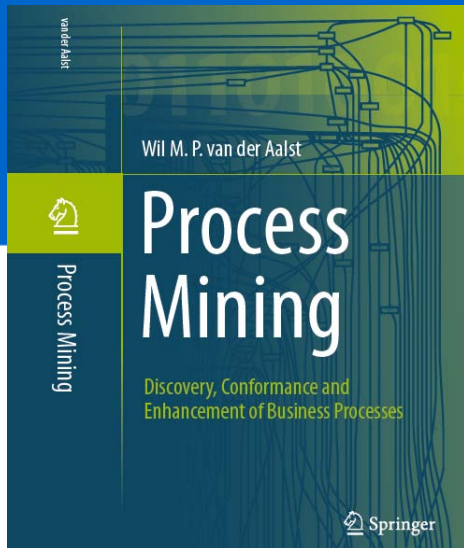


Chapter 9

Operational Support

prof.dr.ir. Wil van der Aalst
www.processmining.org



TU/e Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Overview

Chapter 1
Introduction

Part I: Preliminaries

Chapter 2
Process Modeling and
Analysis

Chapter 3
Data Mining

Part II: From Event Logs to Process Models

Chapter 4
Getting the Data

Chapter 5
Process Discovery: An
Introduction

Chapter 6
Advanced Process
Discovery Techniques

Part III: Beyond Process Discovery

Chapter 7
Conformance
Checking

Chapter 8
Mining Additional
Perspectives

Chapter 9
Operational Support

Part IV: Putting Process Mining to Work

Chapter 10
Tool Support

Chapter 11
Analyzing “Lasagna
Processes”

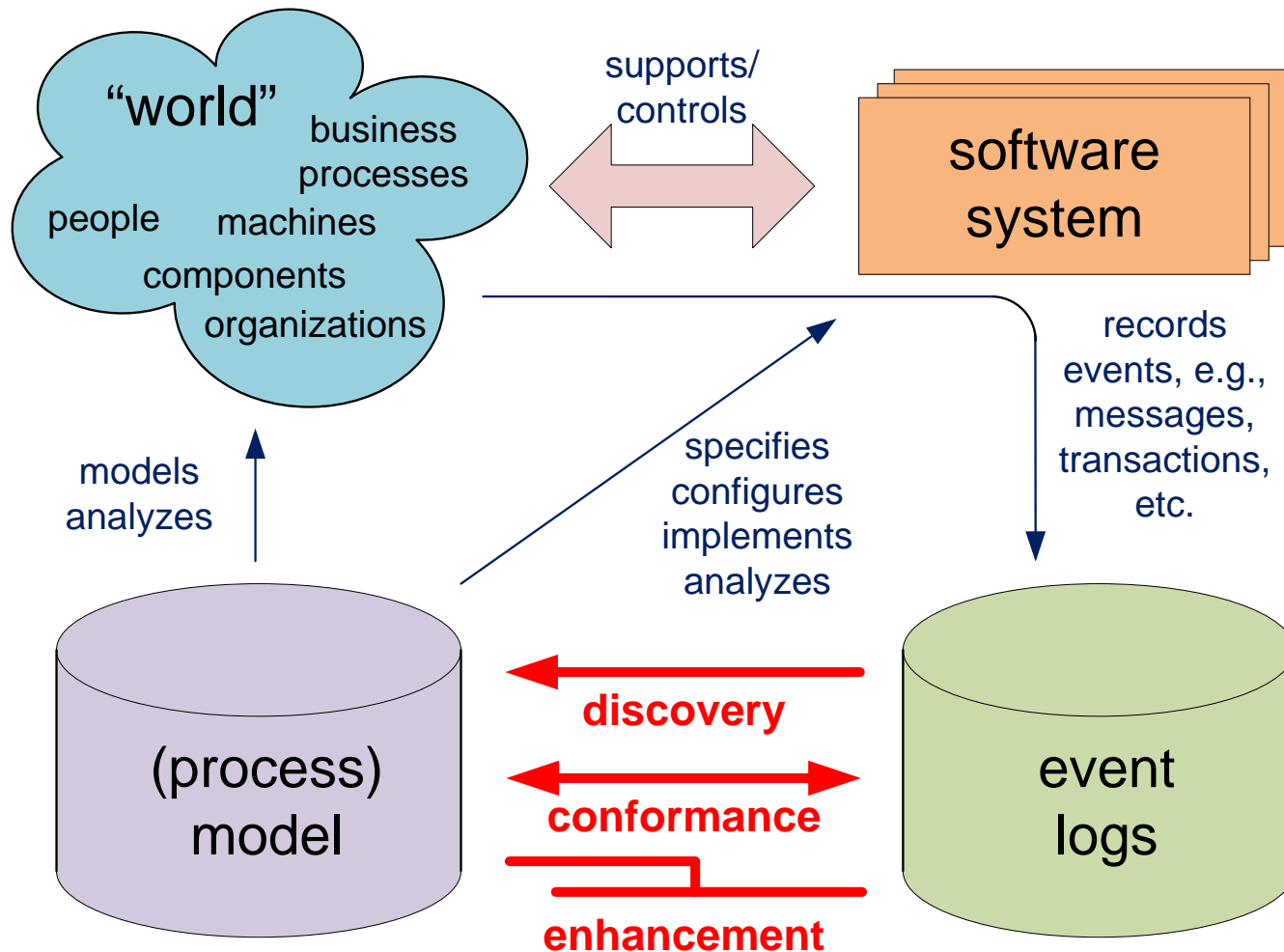
Chapter 12
Analyzing “Spaghetti
Processes”

