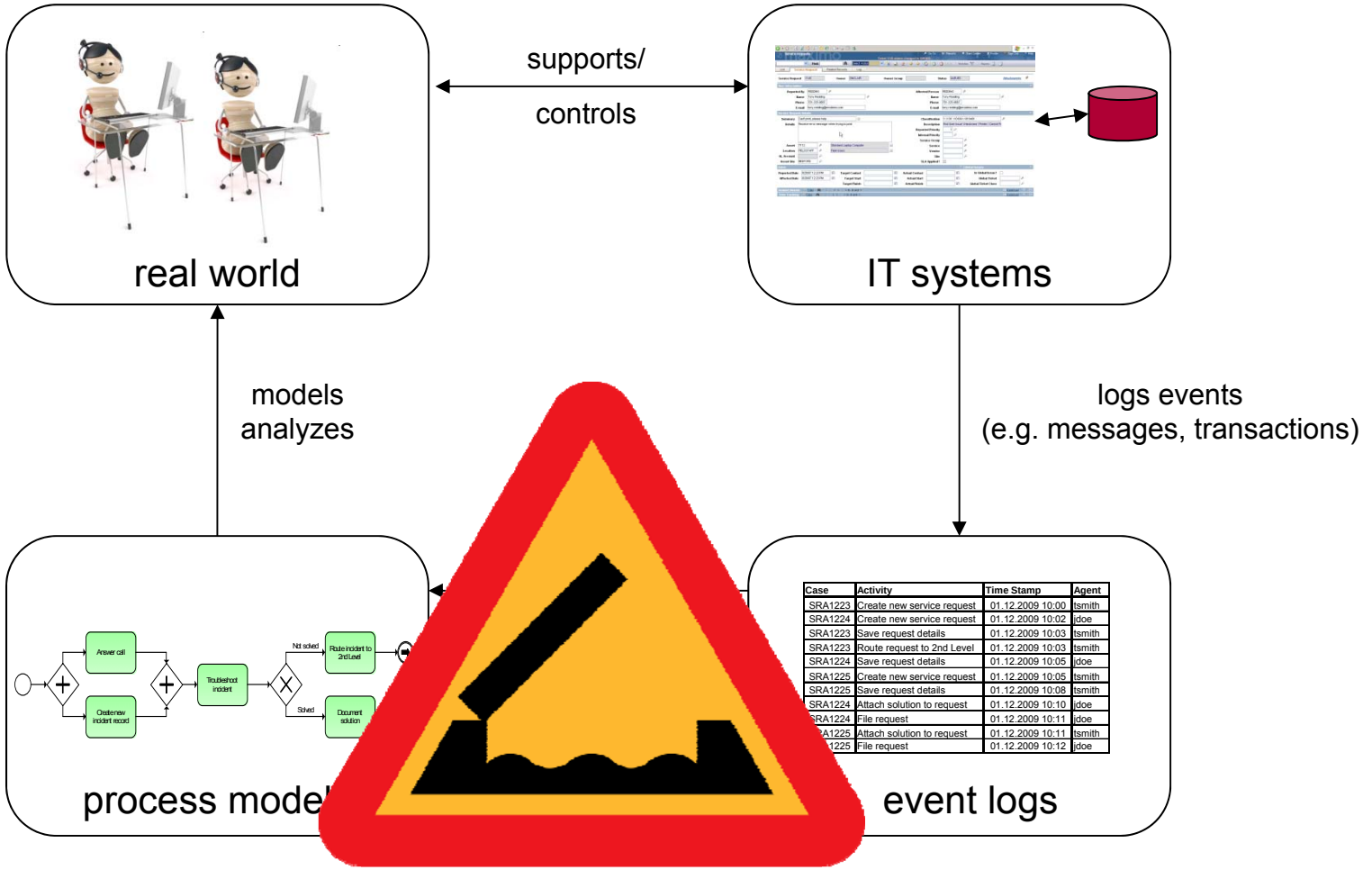
Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

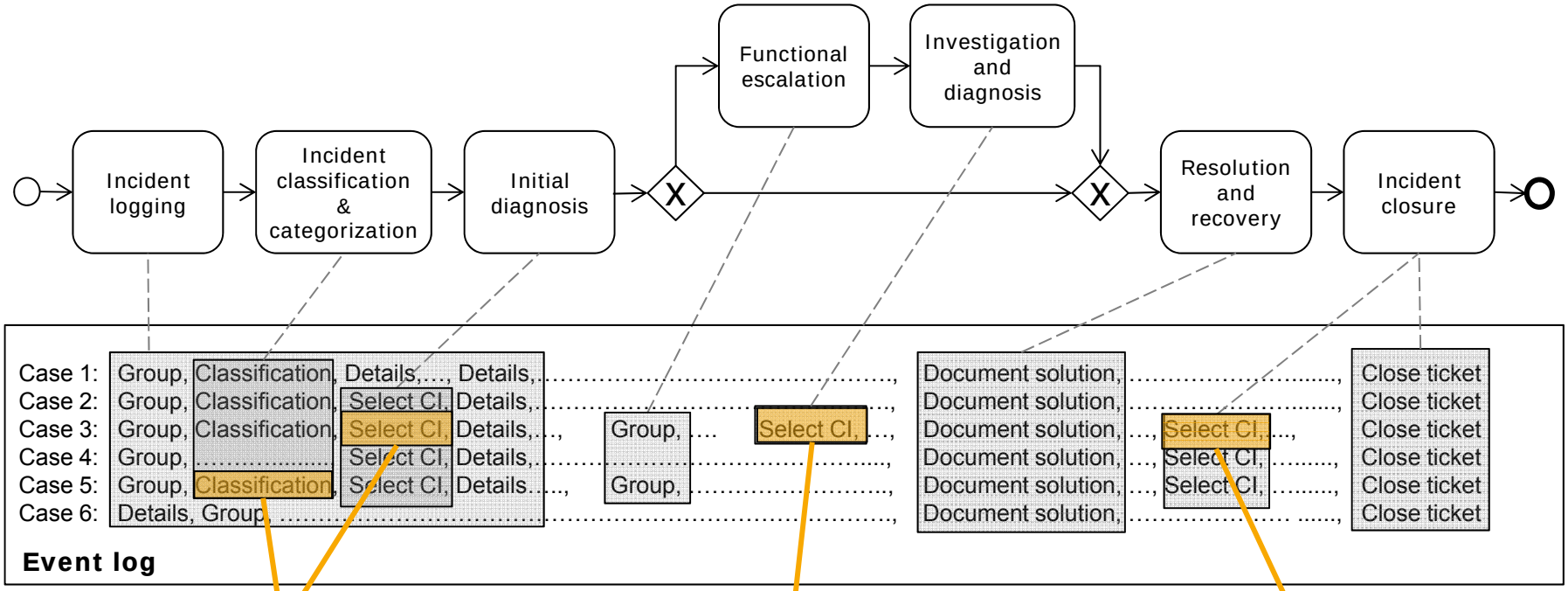
Bridging Abstraction Layers in Process Mining by automated Matching of Events and Activities

