# Process Mining

COMP9313: Big Data Management

# What’s process mining?

**Wikipedia:**

“Process mining is a family of techniques in the field of process management that support the analysis of business processes based on event logs. During process mining, specialized data mining algorithms are applied to event log data in order to identify trends, patterns and details contained in event logs recorded by an information system”

**processmining.org**

“Process mining techniques allow for extracting information from event logs. For example, the audit trails of a workflow management system or the transaction logs of an enterprise resource planning system can be used to discover models describing processes, organizations, and products.”

工作流程中的审计跟踪

(*) http://www.processmining.org

image: experienceswithlanguage.wordpress.com

# What’s process mining?

Process mining is about discovering what people really do in practice.

HTNG Check In Process

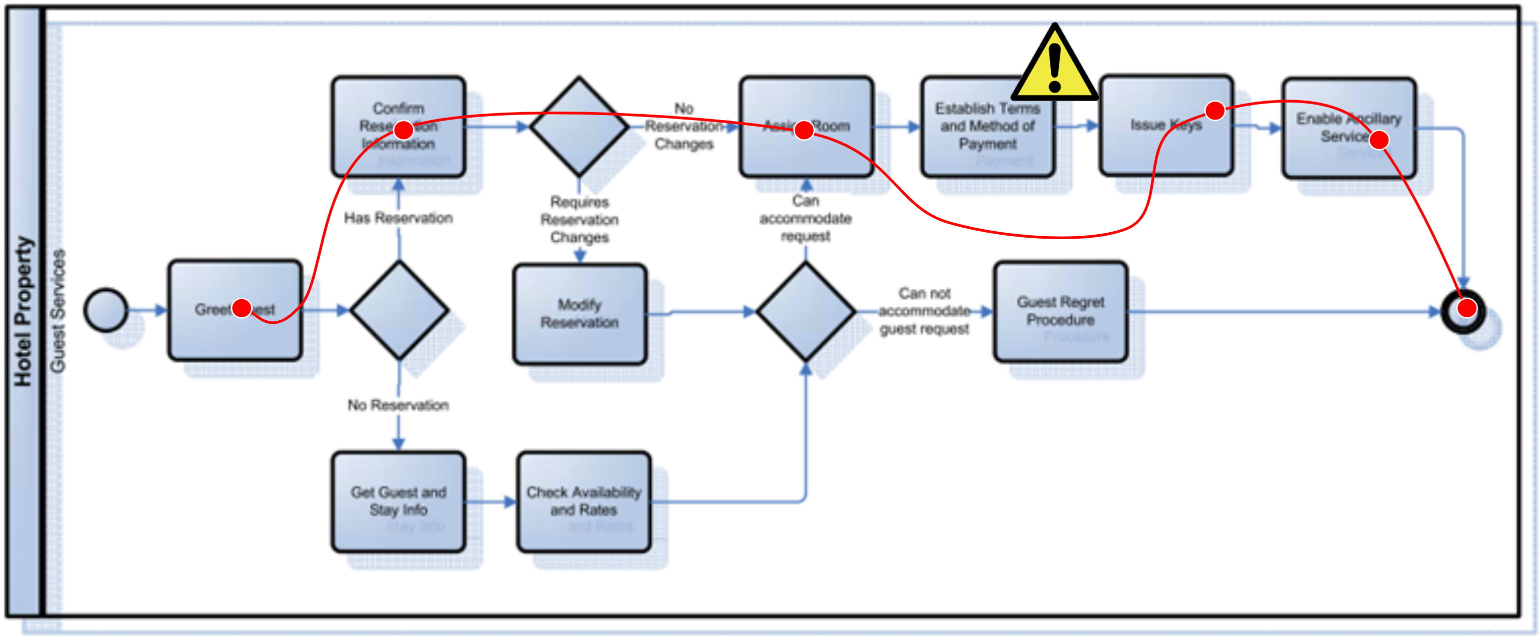
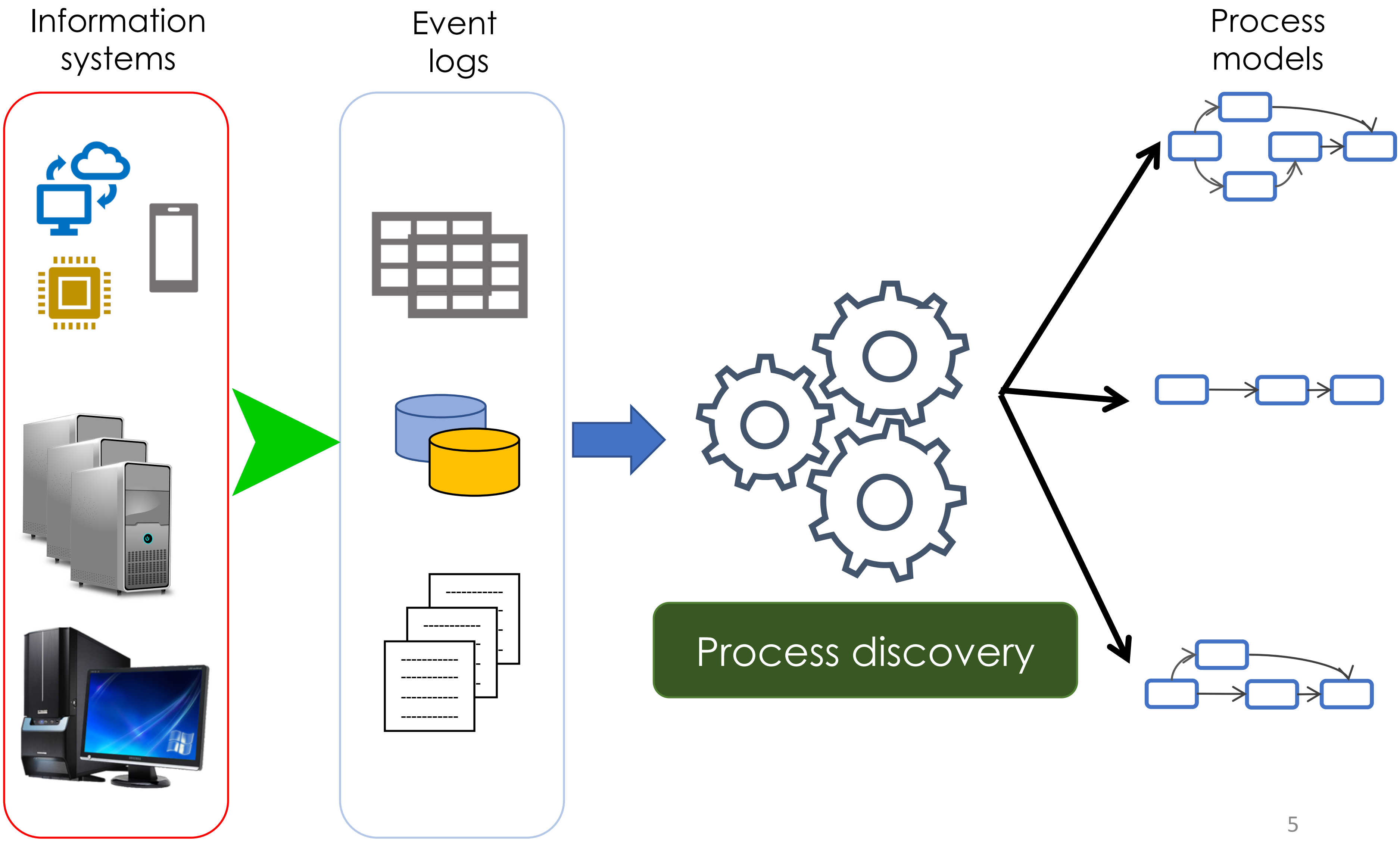