Part V: Reflection

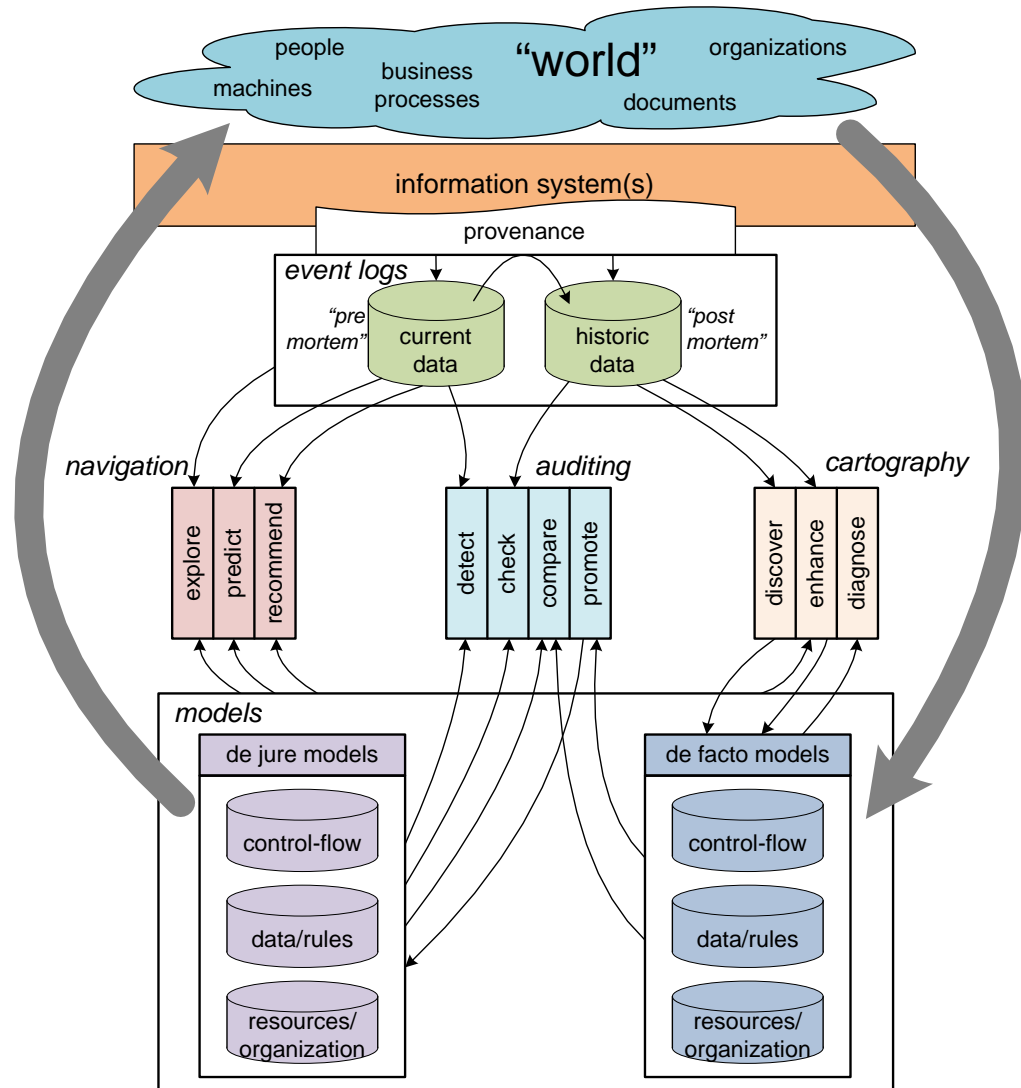
Chapter 13
Cartography and
Navigation

Chapter 14
Epilogue

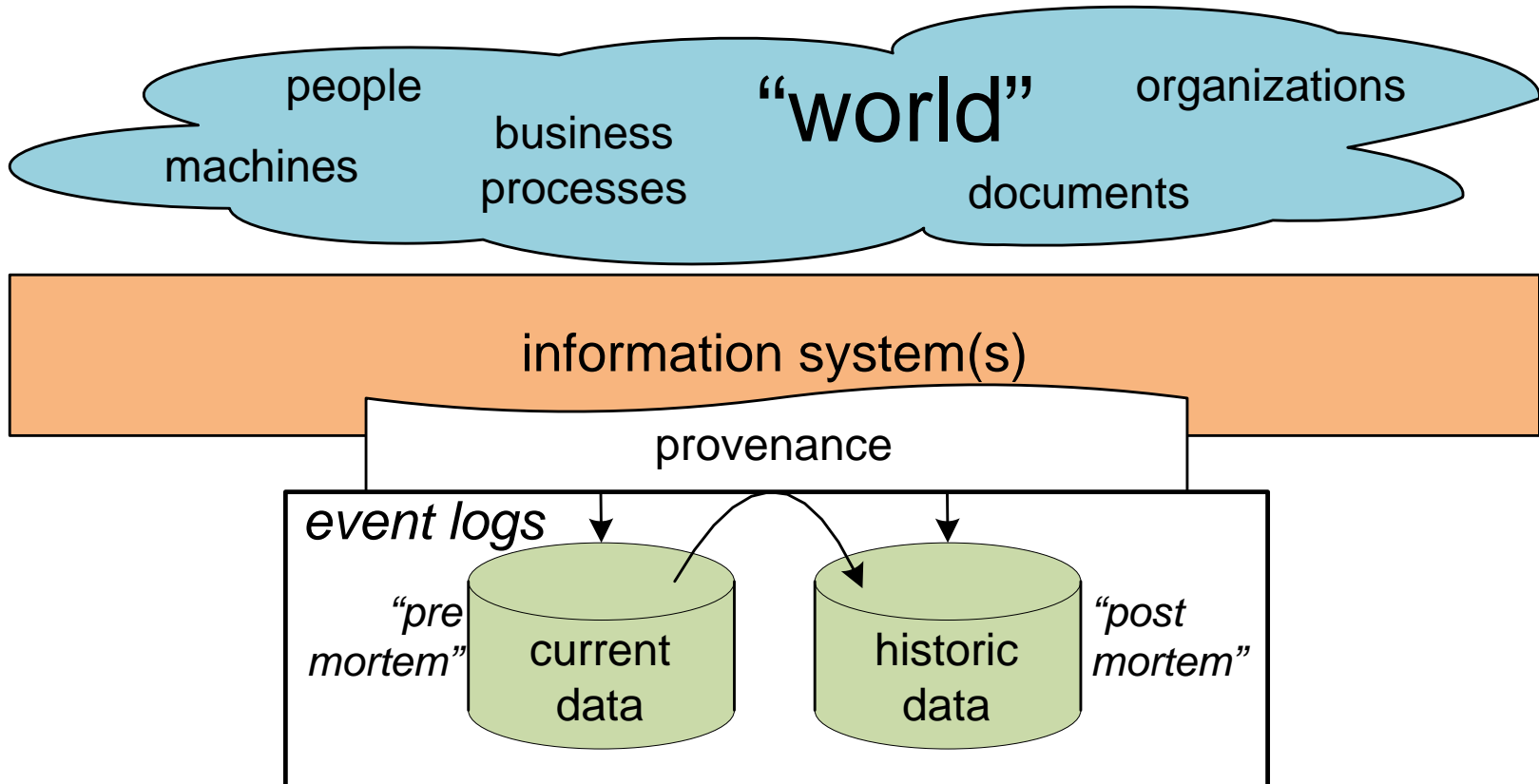
Process mining spectrum



Refined process mining framework



Business process provenance



Two types of event data: post and pre mortem

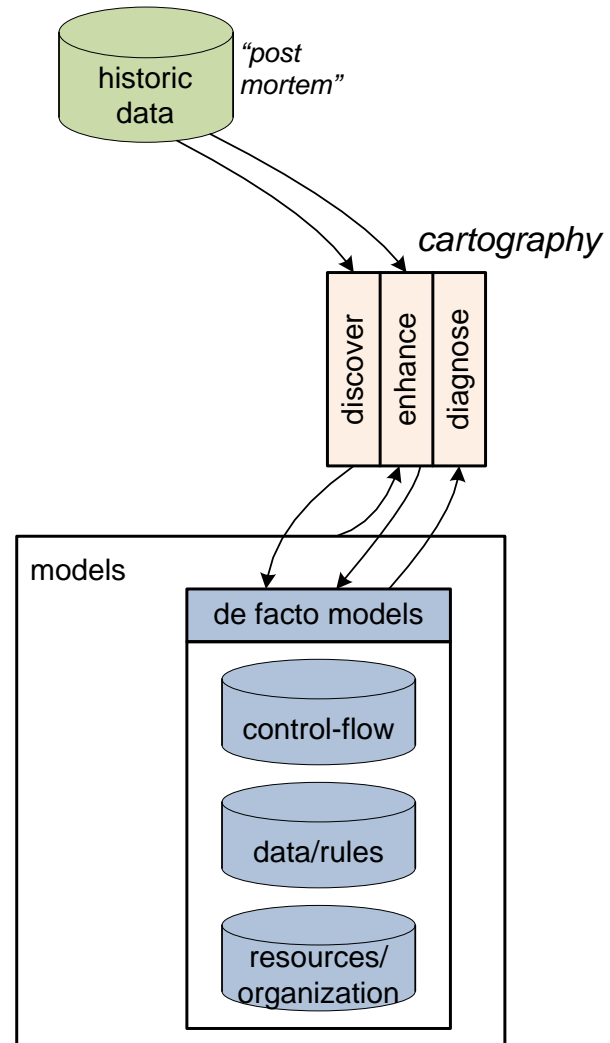
- **“Post mortem” event data** refer to information about cases that have completed, i.e., these data can be used for process improvement and auditing, but **not for influencing** the cases they refer to.
- **“Pre mortem” event data** refer to cases that have not yet completed. If a case is still running, i.e., the case is still “alive” (pre mortem), then it may be possible that information in the event log about this case (i.e., current data) **can be exploited** to ensure the correct or efficient handling of this case.