Thomas Baier and Jan Mendling

Process Mining



Abstraction of event logs – Example



Trace: Case 3
Event class: Select CI
Timestamp: 12.06.2013 10:25:00
Resource: John Doe
Role: 1st level
Component: Mail client XYZ
Classification: E-Mail

Trace: Case 3
Event class: Select CI
Timestamp: 12.06.2013 16:45:10
Resource: Carla Coe
Role: 2nd level
Component: Mail Server 123

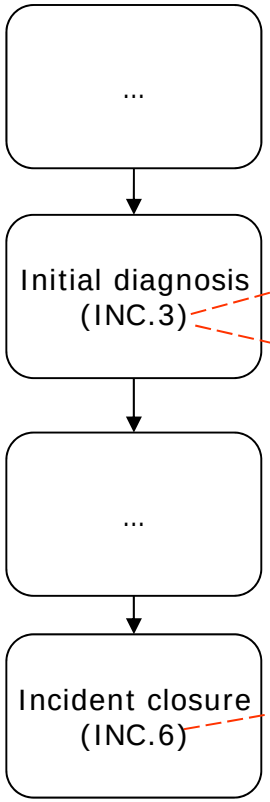
Trace: Case 3
Event class: Select CI
Timestamp: 13.06.2013 09:45:56
Resource: Carla Coe
Role: 2nd level
Component: Hard disk 1234

Challenges

Abstracting event logs with

- a given abstraction level
- n:m relations between events and activities
- concurrency of activities

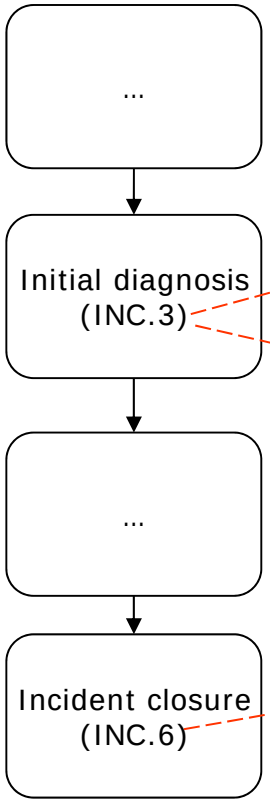
1st step: annotation of process model activities



Model reference	Task	Description
...	...	...
7. **INC.3.**1	Select configuration item	Choose the affected configuration item (CI) from the CMDB
8. **INC.3**.2	Determine affected person	Determine the affected person . If multiple persons are affected by the incident...
...	...	...
12. **INC.6**.3	Assess and improve documentation	Check whether the correct configuration item (**CI**) has been chosen and select the correct **CI** if necessary...

Case	Event	Time
INC12345	…	…
INC12345	Select CI	...
INC12345	Affected person	...
INC12345	…	…
INC12345	Select CI	...
INC12345	...	...