image: wiki.htng.org

# What can I do with process mining?



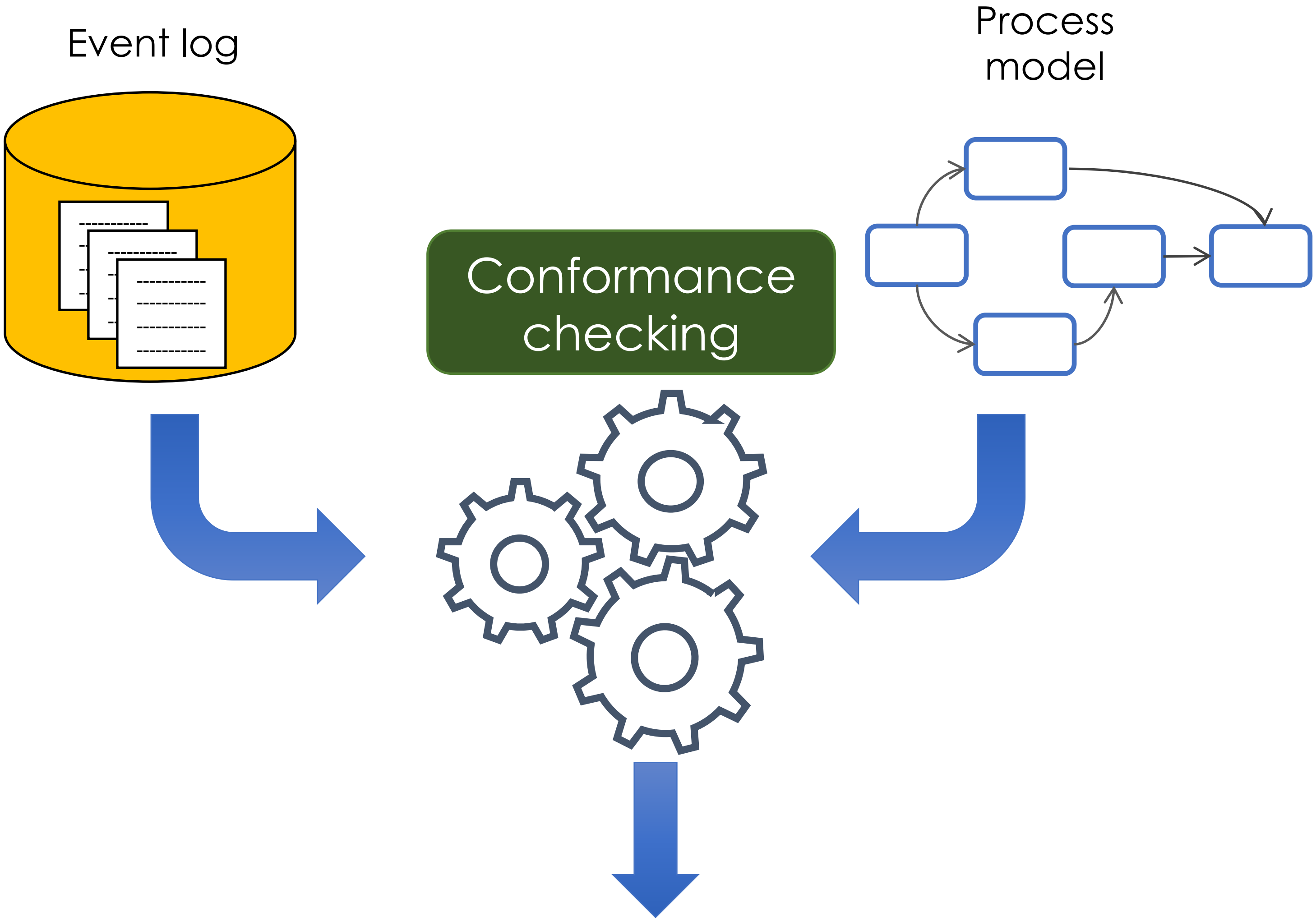
Information systems

Event logs

Process models

Process discovery

# What can I do with process mining?



Event log

Process model

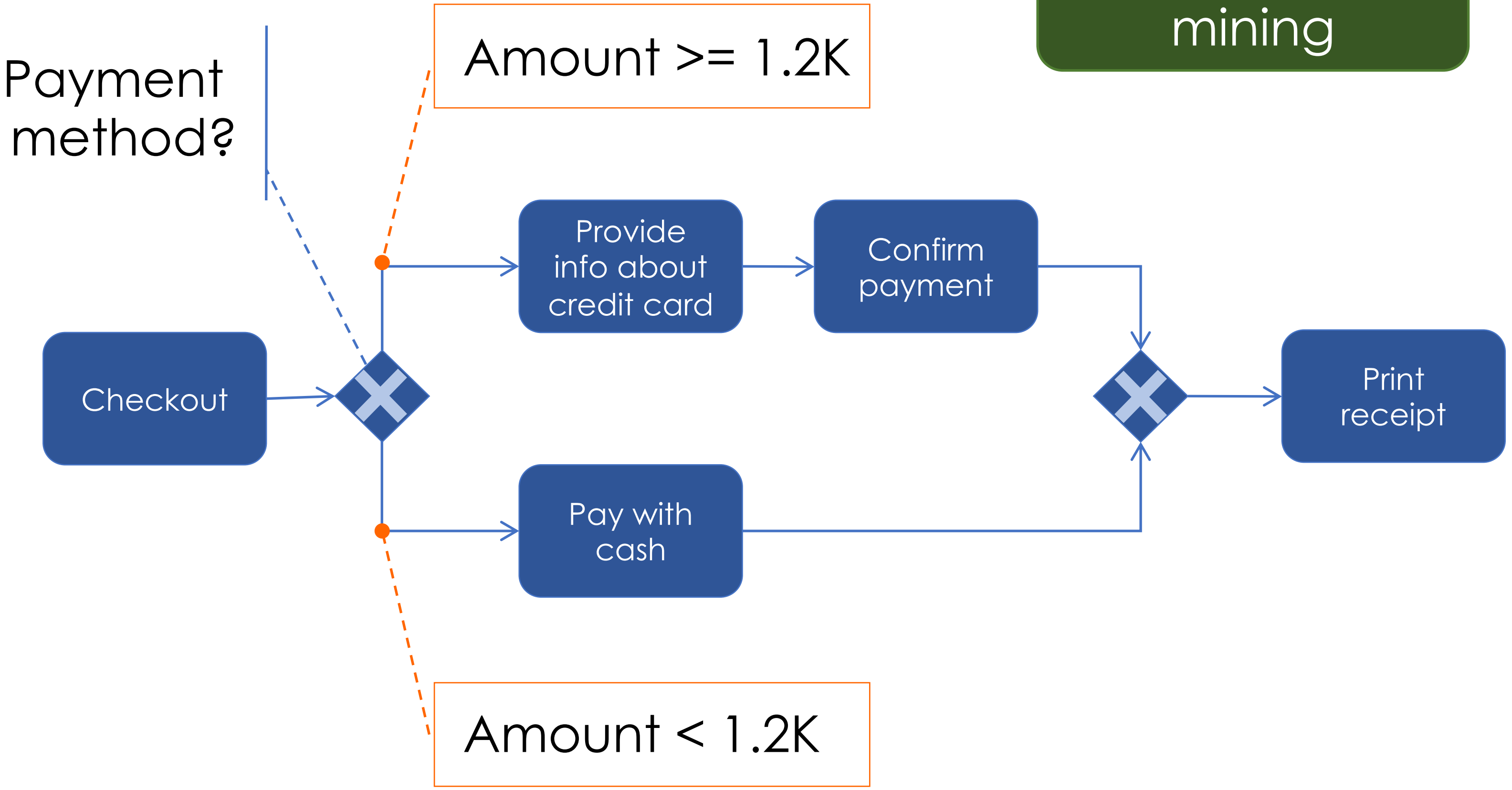
Conformance checking

Conformant / Non conformant

to what extent?