Two types of models: “de jure models” and “de facto models”

- A **de jure model** is **normative**, i.e., it specifies how things should be done or handled. For example, a process model used to configure a BPM system is normative and forces people to work in a particular way.
- A **de facto model** is **descriptive** and its goal is not to steer or control reality. Instead, de facto models aim to capture reality.

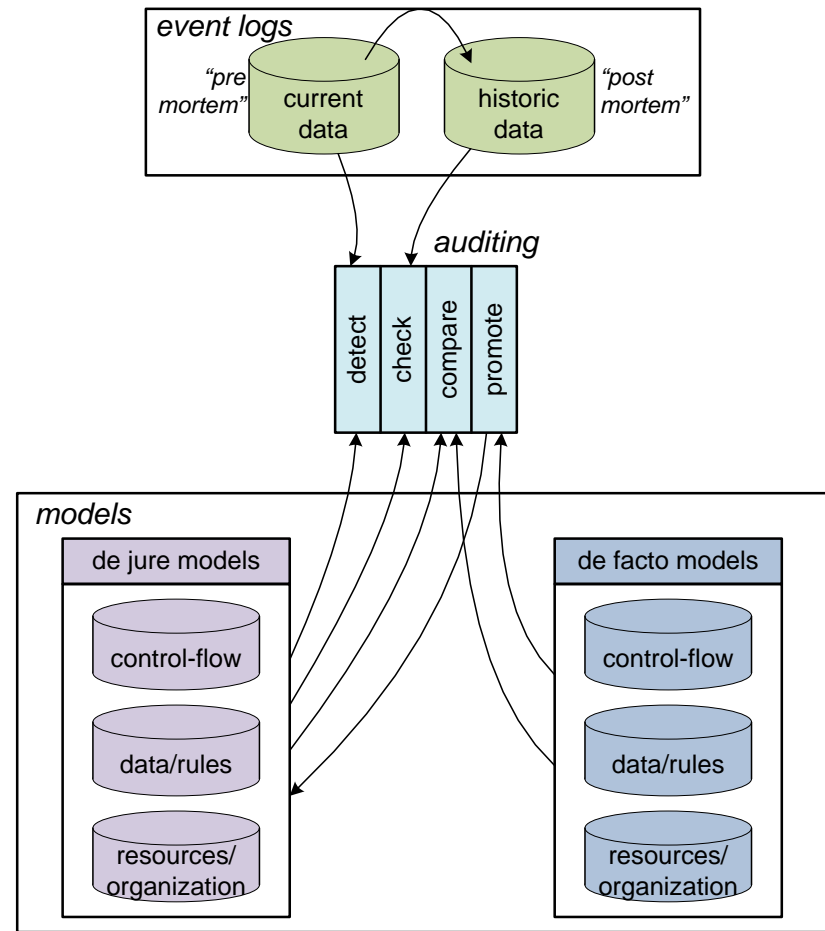
Cartography

- **Discover.** This activity is concerned with the extraction of (process) models.
- **Enhance.** When existing process models (either discovered or hand-made) can be related to events logs, it is possible to enhance these models.
- **Diagnose.** This activity does not directly use event logs and focuses on classical model-based analysis.