2nd step: matching of activities and events on type level

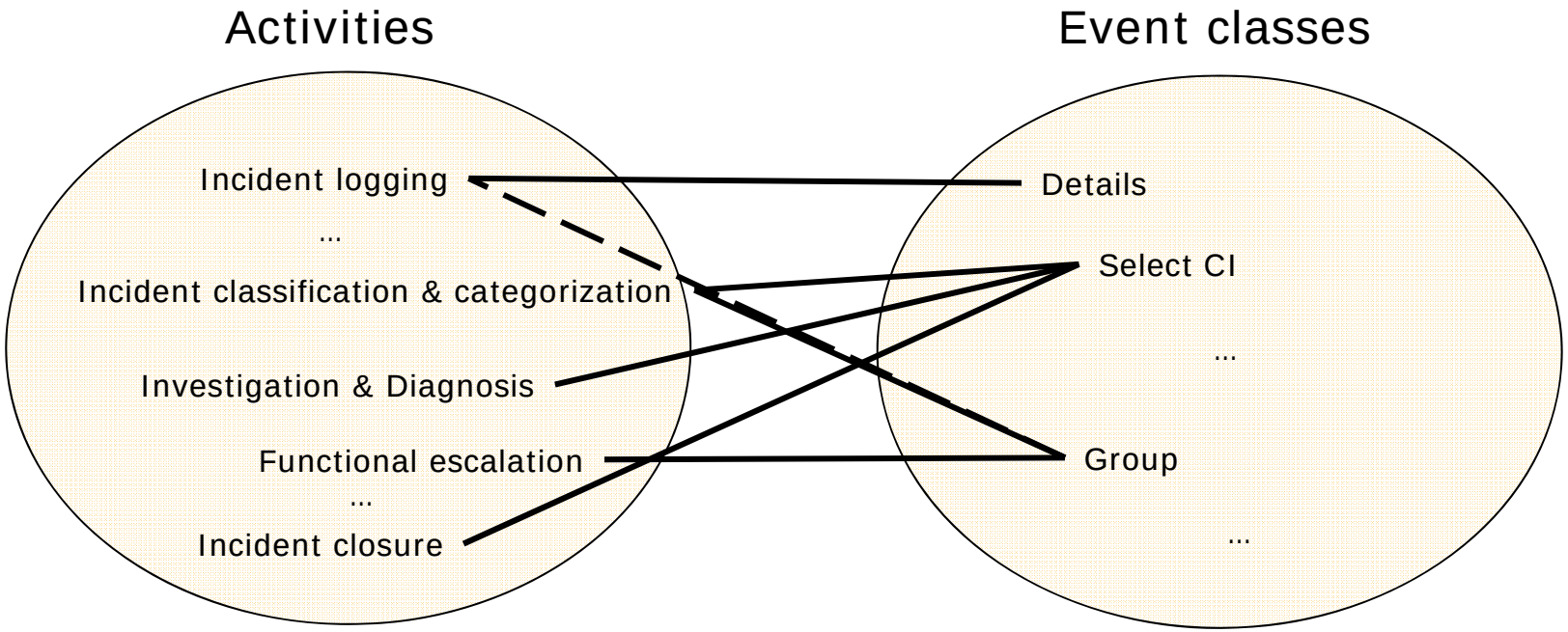


Model reference	Task	Description
...	...	...
7. **INC.3.**1	Select configuration item	Choose the affected configuration item (CI) from the CMDB
8. **INC.3**.2	Determine affected person	Determine the affected person . If multiple persons are affected by the incident..
...	...	...
12. **INC.6**.3	Assess and improve documentation	Check whether the correct configuration item (**CI**) has been chosen and select the correct **CI** if necessary...

1. Determine business objects
2. Match activities and event classes over business objects

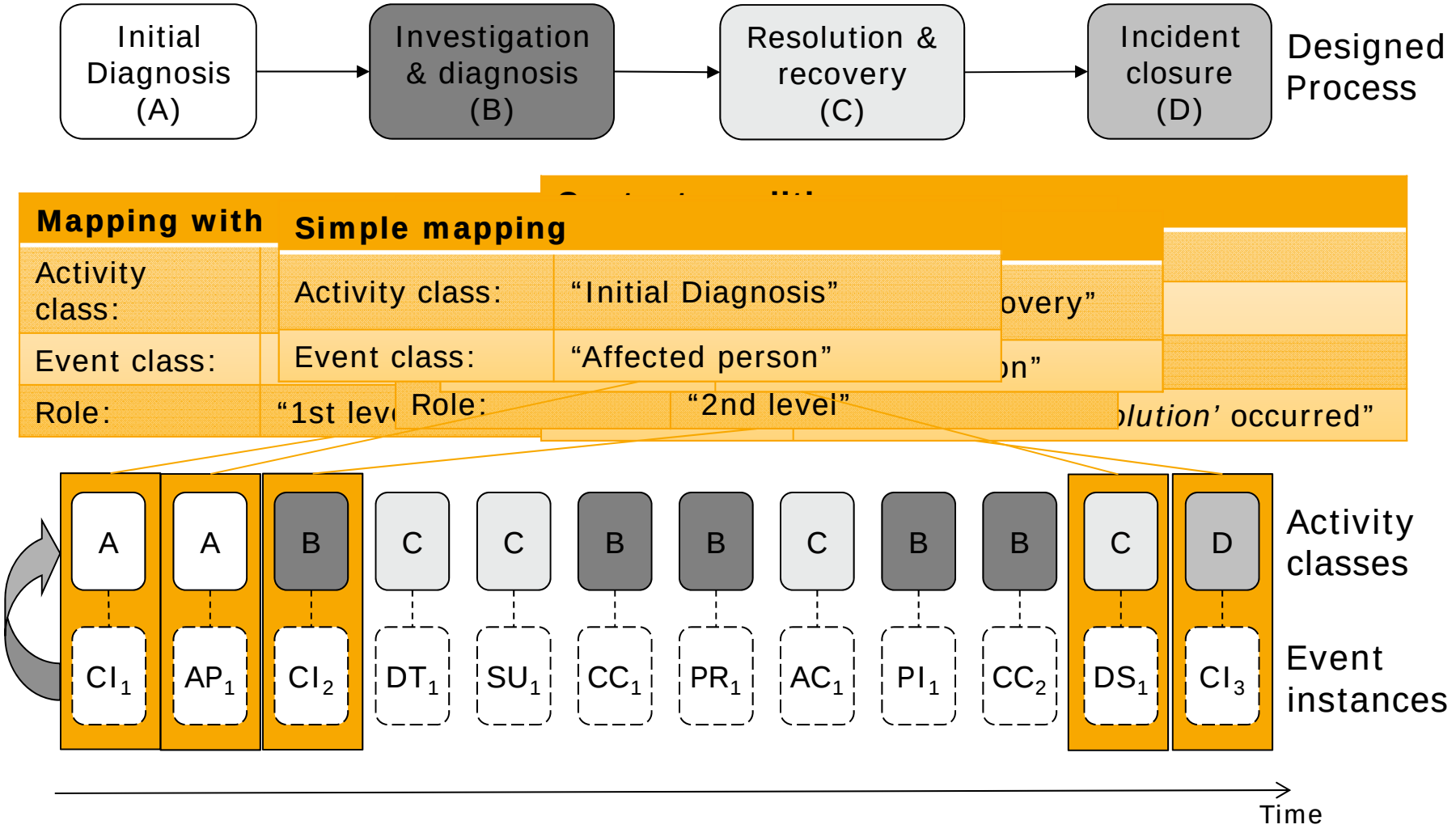
Case	Event	Time
INC12345	…	…
INC12345	Select CI	...
INC12345	Affected person	...
INC12345	…	…
INC12345	Select CI	...
INC12345	...	...