# What can I do with process mining?

Decision point mining



# Data mining vs. Process mining

Data mining

- association rules
- graphs
- sequences (of items)
- clusters

Process mining

- process models
- control flows
- decision points
- process execution data

# In this lecture…

process discovery

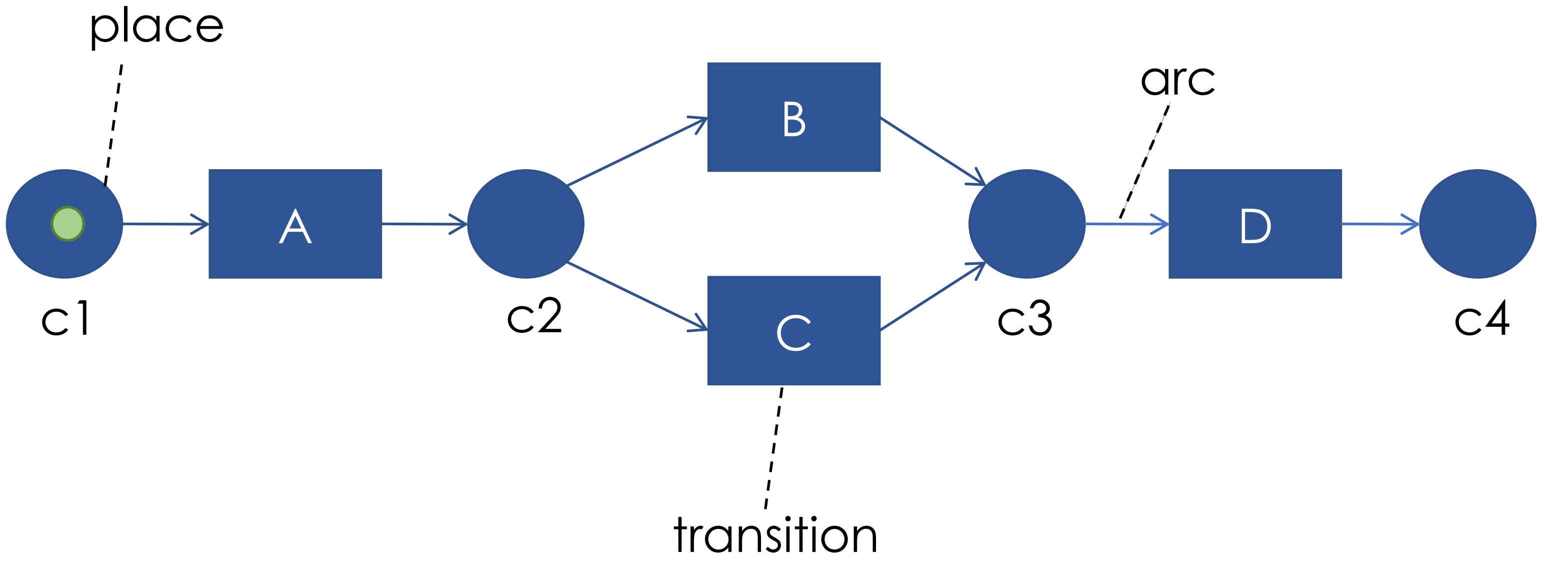
conformance checking

decision point mining

# Petri nets for business process modeling

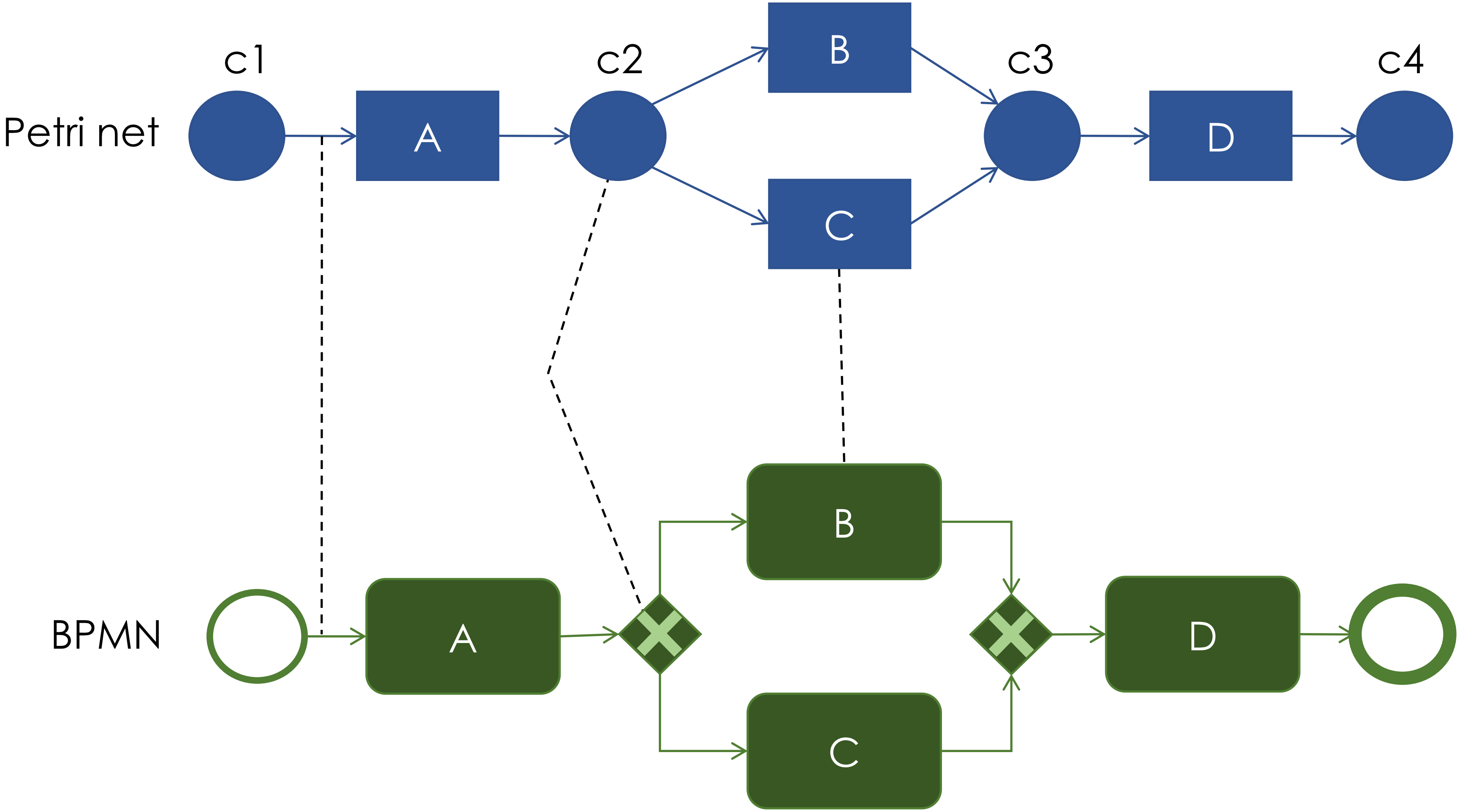
# What’s a Petri net?

“A Petri net, also known as a place/transition (PT) net, is one of several mathematical modeling languages for the description of distributed systems” (*)