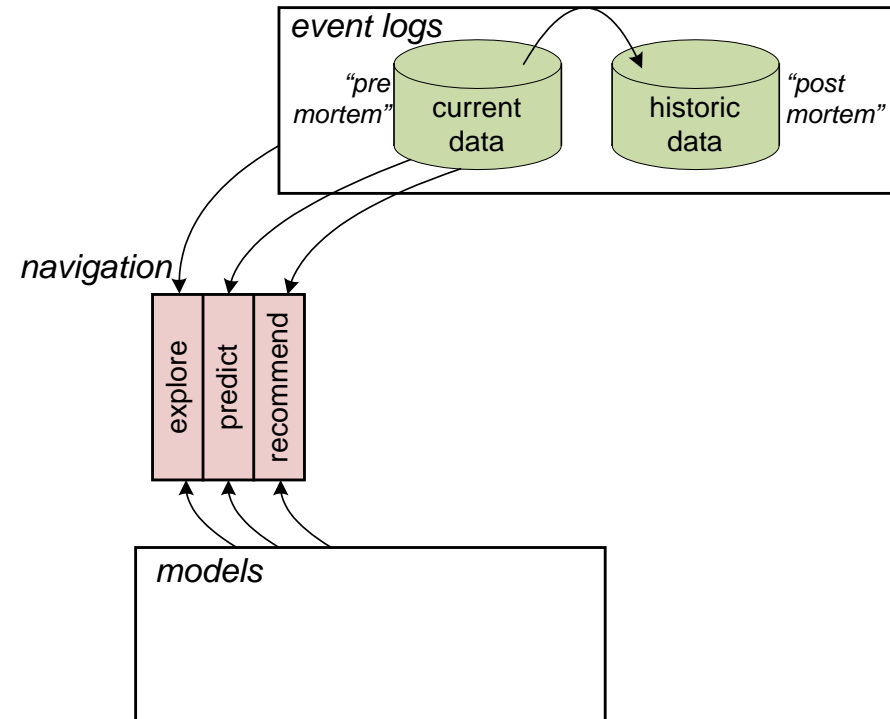
Auditing

- **Detect.** Compares de jure models with current “pre mortem” data. The moment a predefined rule is violated, an alert is generated (**online**).
- **Check.** The goal of this activity is to pinpoint deviations and quantify the level of compliance (**offline**).
- **Compare.** De facto models can be compared with de jure models to see in what way reality deviates from what was planned or expected.
- **Promote.** Promote parts of the de facto model to a new de jure model.

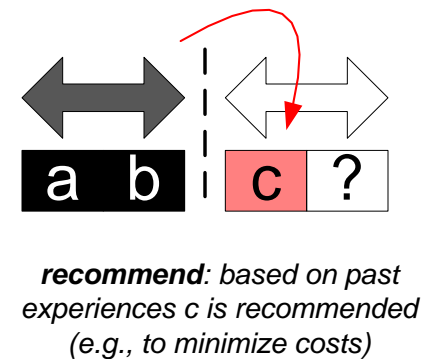
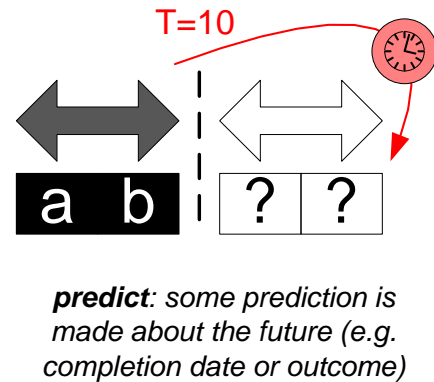
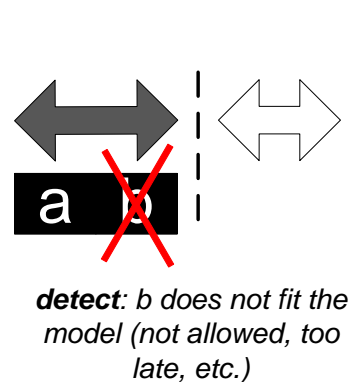
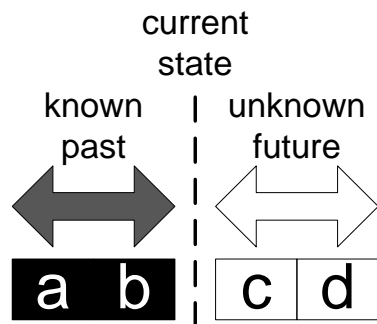


Navigation

- **Explore.** The combination of event data and models can be used to explore business processes at run-time.
- **Predict.** By combining information about running cases with models, it is possible to make predictions about the future, e.g., the remaining flow time and the probability of success.
- **Recommend.** The information used for predicting the future can also be used to recommend suitable actions (e.g. to minimize costs or time).



Operational support: online process mining using “pre mortem” event data



Let us focus one time

case id trace

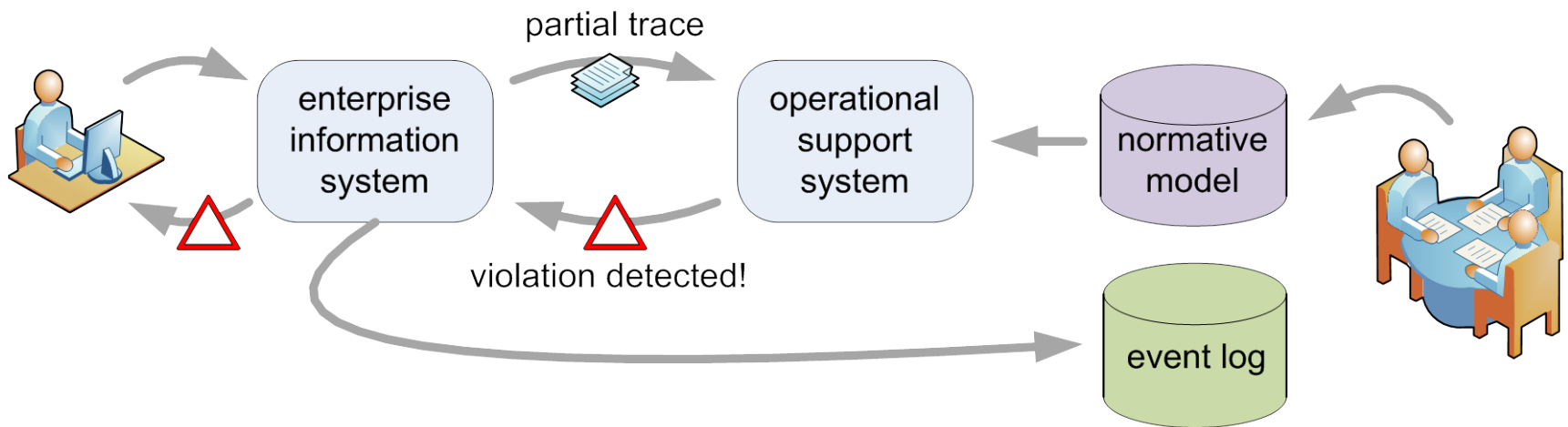
1 $\langle a_{start}^{12}, a_{complete}^{19}, b_{start}^{25}, d_{start}^{26}, b_{complete}^{32}, d_{complete}^{33}, e_{start}^{35}, e_{complete}^{40}, h_{start}^{50}, h_{complete}^{54} \rangle$

2 $\langle a_{start}^{17}, a_{complete}^{23}, d_{start}^{28}, c_{start}^{30}, d_{complete}^{32}, c_{complete}^{38}, e_{start}^{50}, e_{complete}^{59}, g_{start}^{70}, g_{complete}^{73} \rangle$

3 $\langle a_{start}^{25}, a_{complete}^{30}, c_{start}^{32}, c_{complete}^{35}, d_{start}^{35}, d_{complete}^{40}, e_{start}^{45}, e_{complete}^{50}, f_{start}^{50}, f_{complete}^{55},$
 $b_{start}^{60}, d_{start}^{62}, b_{complete}^{65}, d_{complete}^{67}, e_{start}^{80}, e_{complete}^{87}, g_{start}^{90}, g_{complete}^{98} \rangle$

... ...

