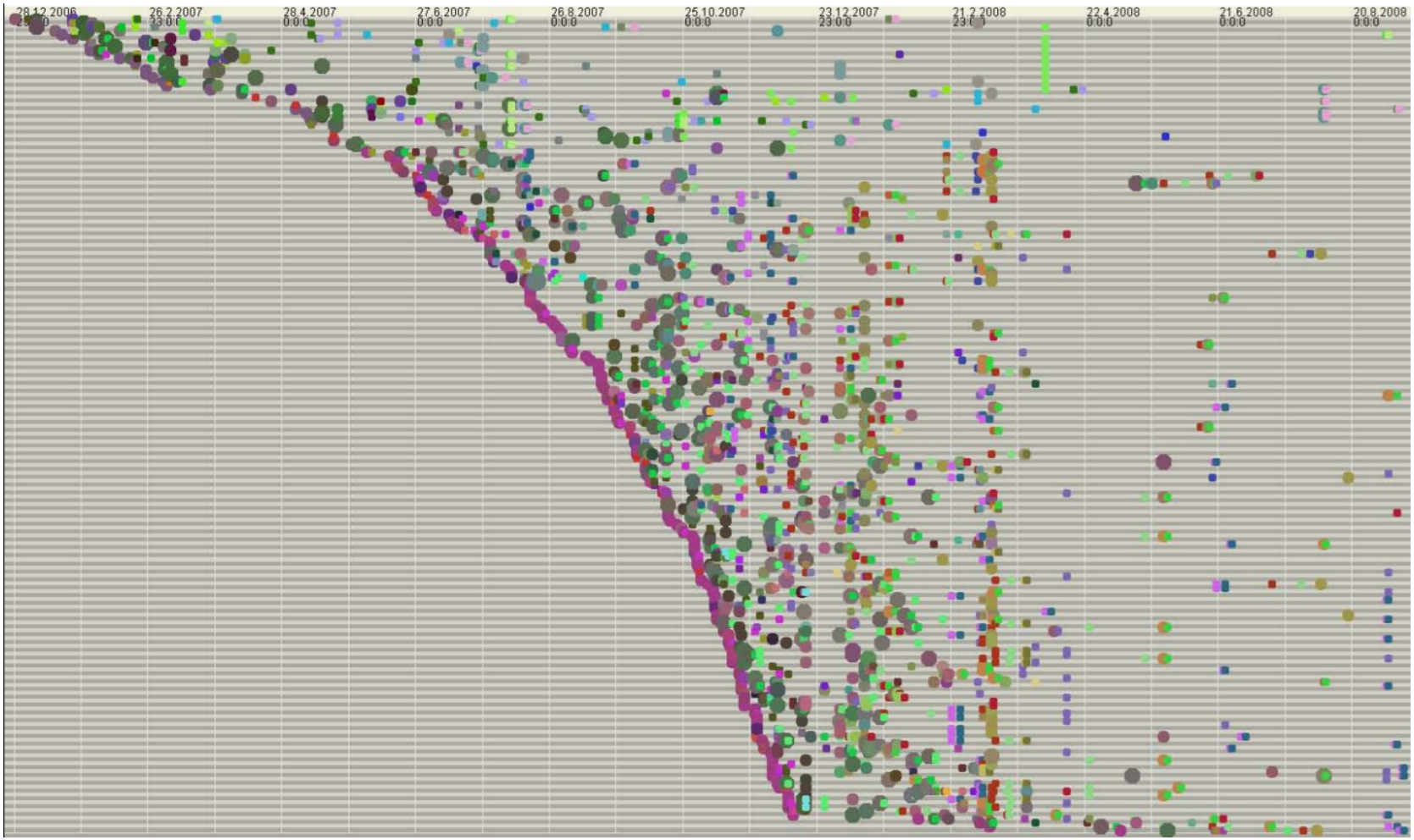
Dotted Chart Analysis

- Visualization of cases
- Colors indicate the same event within a case
- Horizontal axis indicates time
- Cases along the vertical axis
- Absolute or relative time
- Attributes can be added as labels