(*) https://en.wikipedia.org/wiki/Petri_net

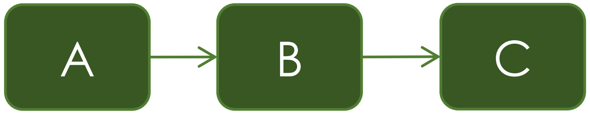
# Relation to BPMN



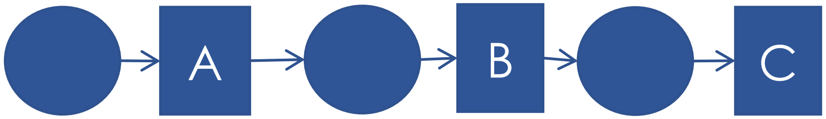
# Relation to BPMN

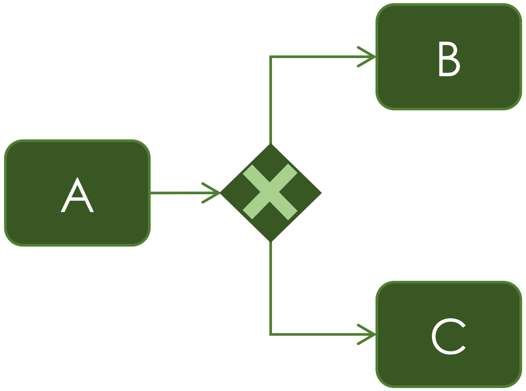
BPMN

Petri Net

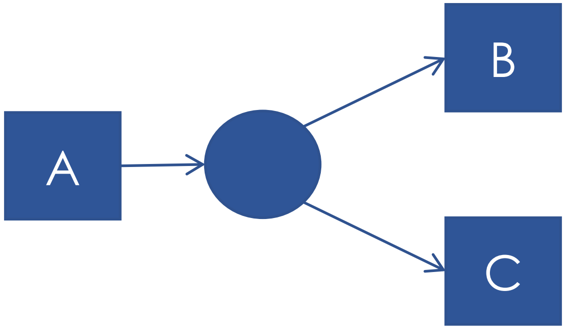


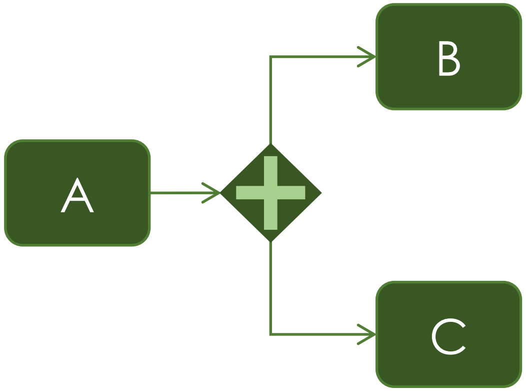
sequential
routing



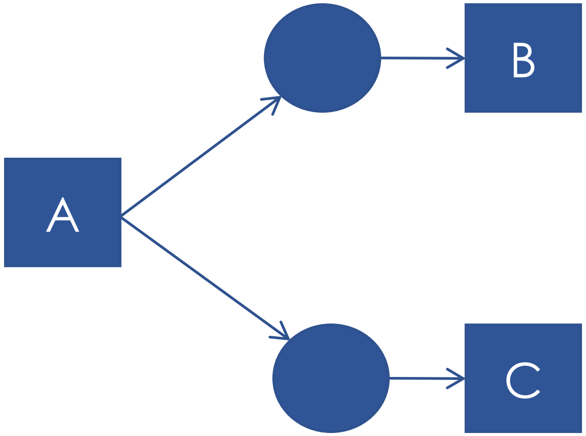


conditional
routing



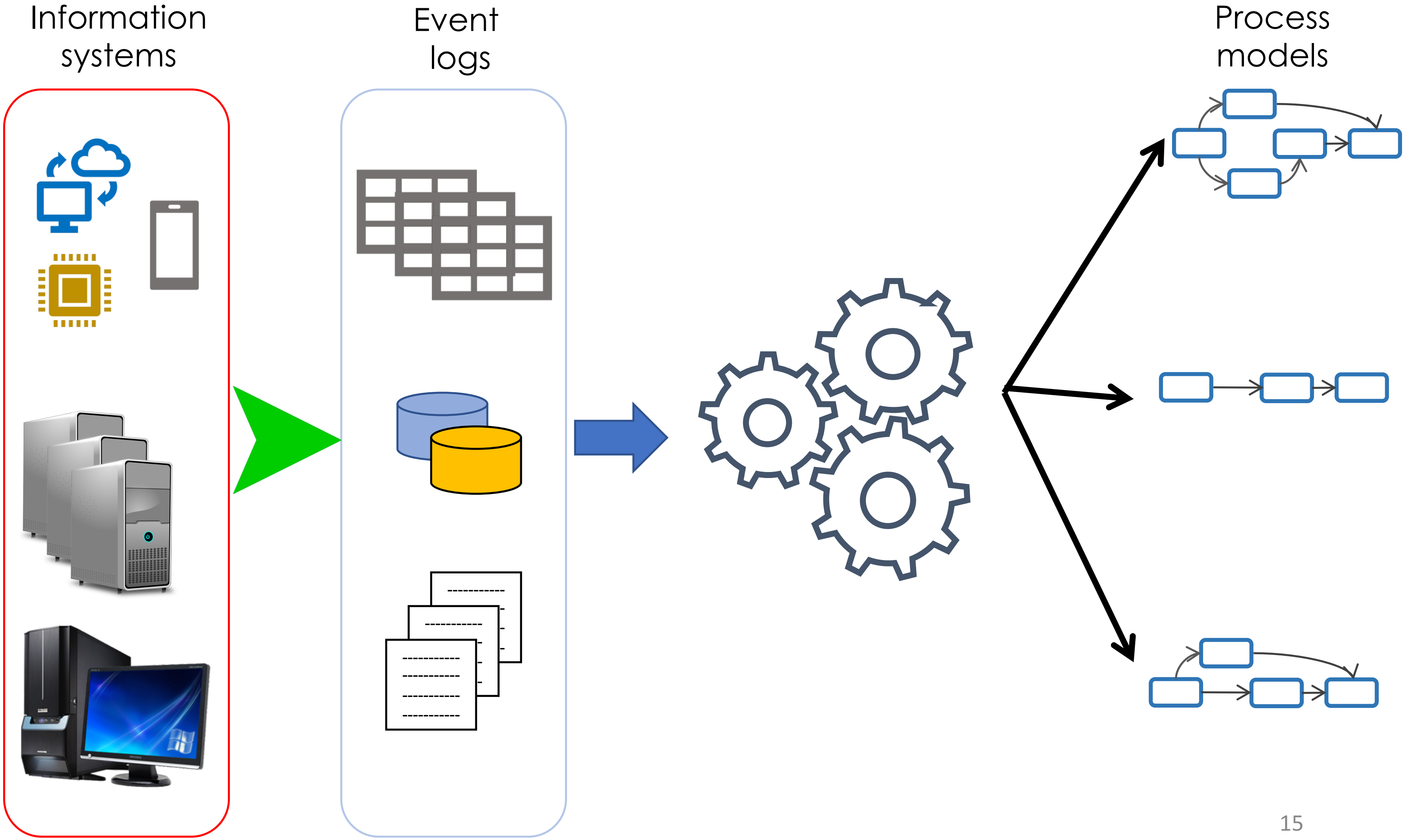


parallel
routing

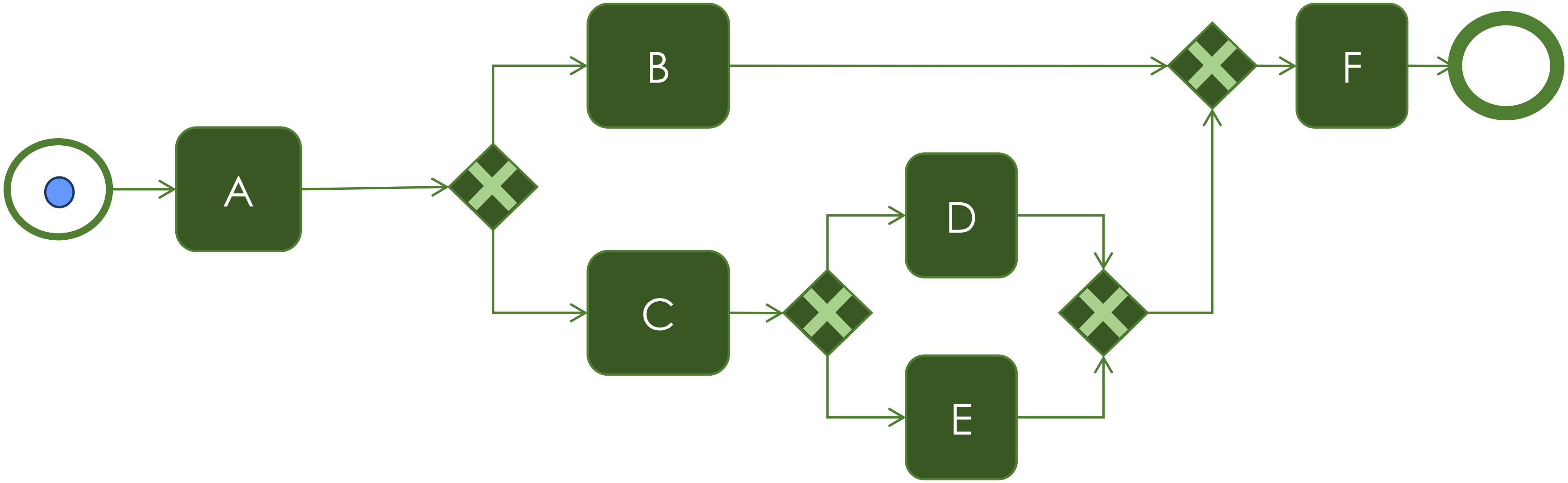


# Process Discovery

# recall…

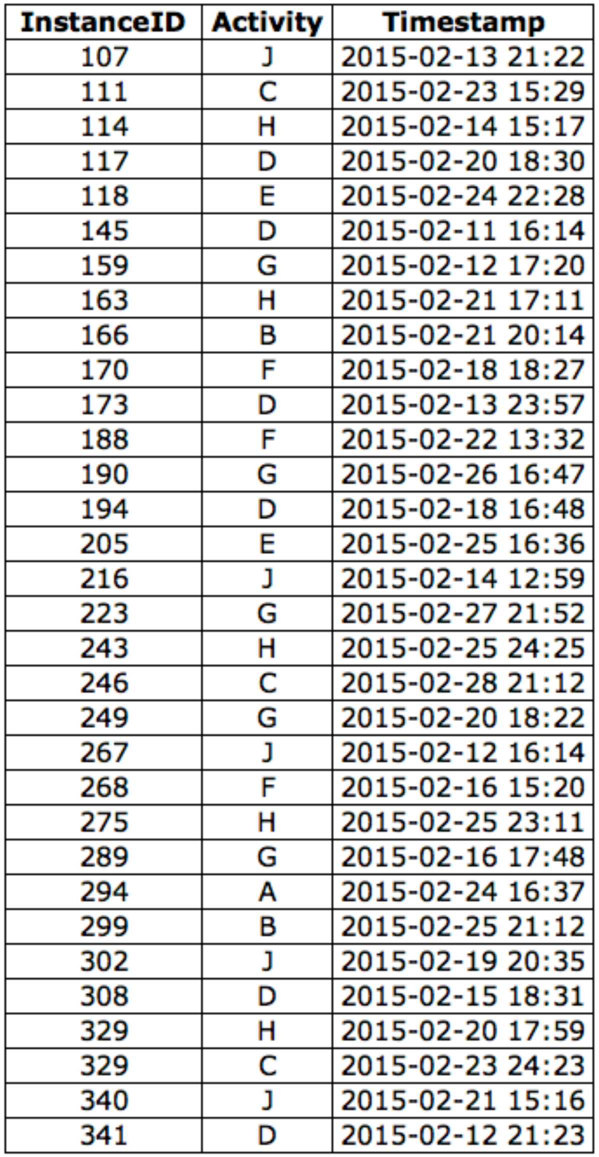


# Event logs



| Case ID | Activity | Timestamp |
|---|---|---|
| 1 | A | 2019-03-25 11:15:01 |
| 1 | C | 2019-03-25 11:15:05 |
| 1 | D | 2019-03-25 11:15:10 |
| 1 | F | 2019-03-25 11:15:18 |

# Event logs



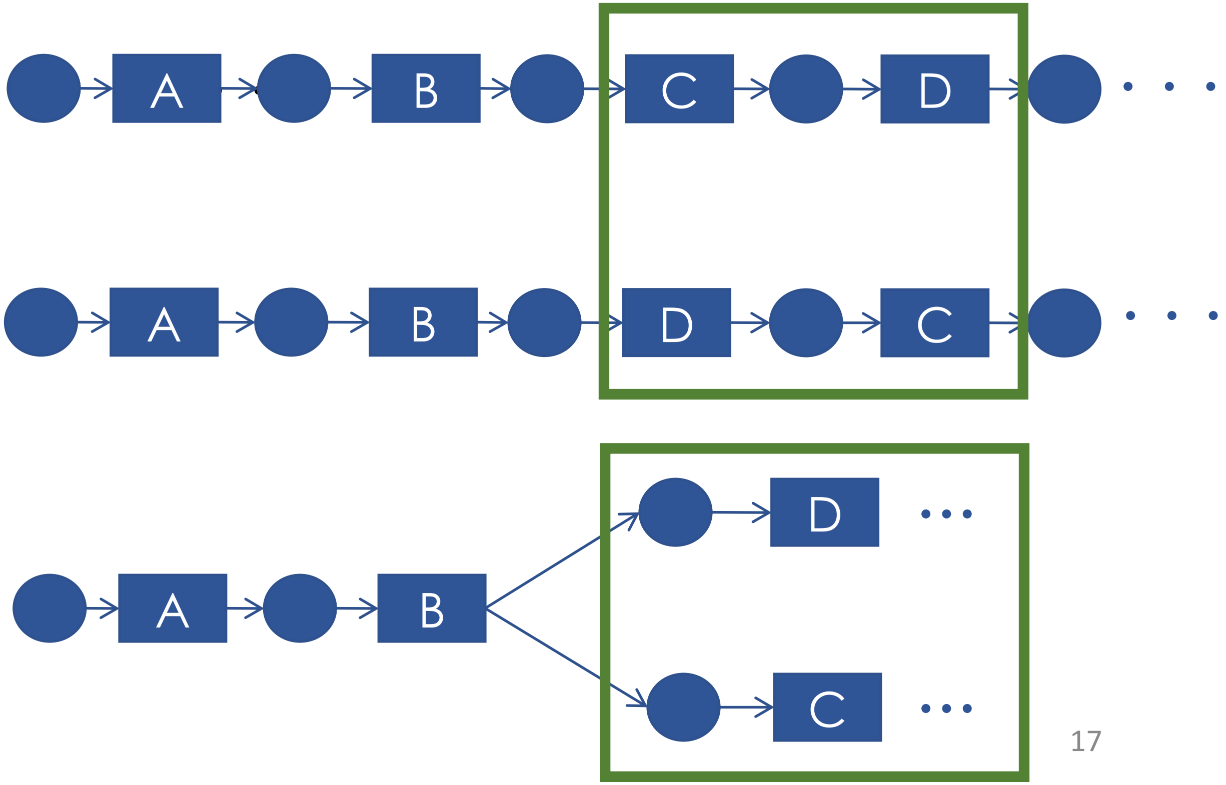
| InstanceID | Activity | Timestamp |
|---|---|---|
| 107 | J | 2015-02-13 21:22 |
| 111 | C | 2015-02-23 15:29 |
| 114 | H | 2015-02-14 15:17 |
| 117 | D | 2015-02-20 18:30 |
| 118 | E | 2015-02-24 22:28 |
| 145 | D | 2015-02-11 16:14 |
| 159 | G | 2015-02-12 17:20 |
| 163 | H | 2015-02-21 17:11 |
| 166 | B | 2015-02-21 20:14 |
| 170 | F | 2015-02-18 18:27 |
| 173 | D | 2015-02-13 23:57 |
| 188 | F | 2015-02-22 13:32 |
| 190 | G | 2015-02-26 16:47 |
| 194 | D | 2015-02-18 16:48 |
| 205 | E | 2015-02-25 16:36 |
| 216 | J | 2015-02-14 12:59 |
| 223 | G | 2015-02-27 21:52 |
| 243 | H | 2015-02-25 24:25 |
| 246 | C | 2015-02-28 21:12 |
| 249 | G | 2015-02-20 18:22 |
| 267 | J | 2015-02-12 16:14 |
| 268 | F | 2015-02-16 15:20 |
| 275 | H | 2015-02-25 23:11 |
| 289 | G | 2015-02-16 17:48 |
| 294 | A | 2015-02-24 16:37 |
| 299 | B | 2015-02-25 21:12 |
| 302 | J | 2015-02-19 20:35 |
| 308 | D | 2015-02-15 18:31 |
| 329 | H | 2015-02-20 17:59 |
| 329 | C | 2015-02-23 24:23 |
| 340 | J | 2015-02-21 15:16 |
| 341 | D | 2015-02-12 21:23 |