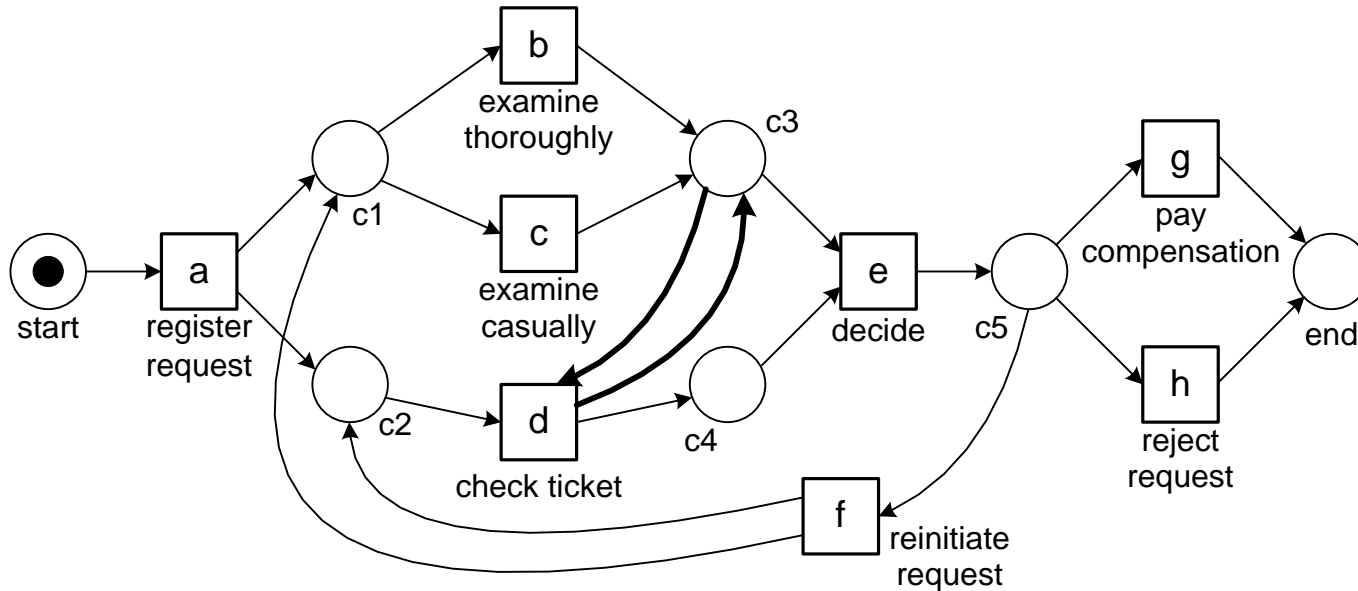
Operational support: Detect



Example

$\langle a_{start}^{12}, a_{complete}^{19}, b_{start}^{25}, d_{start}^{26} \rangle$

alert!!!!

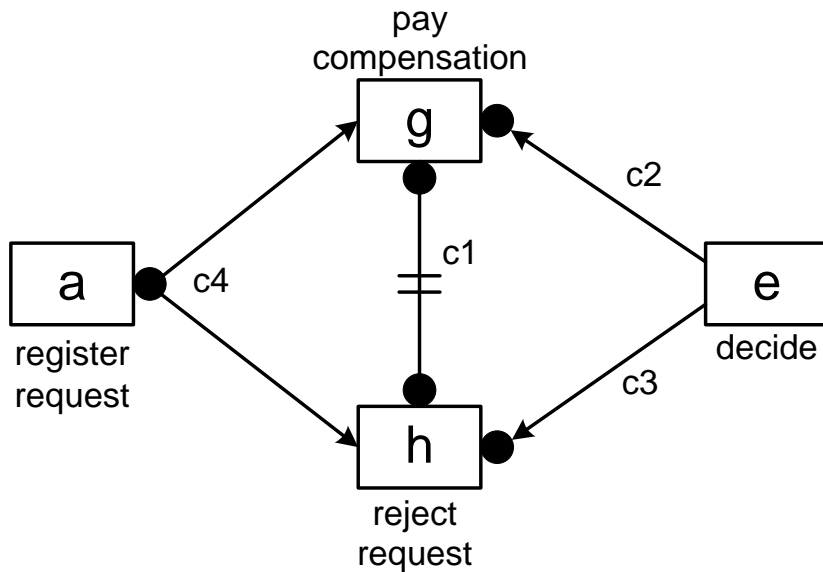


case id trace

1	$\langle a_{start}^{12}, a_{complete}^{19}, b_{start}^{25}, d_{start}^{26}, b_{complete}^{32}, d_{complete}^{33}, e_{start}^{35}, e_{complete}^{40}, h_{start}^{50}, h_{complete}^{54} \rangle$
2	$\langle a_{start}^{17}, a_{complete}^{23}, d_{start}^{28}, c_{start}^{30}, d_{complete}^{32}, c_{complete}^{38}, e_{start}^{50}, e_{complete}^{59}, g_{start}^{70}, g_{complete}^{73} \rangle$
3	$\langle a_{start}^{25}, a_{complete}^{30}, c_{start}^{32}, c_{complete}^{35}, d_{start}^{35}, d_{complete}^{40}, e_{start}^{45}, e_{complete}^{50}, f_{start}^{50}, f_{complete}^{55}, b_{start}^{60}, d_{start}^{62}, b_{complete}^{65}, d_{complete}^{67}, e_{start}^{80}, e_{complete}^{87}, g_{start}^{90}, g_{complete}^{98} \rangle$