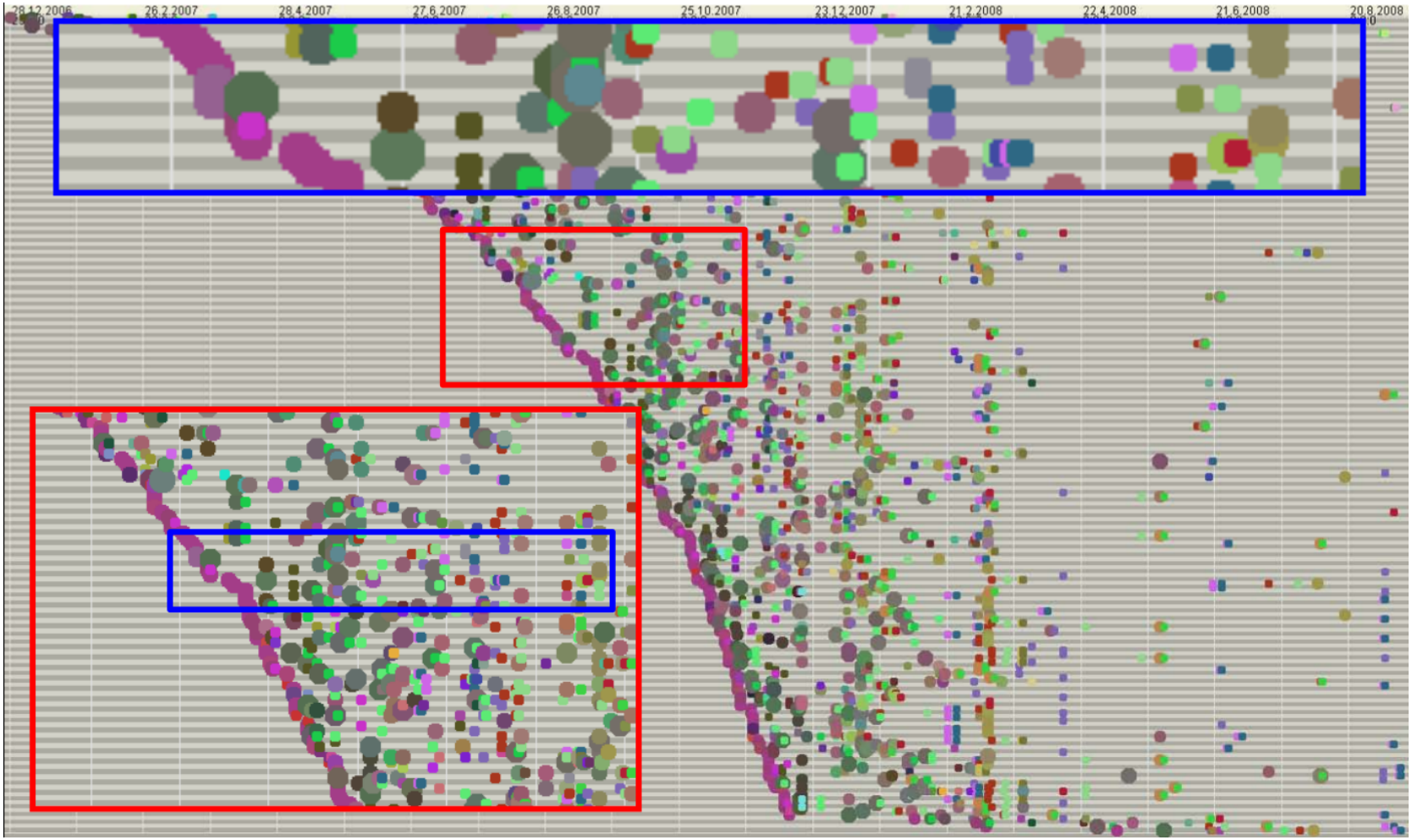
Dotted charts