Traces:

case 1: A B C D E F

case 2: A B D C F I

case 3: A D E F G H J



# Example:α - Algorithm

Event log

```
instance 1 : task A
instance 2 : task A
instance 3 : task A
instance 3 : task B
instance 1 : task B
instance 1 : task C
instance 2 : task C
instance 4 : task A
instance 2 : task B
instance 2 : task D
instance 5 : task E
instance 4 : task C
instance 1 : task D
instance 3 : task C
instance 3 : task D
instance 4 : task B
instance 5 : task F
instance 4 : task D
```

- Direct succession: x>y iff for some case x is directly followed by y.
- Causality: x→y iff x>y and not y>x
- Parallel: x||y iff x>y and y>x
- Choice: x#y iff not x>y and not y>x

```
trace1: A B C D
trace2: A C B D
trace3: A B C D
trace4: A C B D
trace5: E F
```

A>B
A>C
B>C
B>D
C>B
C>D
E>F

A→B
A→C
B→D
C→D
E→F

B||C

A#E
B#E
C#E
D#E
A#F
B#F
C#F
D#F
...

Reference: Workflow mining: A survey of issues and approaches (van der Aalst et al.)

# Example:α - Algorithm

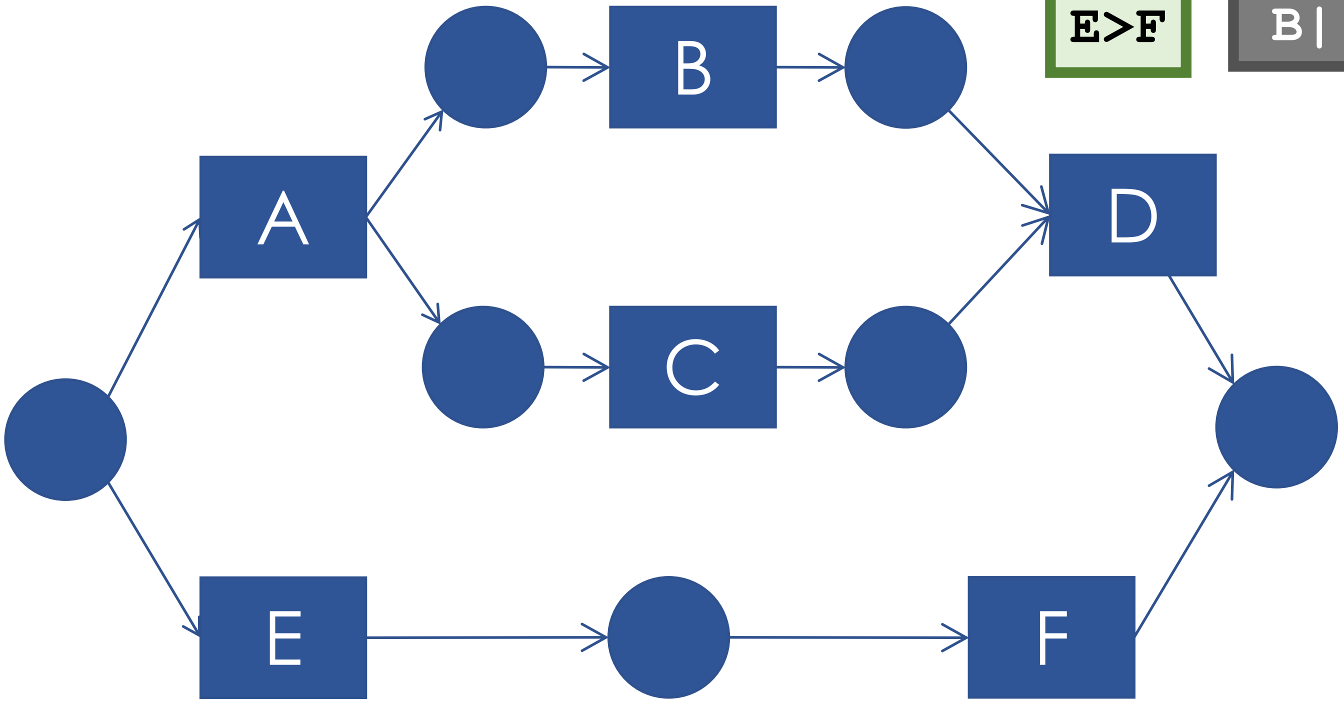
```
Trace1: A B C D
Trace2: A C B D
Trace3: A B C D
Trace4: A C B D
Trace5: E F
```