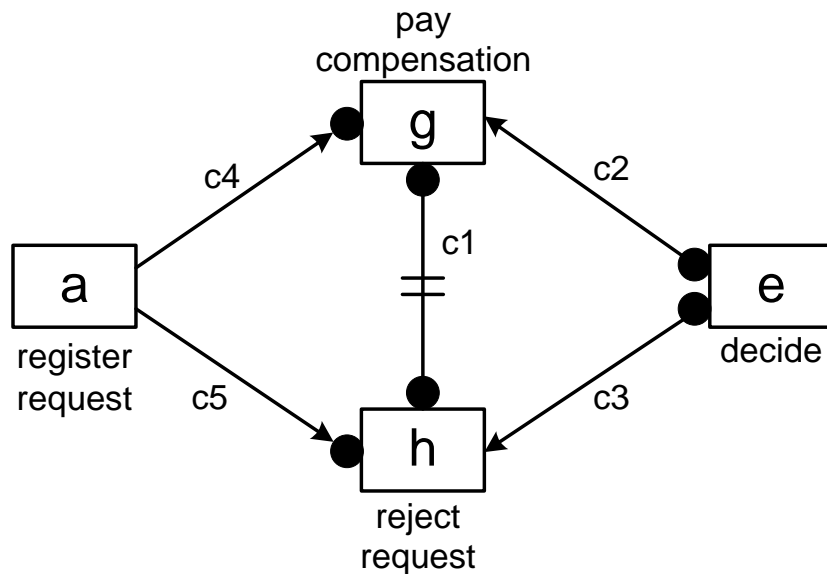
... ..

Declare specifications for detecting violations



- **Satisfied:** the LTL formula evaluates to true for the current partial trace.
- **Temporarily violated:** the LTL formula evaluates to false, however, there is a longer trace that evaluates to true.
- **Permanently violated:** the LTL formula evaluates to false for current trace and all its extensions

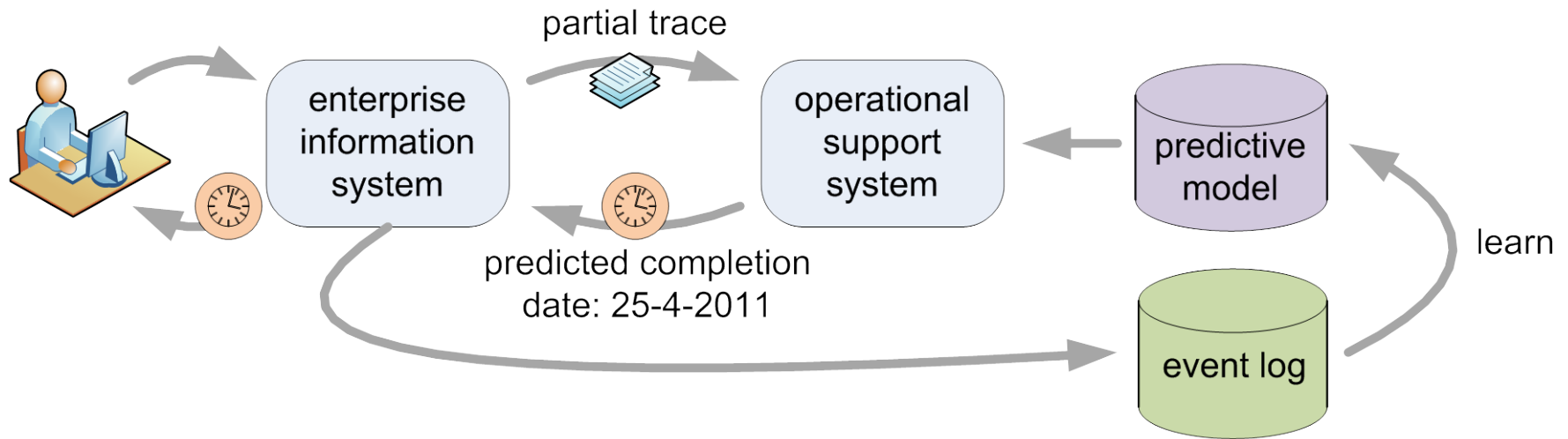
Conflicting constraints



- A Declare specification is **satisfied** for a case if all of its constraints are satisfied.
- A Declare specification is **temporarily violated** by a case if for the current partial trace at least one of the constraints is violated, however, there is a possible future in which all constraints are satisfied.
- A Declare specification is **permanently violated** by a case if no such future exists.

Note that c1, c2, and c3 imply that e cannot be executed without permanently violating the specification.