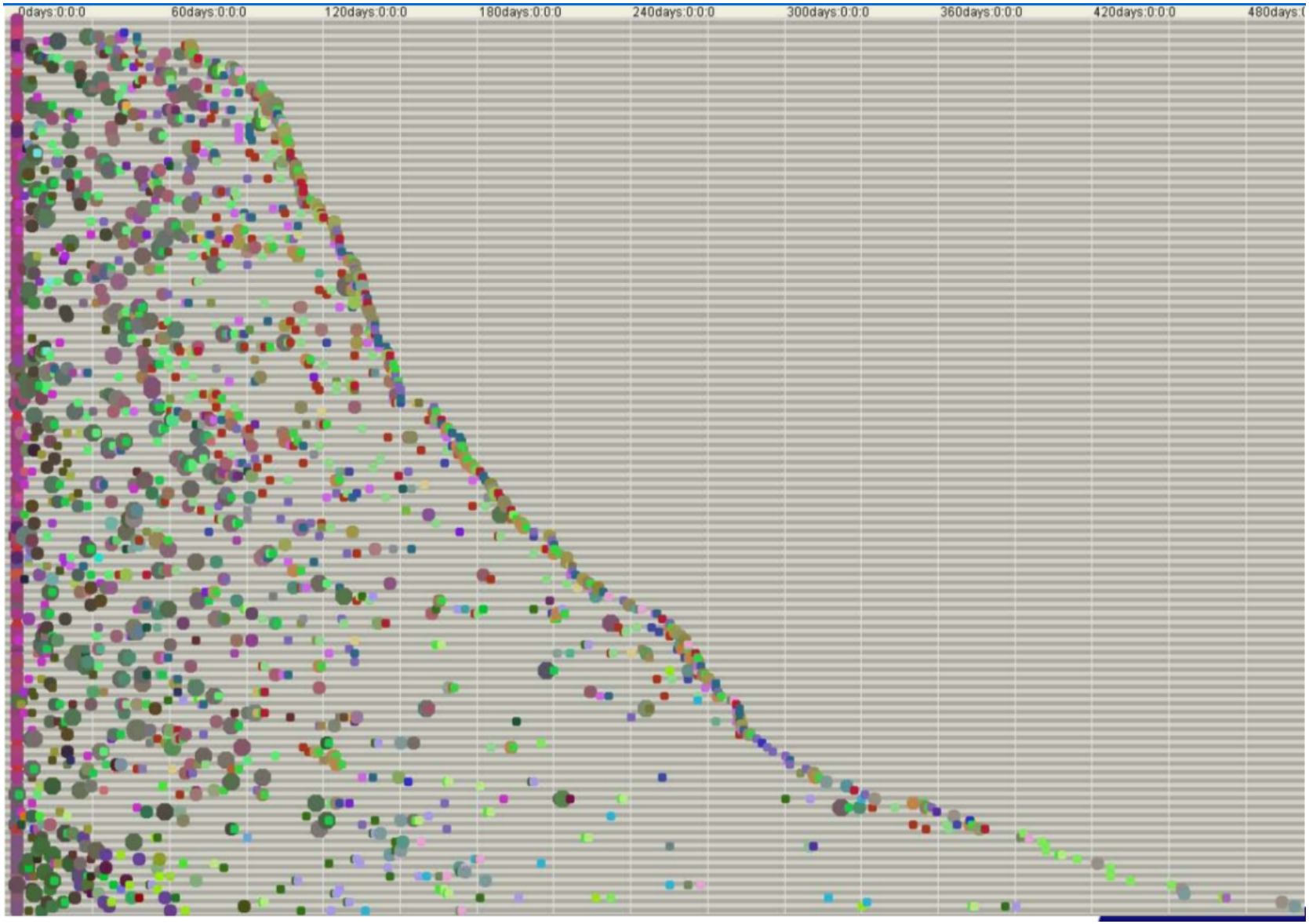
Dotted charts (absolute)