2nd step: matching of activities and events on type level

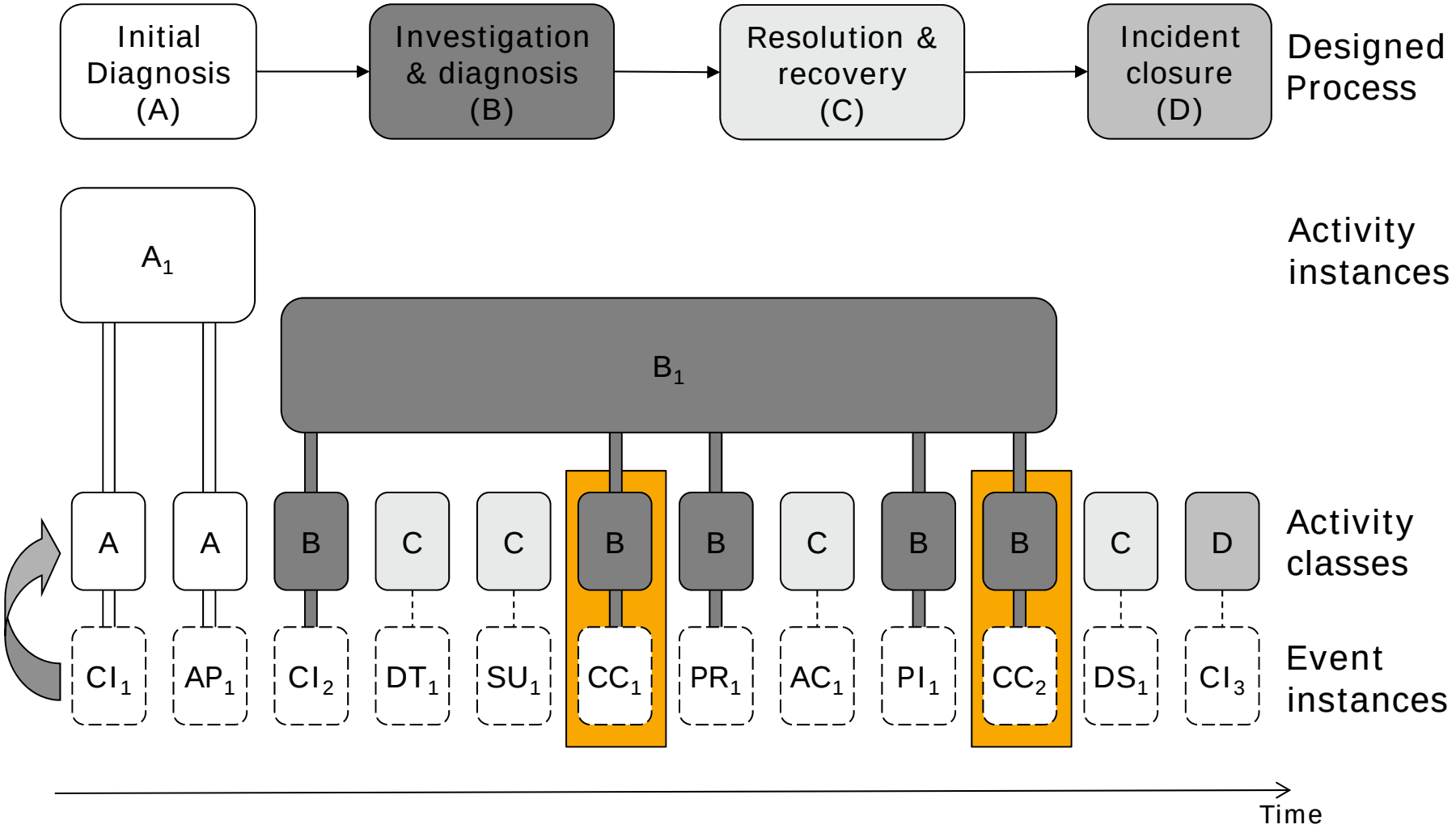


1. Sort out wrong matches
2. Add missing relations
3. Disambiguate n:m relations