A>B
A>C
B>C
B>D
C>B
C>D
E>F

A→B
A→C
B→D
C→D
E→F

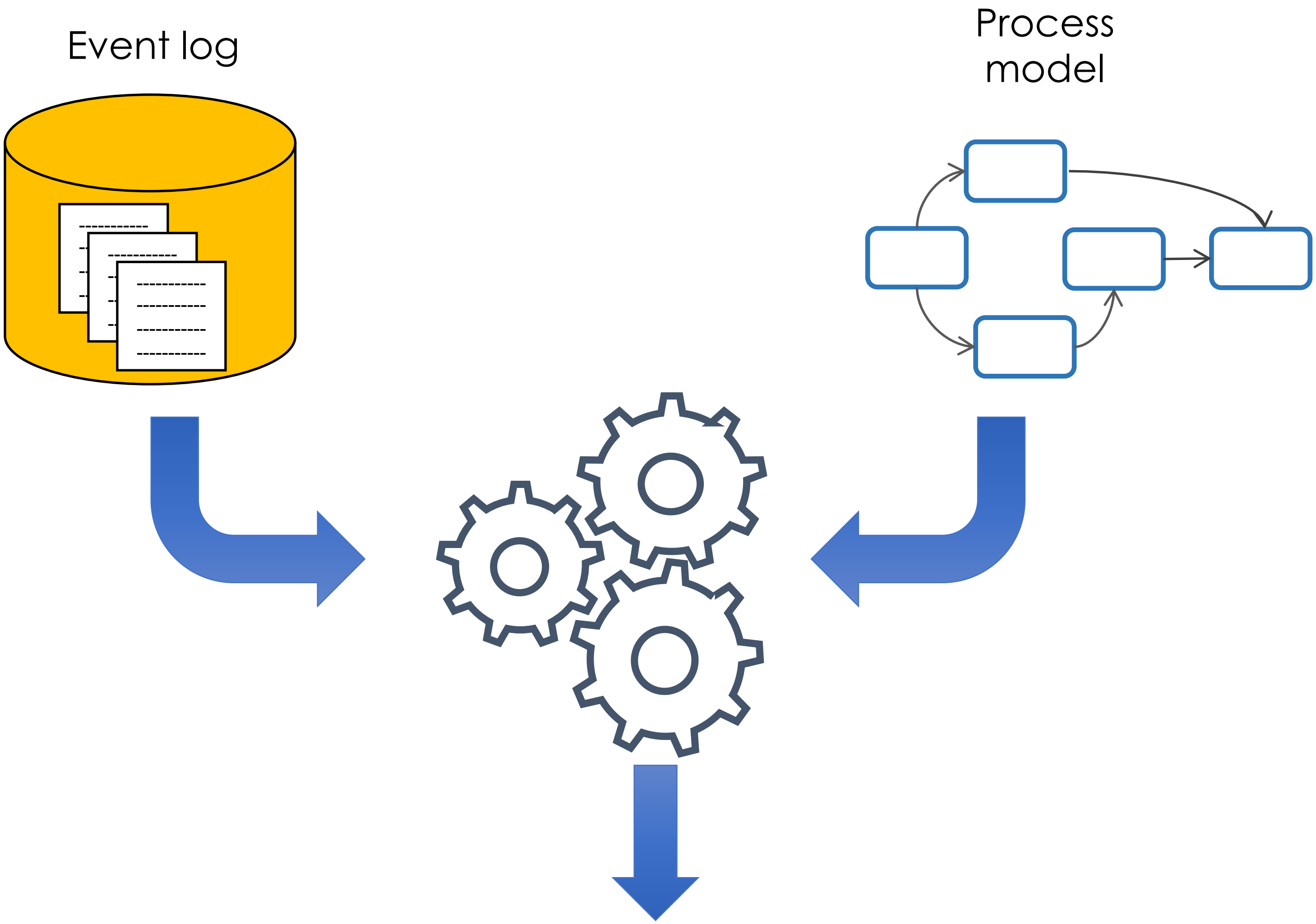
B| |C

A#E
B#E
C#E
D#E
A#F
B#F
C#F
D#F



# Conformance Checking

# recall…



Event log

Process model

Conformant / Non conformant

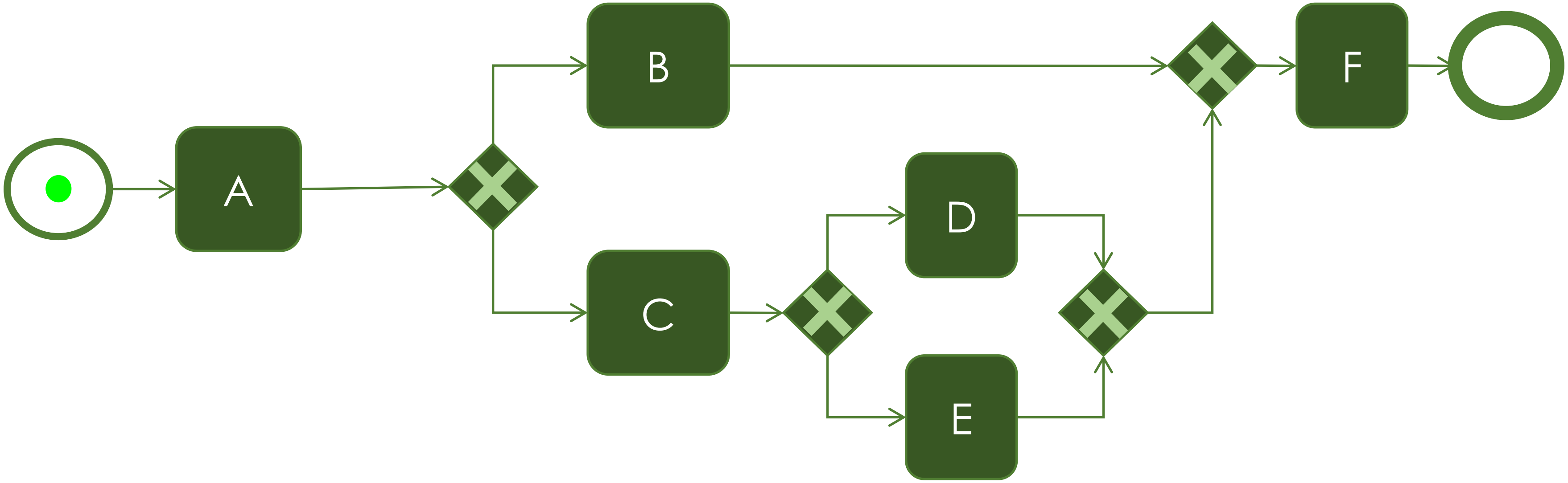
to what extent?

# Main idea of conformance checking

trace 1: A C D F ✓

trace 2: A D F

Trace 3: A B D F



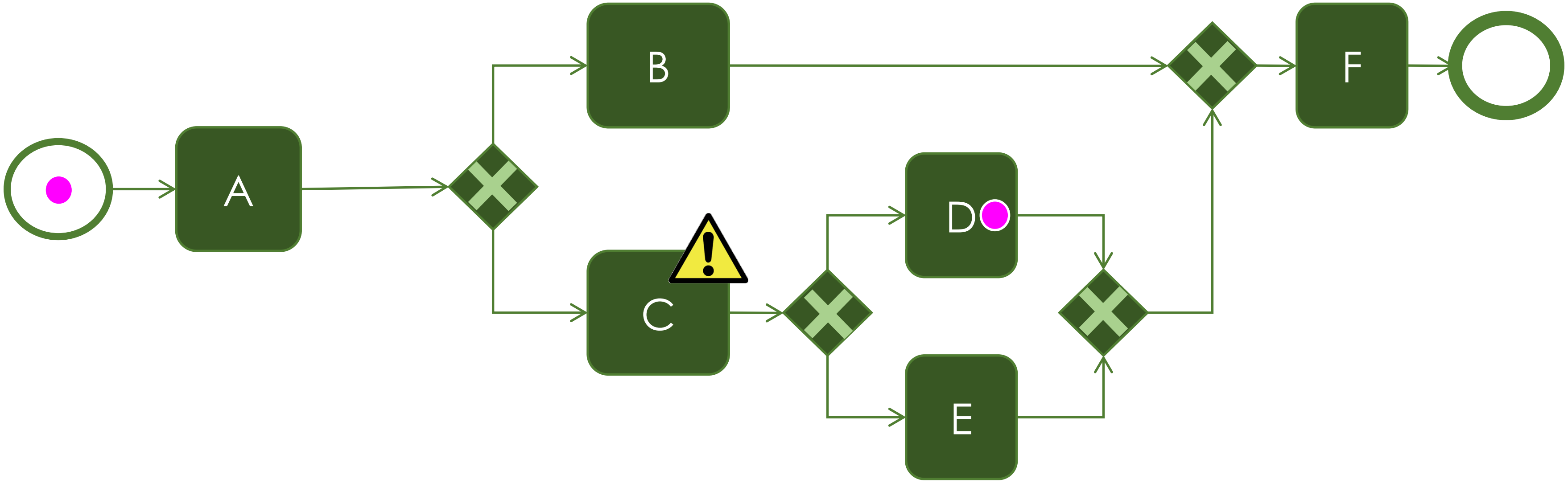
Reference: Conformance checking of processes based on monitoring real behavior (Ronizat et al., 2008)

# Main idea of conformance checking

trace 1: A C D F ✓

trace 2: A D F ✗

Trace 3: A B D F

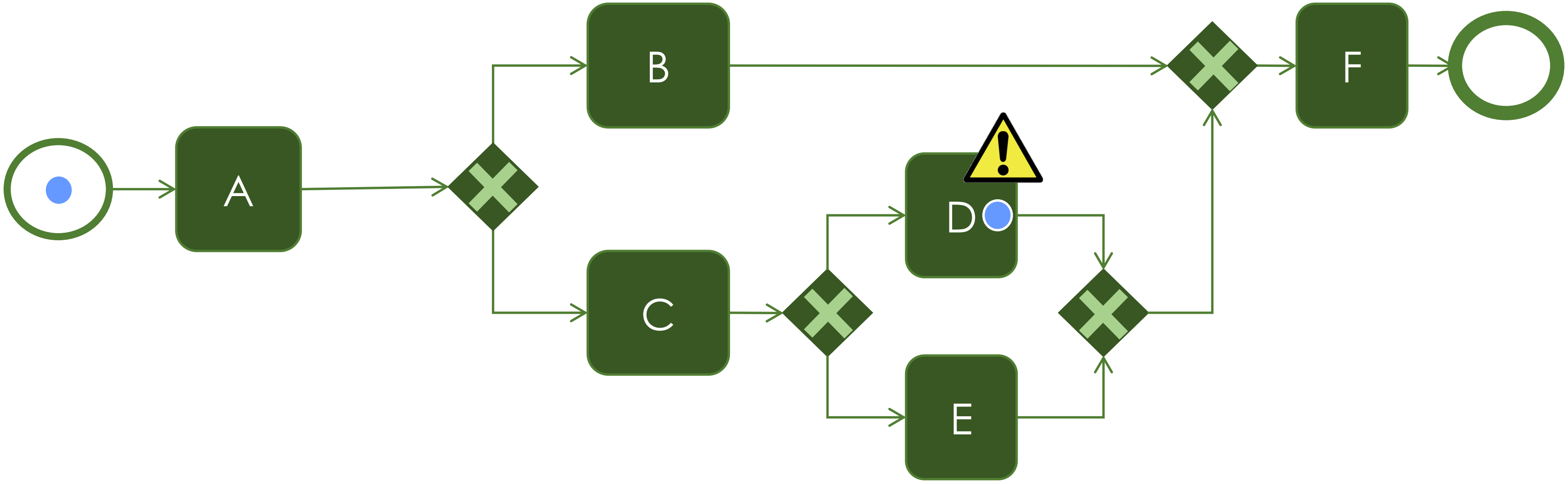


# Main idea of conformance checking

trace 1: A C D F ✓

trace 2: A D F ✗

Trace 3: A B D F ✗



# Metrics for conformance checking

## Behavioral appropriateness (precision)

How much extra-behavior is allowed by the model?

## Fitness

How well is model able to replay the log?

## Structural appropriateness

How many "unnecessary" redundant and invisible tasks are there in the model?