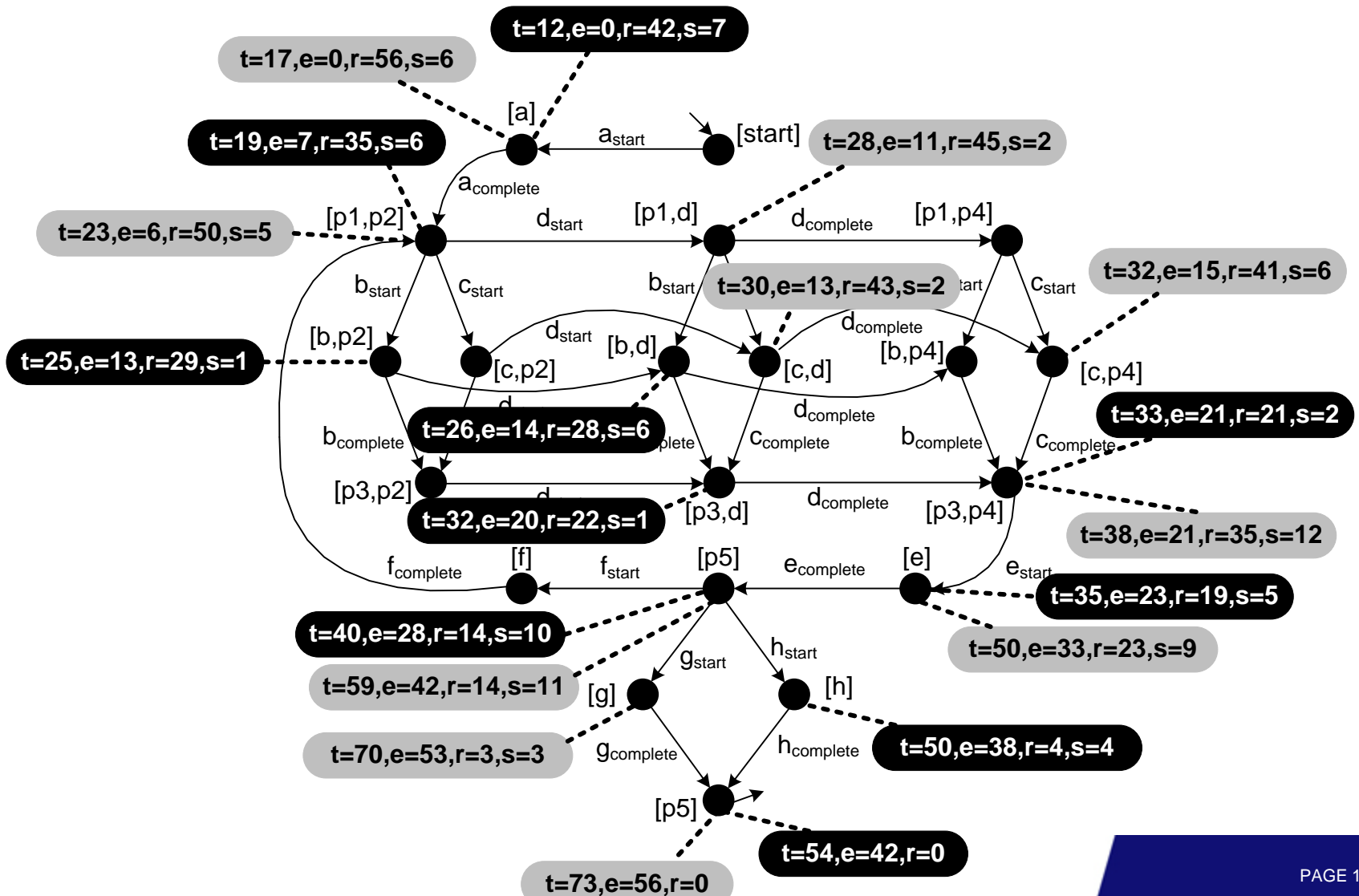
Operational support: Predict



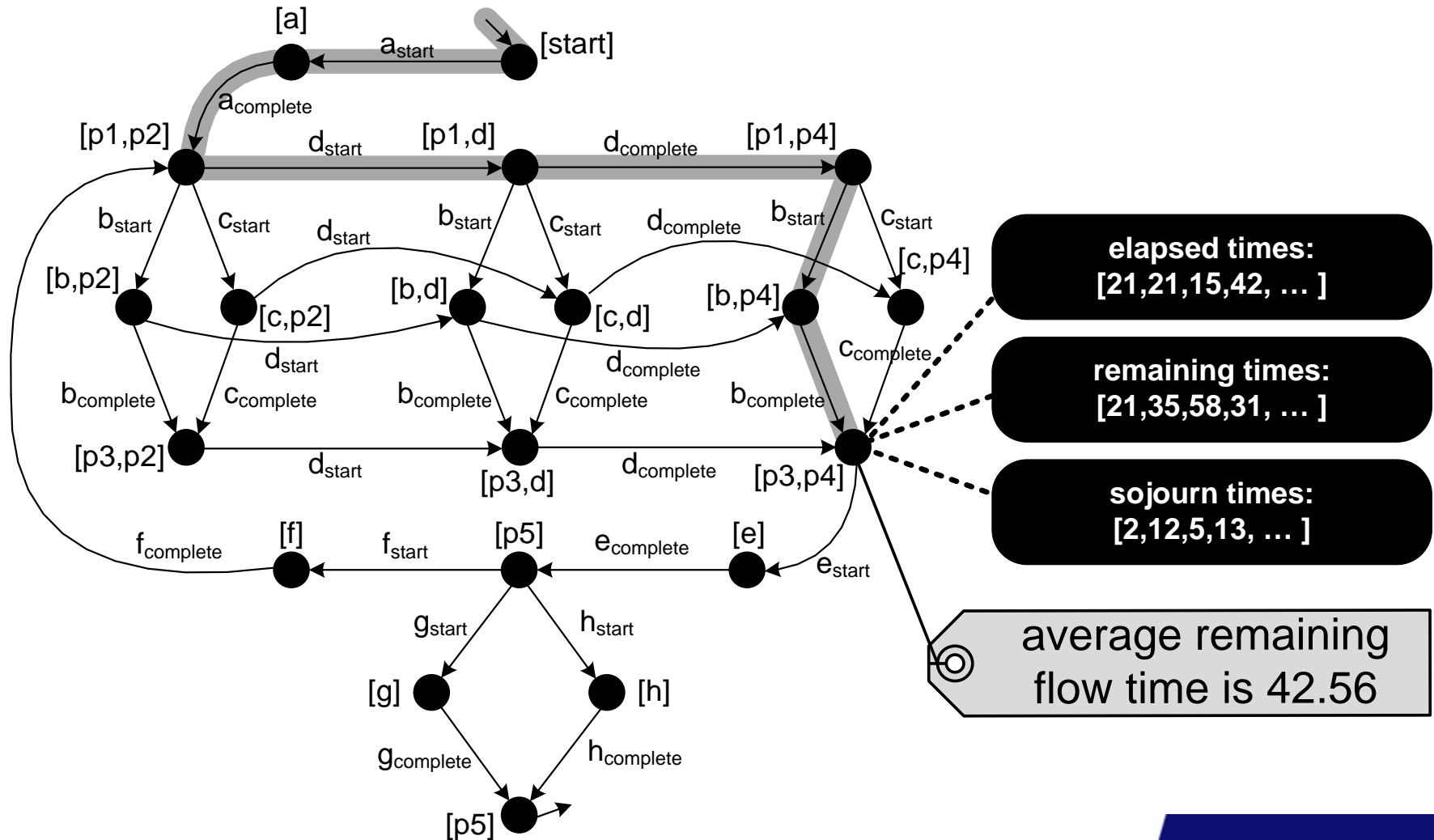
Examples of predictions

- the predicted remaining flow time is 14 days;
- the predicted probability of meeting the legal deadline is 0.72;
- the predicted total cost of this case is 4500 euro;
- the predicted probability that activity a will occur is 0.34;
- the predicted probability that person r will work on this case is 0.57;
- the predicted probability that a case will be rejected is 0.67; and
- the predicted total service time is 98 minutes.