Dotted charts (absolute)



Dotted charts (relative)



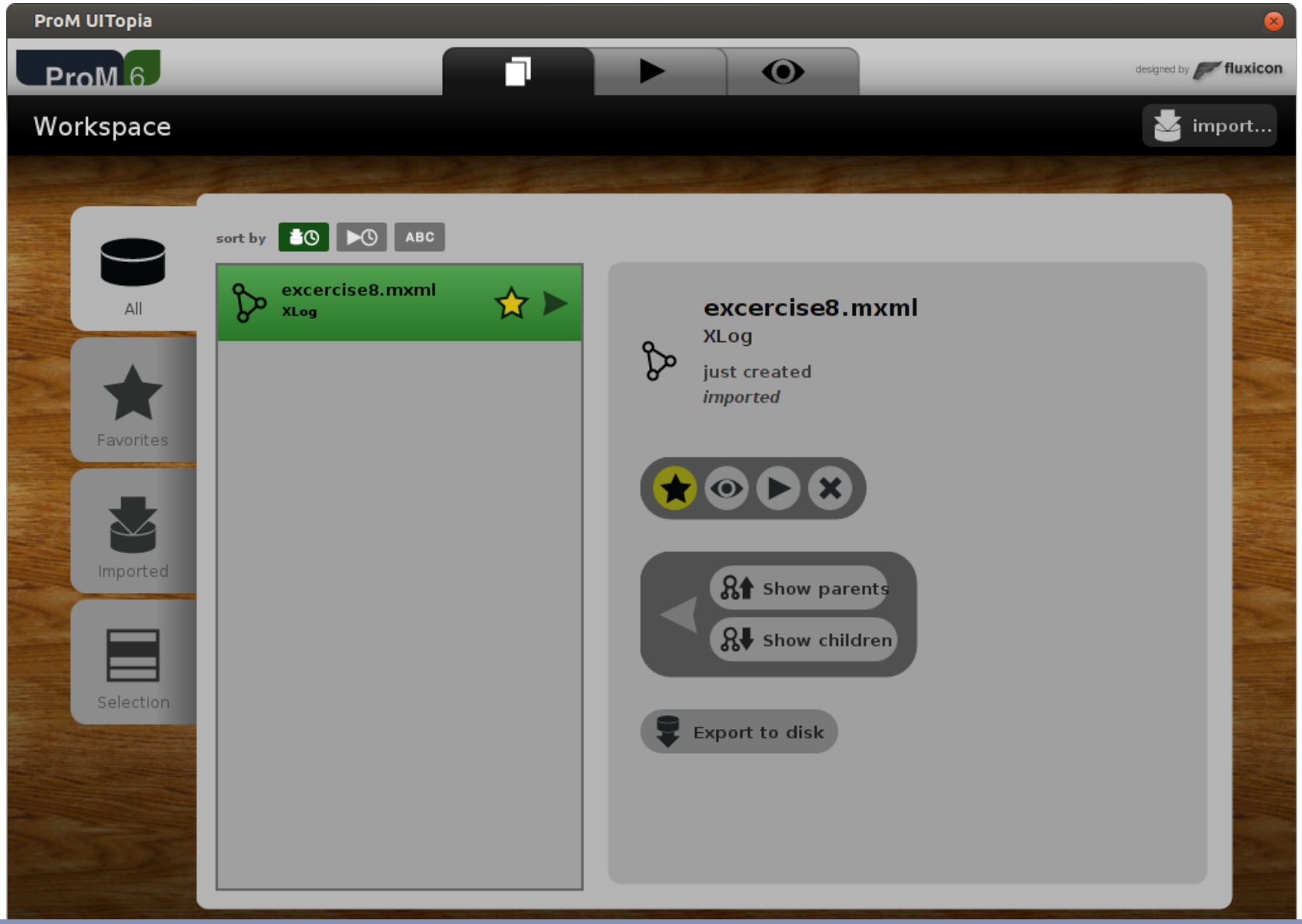
Universiteit Leiden

Tool support

Tool support

product name	type	organization
ARIS Process Performance Manager	C	Software AG (www.softwareag.com)
Enterprise Visualization Suite	C	Businesscape (www.businesscape.no)
Disco	C	Fluxicon (www.fluxicon.com)
Genet/Petrify	A	Universitat Politècnica de Catalunya (www.lsi.upc.edu)
Interstage BPME	C	Fujitsu (www.fujitsu.com)
OKT Process Mining suite	O	Exeura s.r.l. (www.exeura.com)
Process Discovery Focus	C	Iontas (Verint Systems) (www.iontas.com)
ProcessAnalyzer	C	QPR (www.qpr.com)
ProM	O	process mining group (managed by the AIS group at TU/e) (www.processmining.org)
Rbminer/Dbminer	A	Universitat Politècnica de Catalunya (www.lsi.upc.edu)
Reflect|one	C	Pallas Athena (www.pallas-athena.com)
Reflect	C	Futura Process Intelligence (www.futuratech.nl)
ServiceMosaic	A	University of New South Wales (soc.cse.unsw.edu.au)