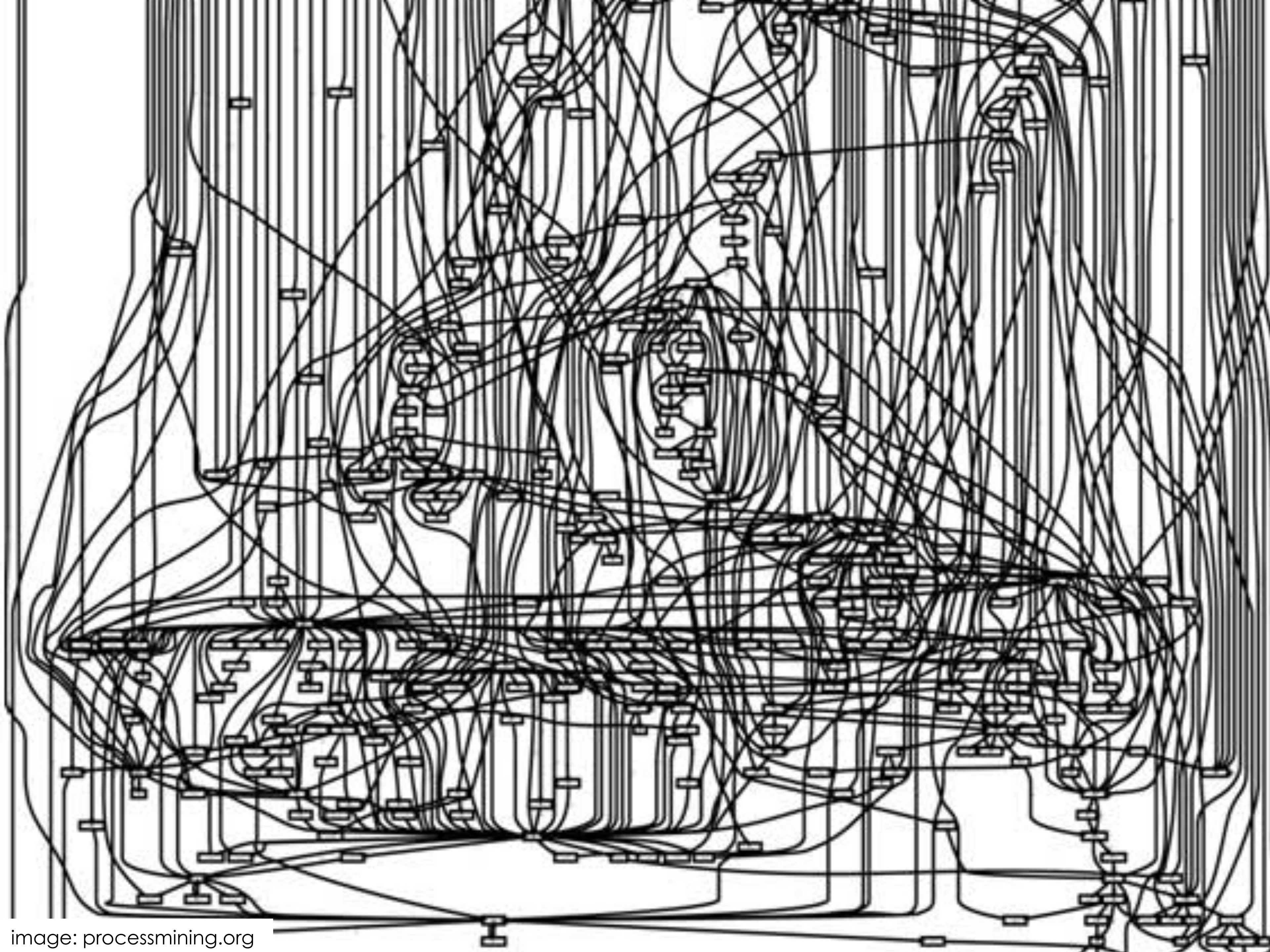


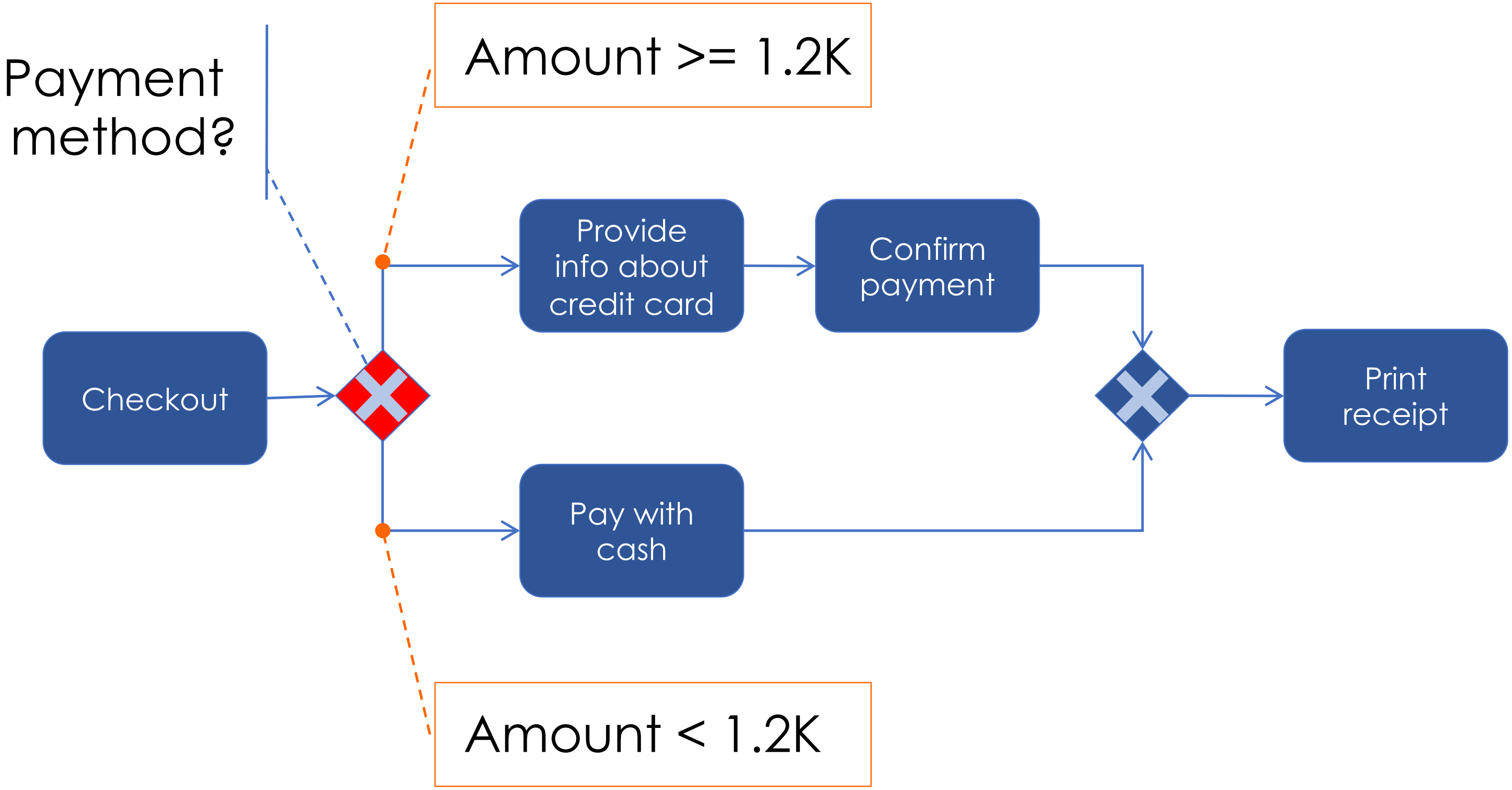
image: processmining.org

# conformant? not conformant? so what?

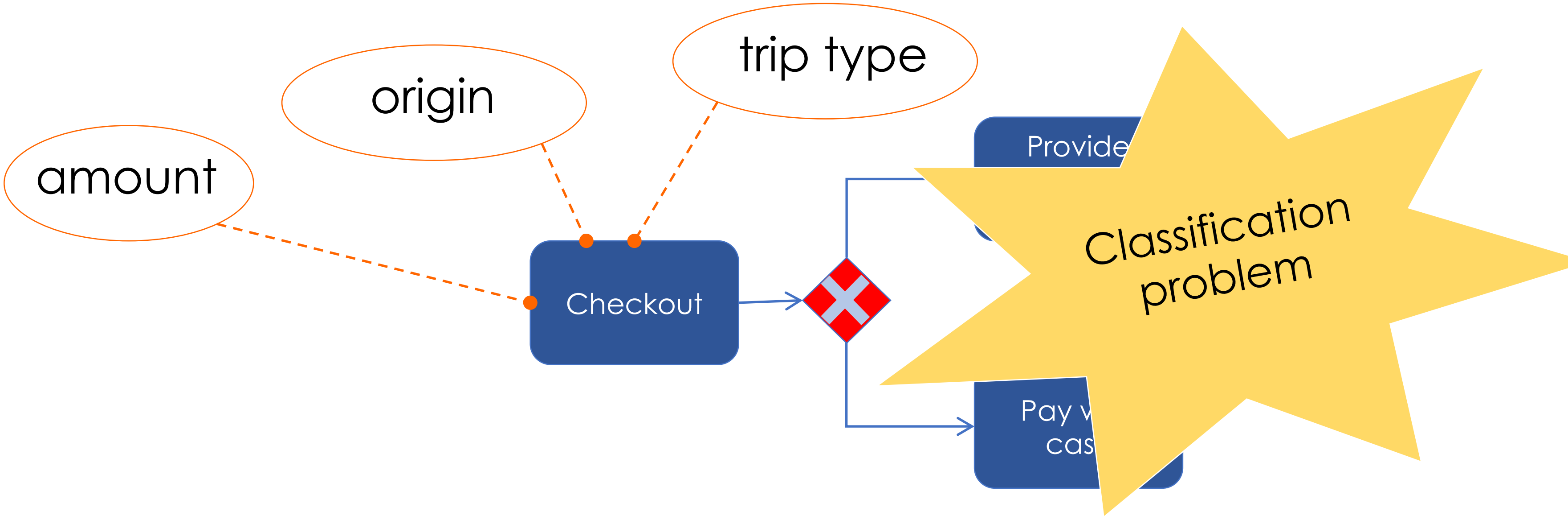
image: experienceswithlanguage.wordpress.com