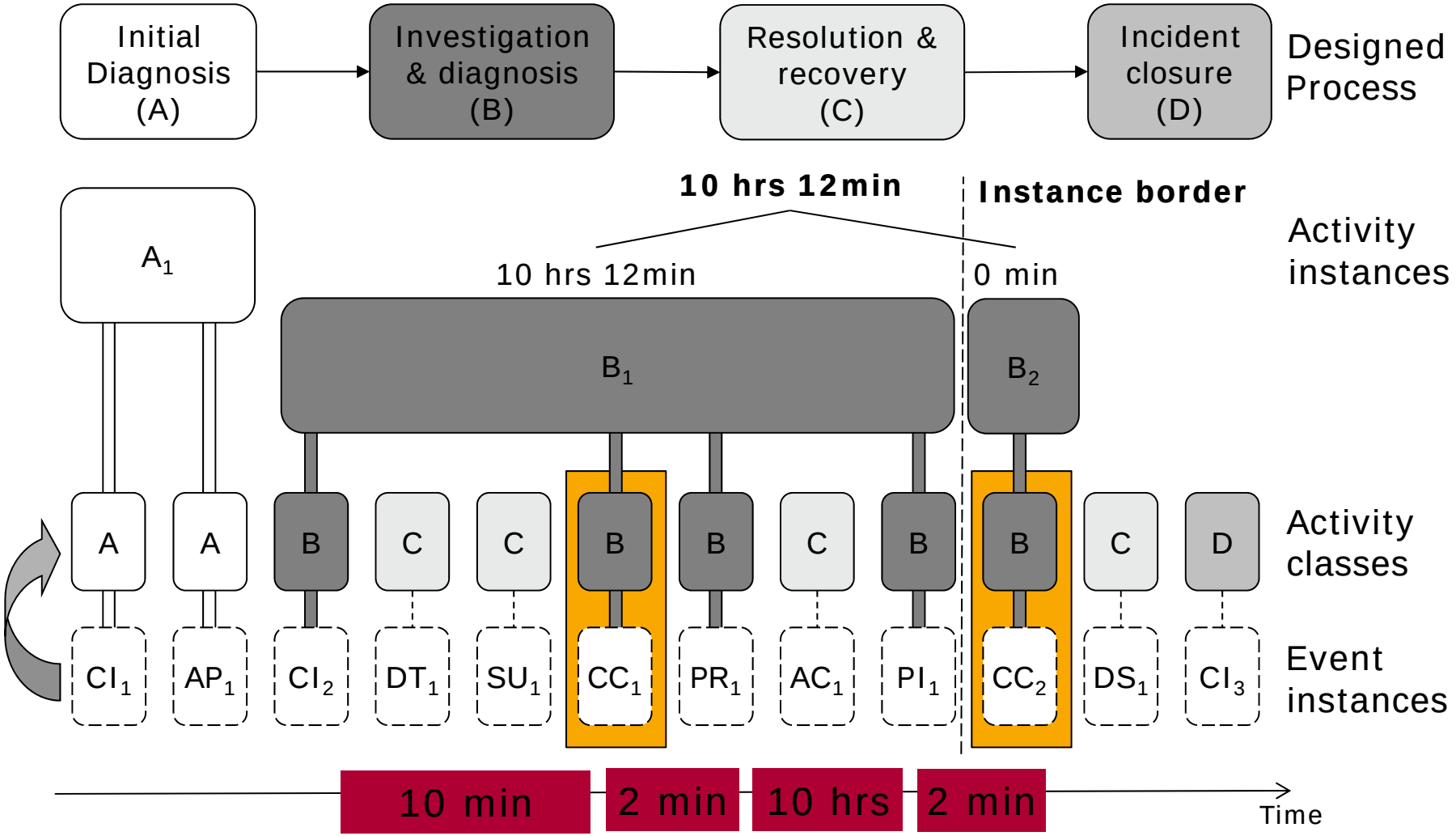
3rd step: Definition of context-sensitive mappings



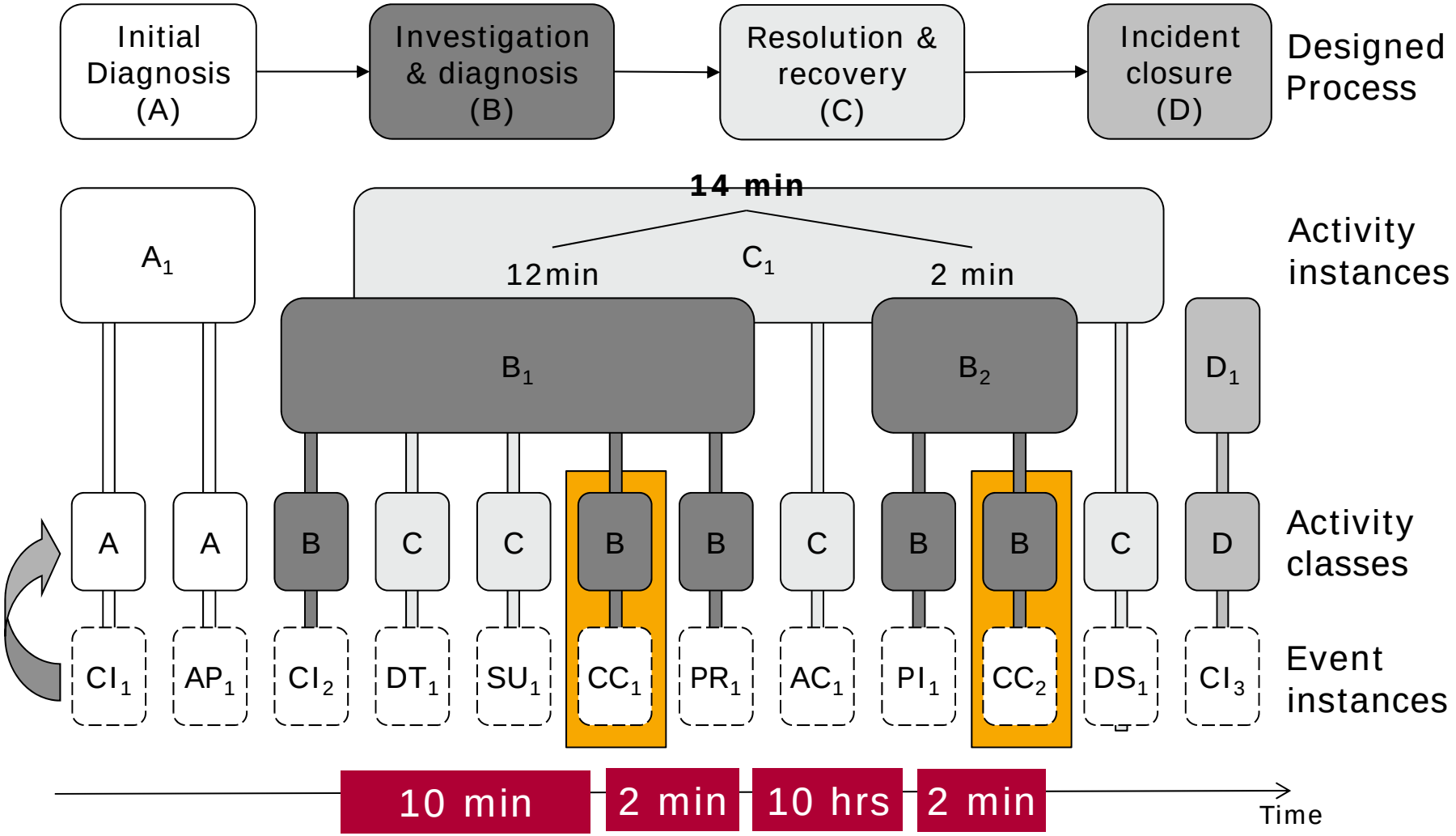
4th step: Clustering of event instances to activity instances