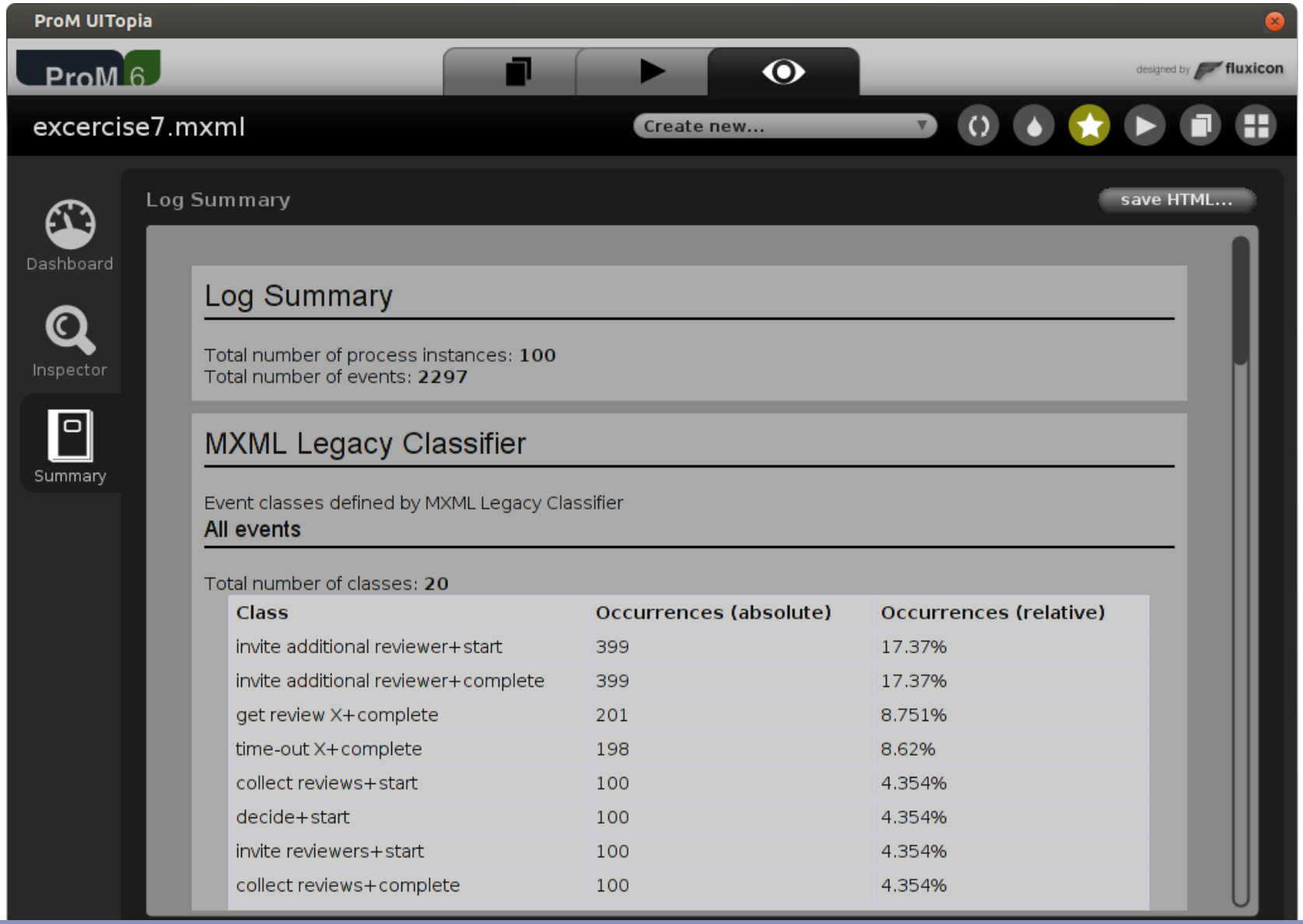
ProM


ProM UITopia
ProM 6
designed by fluxicon
Workspace
import...
All
Favorites
Imported
Selection
sort by
ABC
excercise8.mxml
XLog
excercise8.mxml
XLog
just created
imported
Show parents
Show children
Export to disk