# Decision Point Mining

# recall…


Payment method?
Amount >= 1.2K
Provide info about credit card
Confirm payment
Checkout
Print receipt
Pay with cash
Amount < 1.2K

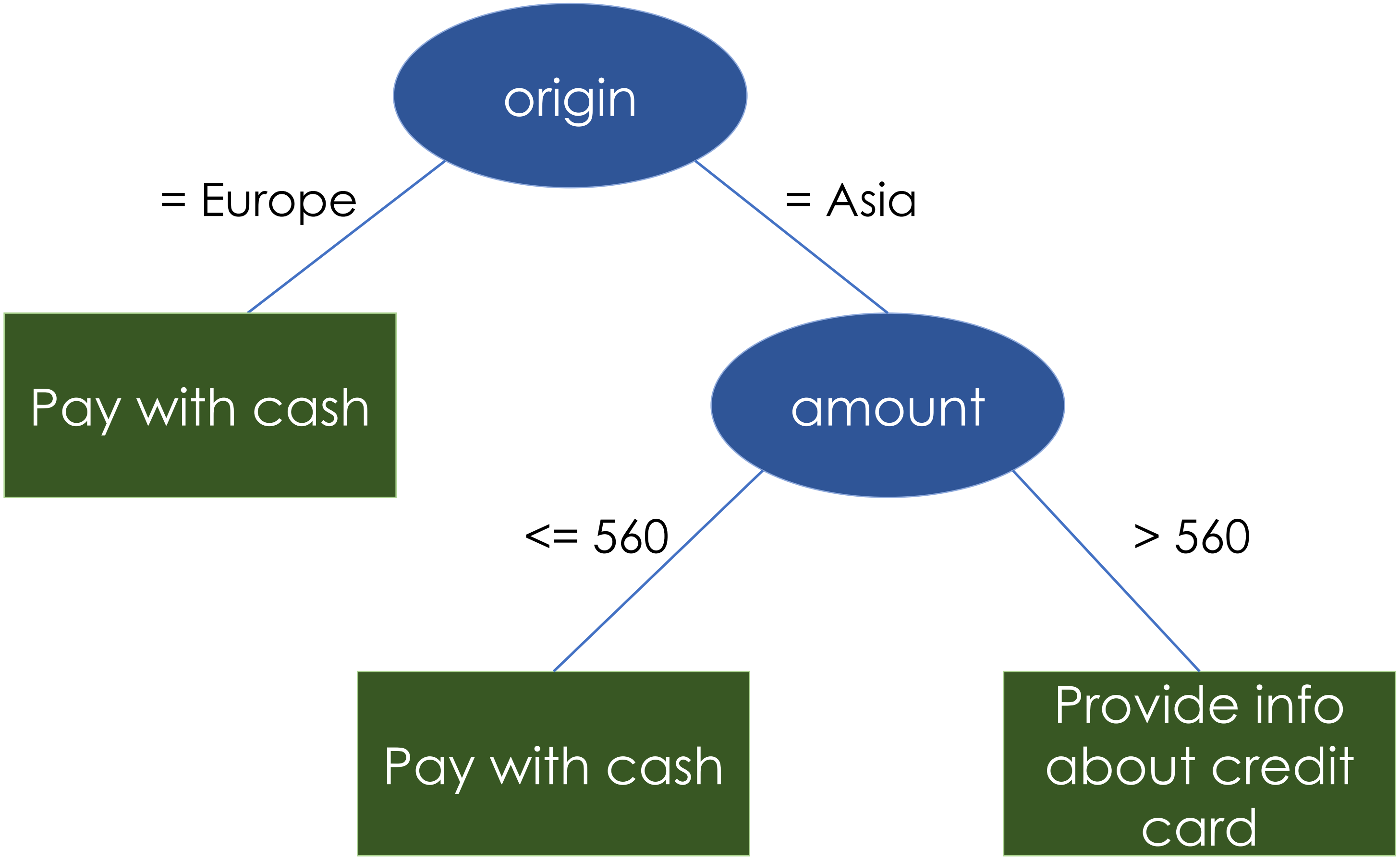
# How does it work?



| Case ID | amount | origin | trip type | branch |
|---|---|---|---|---|
| 1 | 243 | Europe | Leisure | Pay with cash |
| 2 | 325 | Europe | Business | Pay with cash |
| 3 | 1021 | Asia | Business | Provide info about credit card |
| 4 | 560 | Asia | Leisure | Pay with cash |

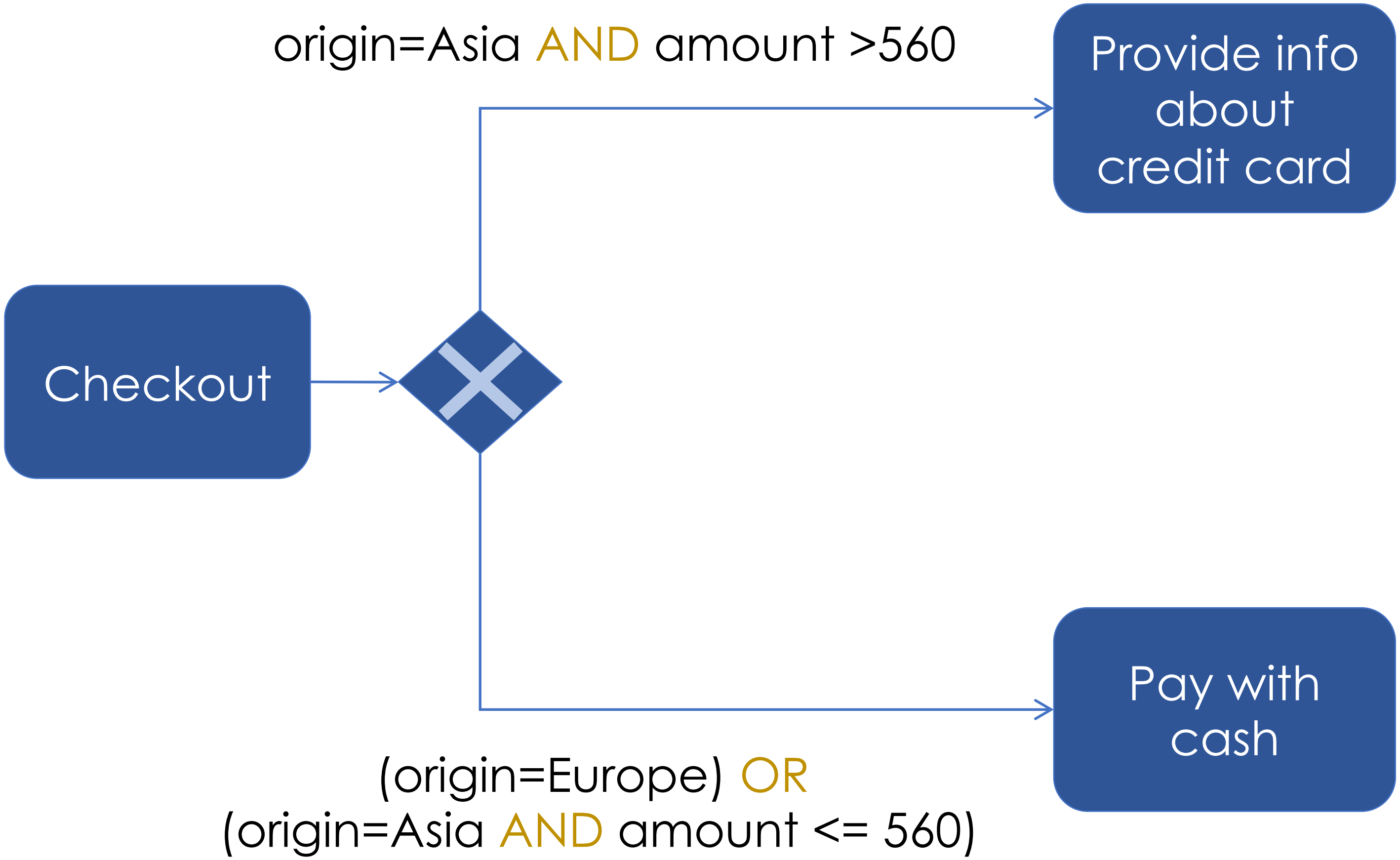
Reference: Decision mining in business process (Rozinat & van der Aalst, 2006)

# Decision point mining with decision trees



Reference: Machine Learning (T. Mitchell, 1997)

# Decision point mining with decision trees


origin=Asia AND amount >560
Provide info about credit card
Checkout
Pay with cash
(origin=Europe) OR
(origin=Asia AND amount <= 560)

Thanks