4th step: Clustering of event instances to activity instances



4th step: Clustering of event instances to activity instances



Evaluation on real-life incident management process

- ProM plugin implementation
- Case study with a German IT outsourcing company

➔ Comparison of our results to a manually abstracted event log

➔ Assessment of the impact on conformance analysis

Main results of the evaluation

Pros	Cons
▪ Recall of over 50 % for automated matching ▪ Effective mapping and clustering approach ▪ Impact of correct mapping & clustering on conformance	▪ Low precision (30 %) of automated matching ▪ High effort for creating context-dependent mappings

Conclusion

- Developed a novel approach for event log abstraction supporting:
  - matching of events and given activities using **annotations** and **natural language processing**
  - n:m relations between events and activities using **context-sensitive mappings**
  - concurrency of activity instances using **tree-based clustering algorithm**

Example: Mapping on type level


```
<EventMapping id="ci_1">
   <task>Initial diagnosis</task>
   <eventName>Change CI</eventName>
   <metaDataCondition>
      <name>org:role</name>
      <value>UHD</value>
      <type>equals</type>
   </metaDataCondition>
</EventMapping>

<EventMapping id="ci_2">
   <task>Incident closure</task>
   <eventName>Change CI</eventName>
   <eventCondition location="BeforeEvent">
      <![CDATA[
         <>(activity == "Resolution and recovery")
      ]]>
   </eventCondition>
</EventMapping>
```