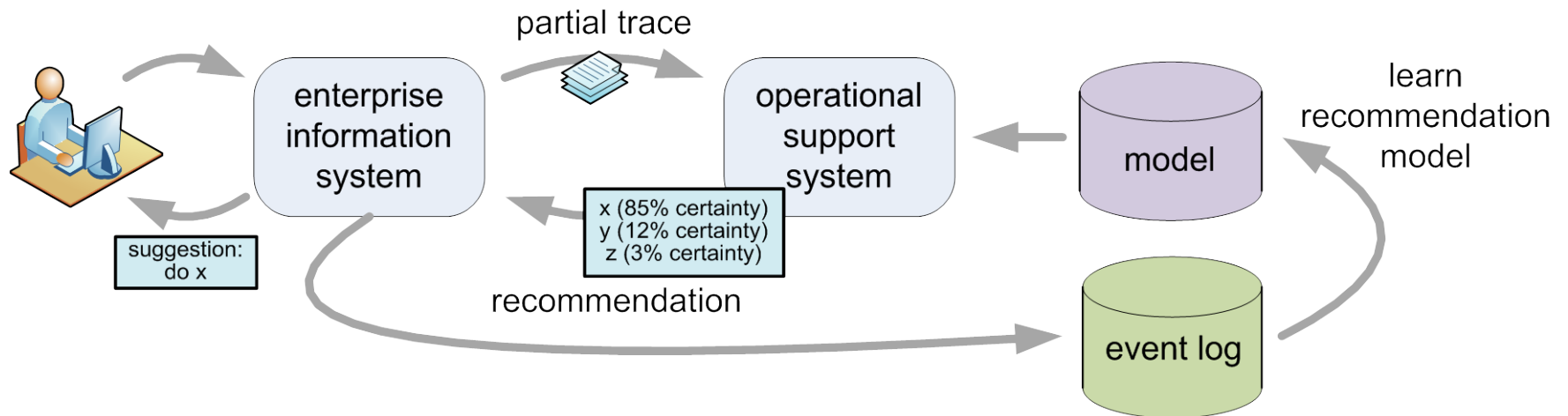
Annotated transition system



Collect results per state



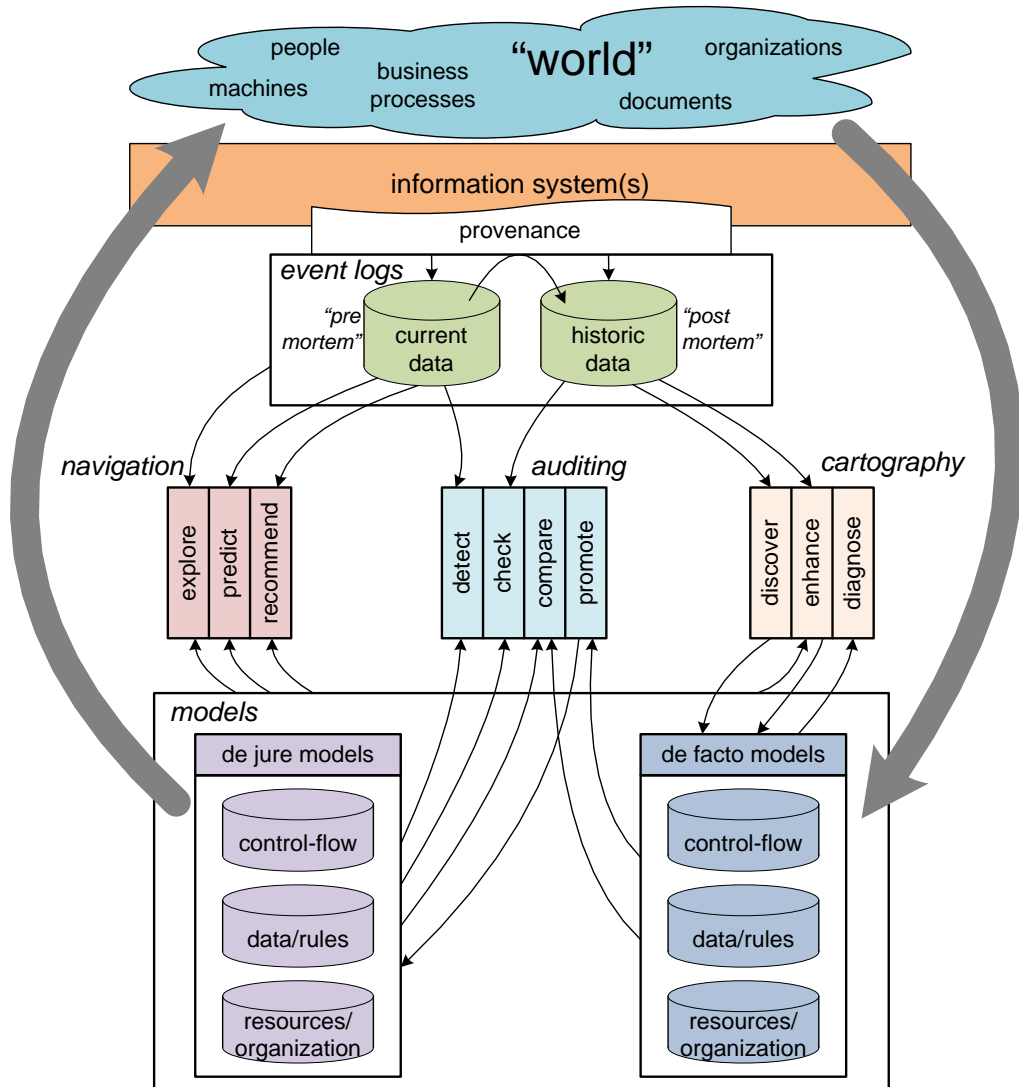
Operational support: Recommend



Recommend

- **Possible recommendations:**
 - next activity;
 - suitable resource; or
 - routing decision.
- **A recommendation is always given with respect to a specific goal.**
- **Examples of goals are:**
 - minimize the remaining flow time;
 - minimize the total costs;
 - maximize the fraction of cases handled within 4 weeks;
 - maximize the fraction of cases that is accepted; and
 - minimize resource usage.

Process mining spectrum



Chapter 1
Introduction

Part I: Preliminaries

Chapter 2
Process Modeling and Analysis

Chapter 3
Data Mining

Part II: From Event Logs to Process Models

Chapter 4
Getting the Data

Chapter 5
Process Discovery: An Introduction

Chapter 6
Advanced Process Discovery Techniques

Part III: Beyond Process Discovery

Chapter 7
Conformance Checking

Chapter 8
Mining Additional Perspectives

Chapter 9
Operational Support

Part IV: Putting Process Mining to Work

Chapter 10
Tool Support

Chapter 11
Analyzing "Lasagna Processes"

Chapter 12
Analyzing "Spaghetti Processes"

Part V: Reflection

Chapter 13
Cartography and Navigation

Chapter 14
Epilogue