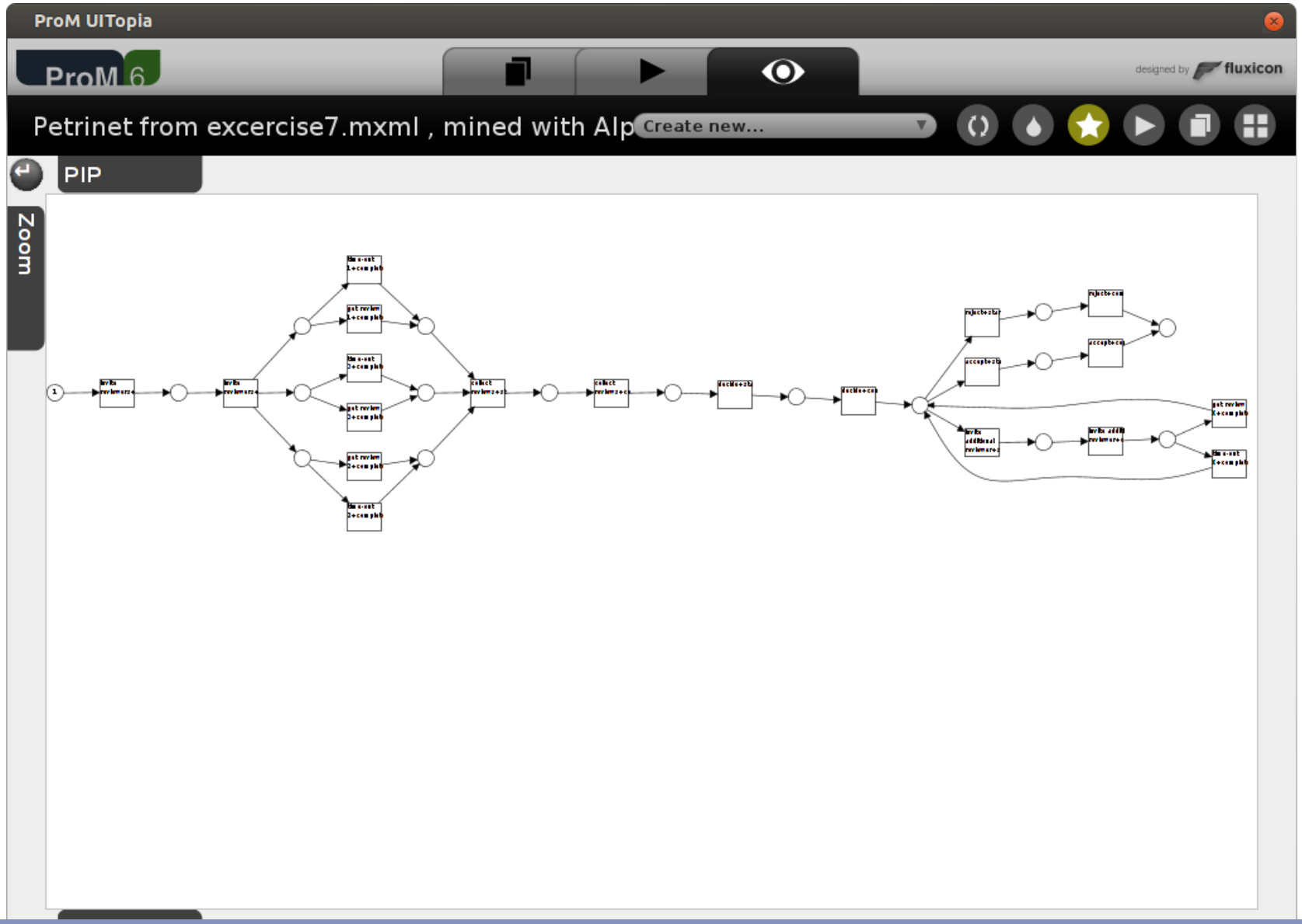
ProM visualization

ProM exploration

ProM summarization



ProM mining


ProM UITopia
ProM 6
designed by fluxicon
Petrinet from excercise7.mxml , mined with Alp
Create new...
PIP
Zoom

ProM plugins

plug-in	description
Alpha miner	discovers a Petri net using the α-algorithm, see Section 5.2
Heuristic miner	discovers a C-net using heuristic mining, see Section 6.2
Genetic miner	discovers a C-net using genetic mining, see Section 6.3
Fuzzy miner	discovers a fuzzy model using fuzzy mining, see Section 13.1.3 and [72]
Transition system miner	discovers a transition system based on a state representation function and a log, see Section 6.4.1
Transition system to Petri net	uses state-based regions to create a Petri net based on a transition system, see Section 6.4.2
Declare miner	discovers a Declare model, see Section 7.3
ILP miner	discovers a Petri net using language-based regions, see Section 6.4.3
Simple log filter	filtering a log by answering simple questions, see Fig. 12.6(b)
Dotted chart analysis	creates a dotted chart showing all events at a glance, see Section 8.2
Trace alignment	similar to dotted chart, but now events are aligned based on their context rather than time [37]
Guide tree miner	clusters cases in a tree based on similarities [36]
Social network miner	creates a social network based on a selected criterion, see Fig. 10.6
LTL checker	checks a property expressed in terms of LTL [6]
Fitness	computes fitness of Petri net based on event log
ETConformance	checks conformance by counting "escaping edges" from the state space of the log to the state space of the model [100]
Replay log on flexible model	conformance checker based on A^* algorithm [25]; can also be applied to Petri nets, C-nets and YAWL models
PomPom	automatically abstracts from infrequently visited parts of a Petri net, see also Section 13.1.3 showing the same idea using fuzzy models
Transition system analyzer	creates a model to predict the remaining flow time, see Section 9.4 and [17, 21]

Assignment 3

- Goal: use a tool (ProM) to perform process mining
- Real-world event data from a financial institution
- Visualize the data
- Extract a sensible model
- Compare algorithms for doing so
- Answer some business-related questions
- Focus on high quality report

Lab session April 20

- Practice with Petri nets on paper
- Snellius room 174

Credits

Lecture partially based on (slides of the (previous edition of the)) course book: W. van der Aalst, *Process Mining: Data Science in Action*, 2nd edition, Springer, 2016.