Activity name

Event class

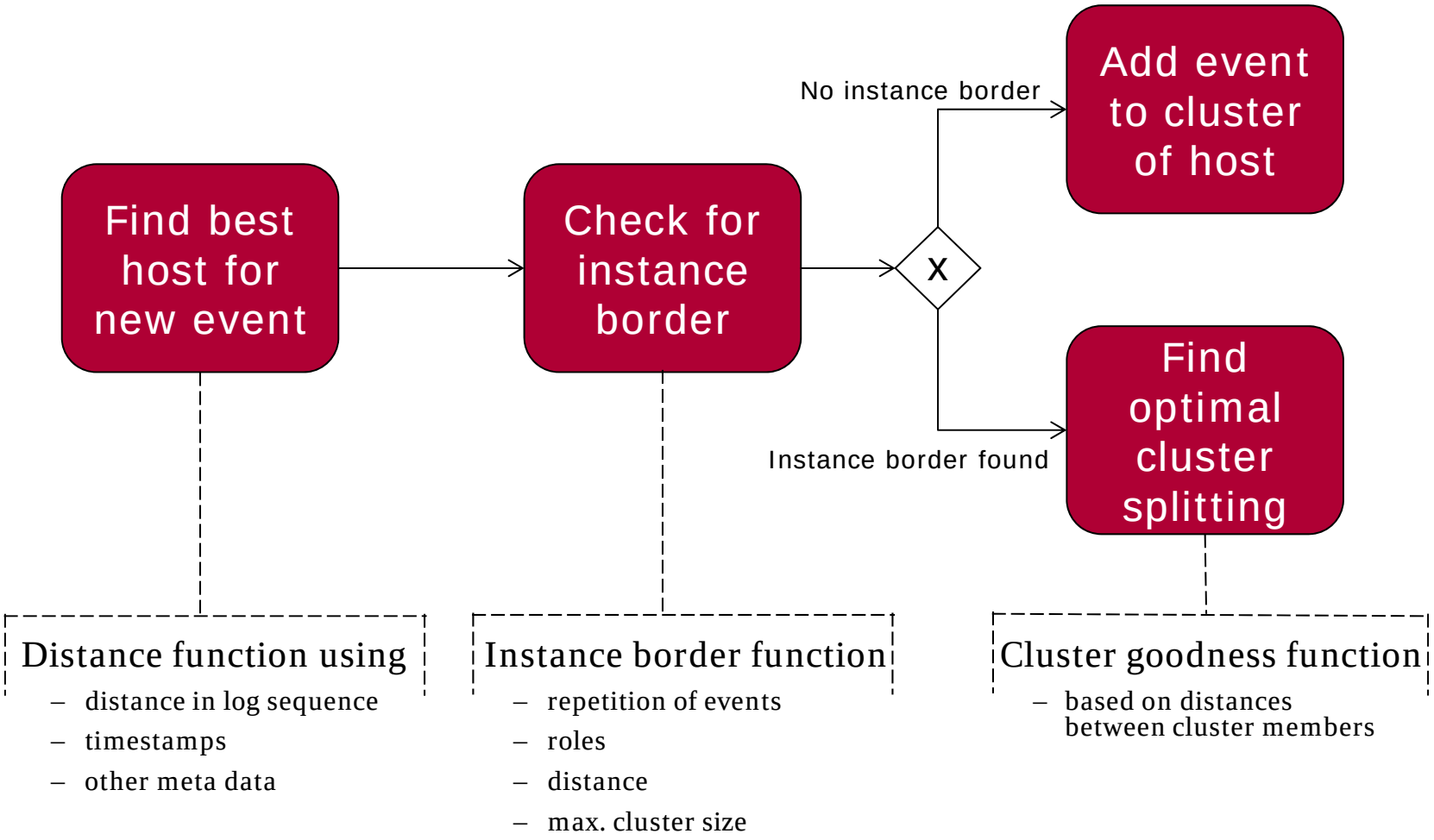
Meta data condition

Activity name

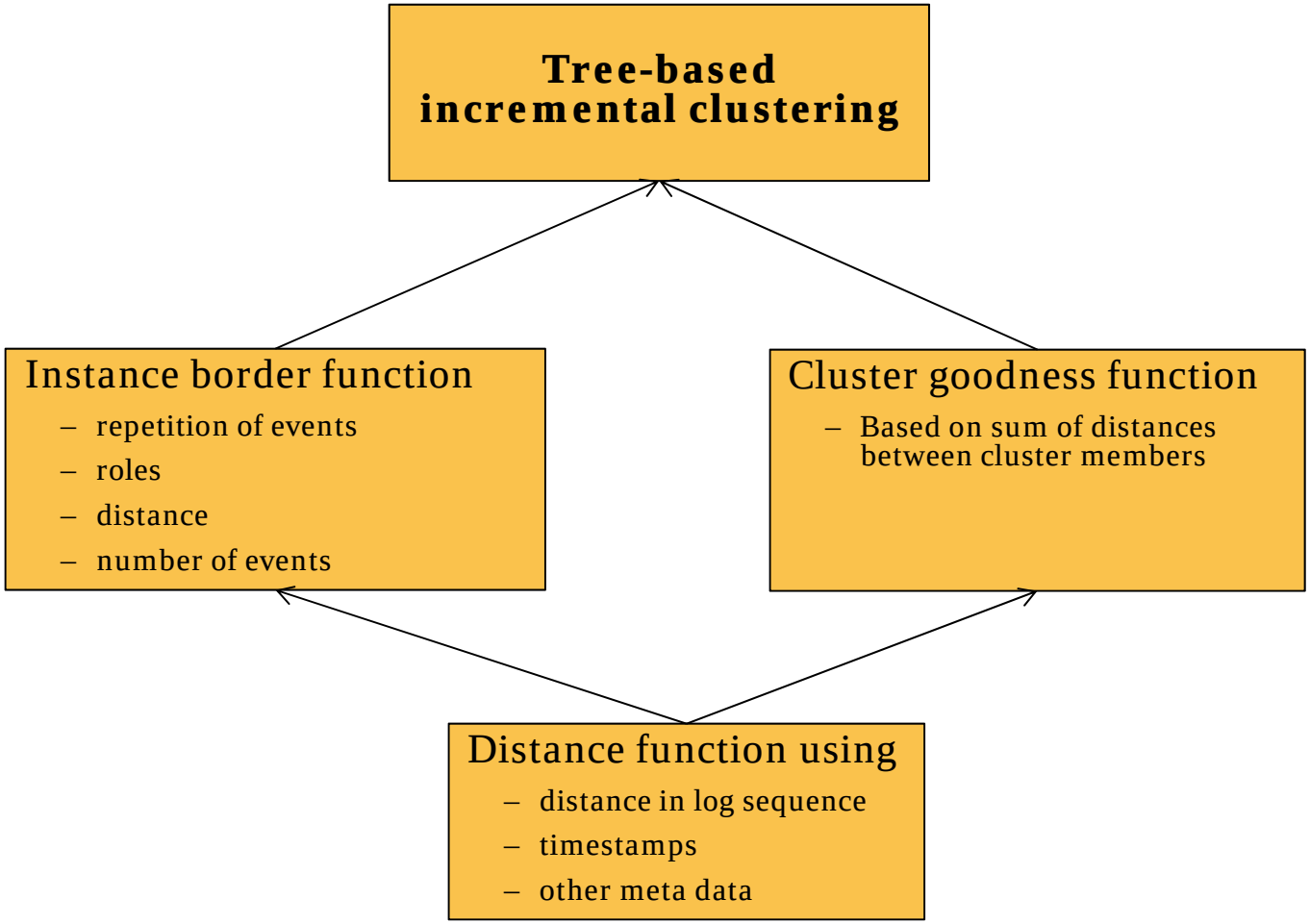
Event class

Context condition

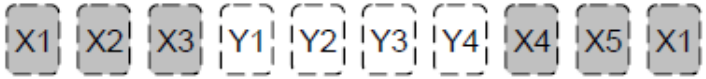
Basic 4 step instance clustering

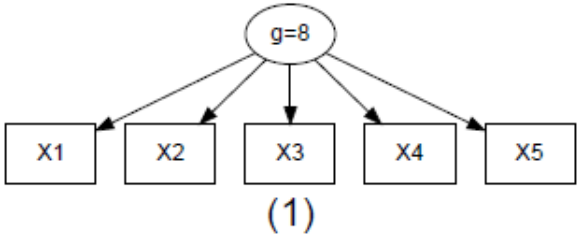


Step 2: Clustering activity instances Algorithm

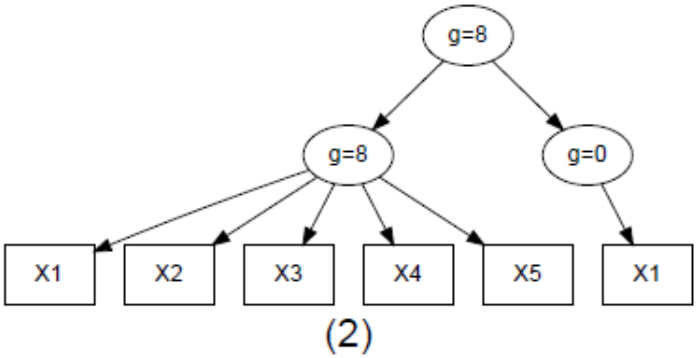


Instance clustering example

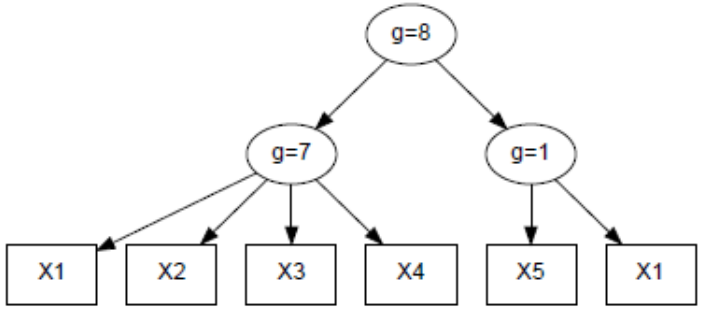




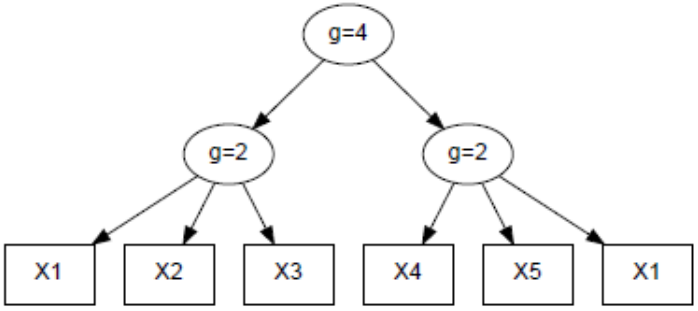
(1)



(2)

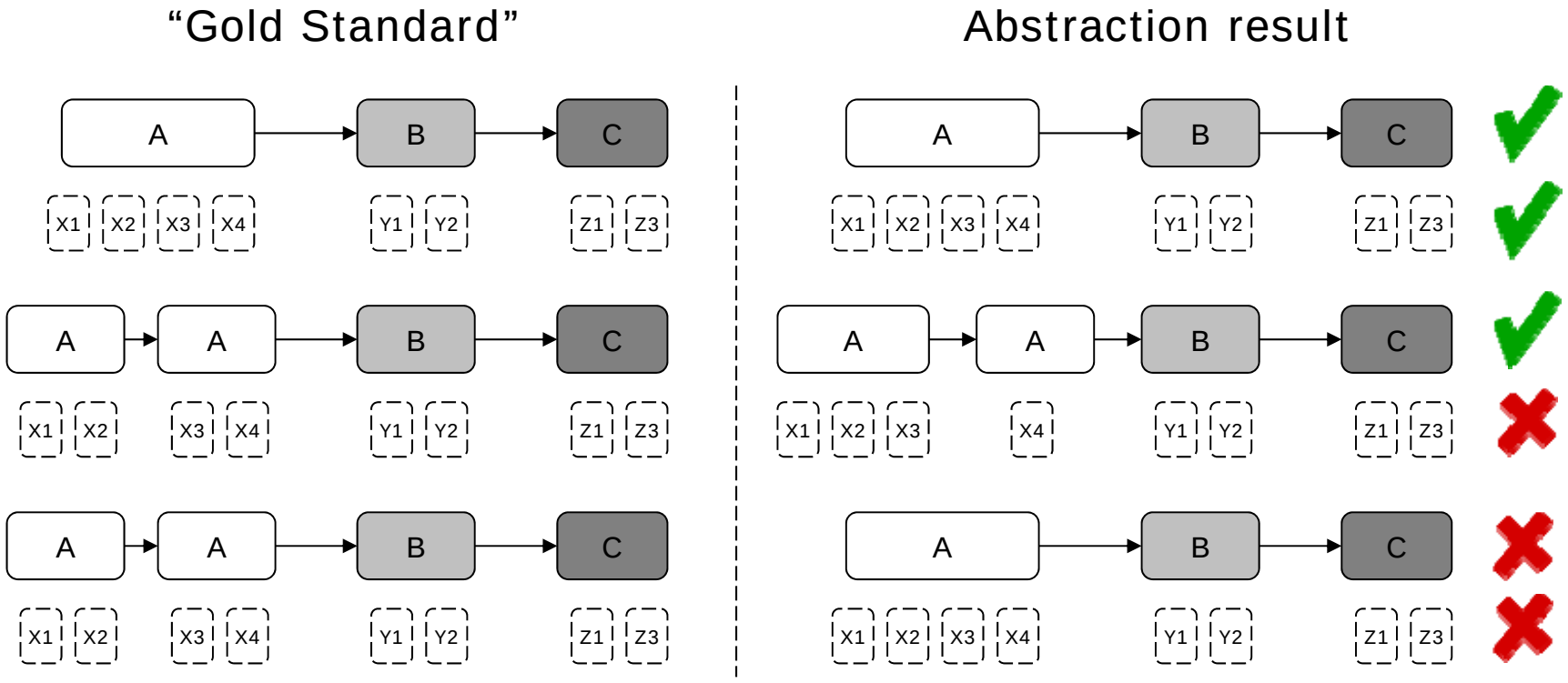


(3)



(4)

Evaluation of instance clustering



Influences of different instance borders