From Low-Level Events to Activities

A Pattern-based Approach

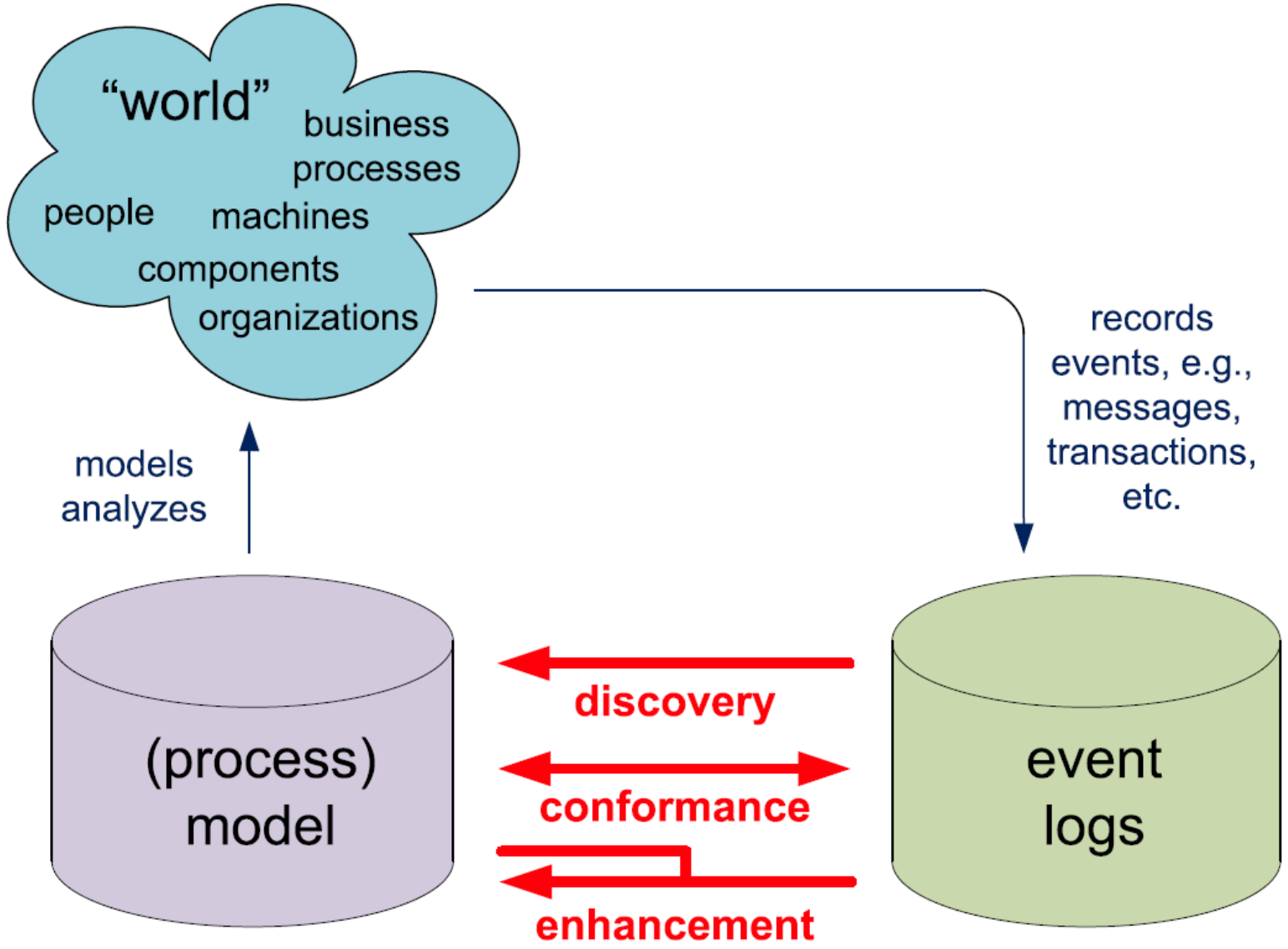
Felix Mannhardt, Massimiliano de Leoni, Hajo A. Reijers,
Wil M.P. van der Aalst, Pieter J. Toussaint (NTNU, Trondheim)

Paper: 10.1007/978-3-319-45348-4_8

TU/e Technische Universiteit Eindhoven University of Technology

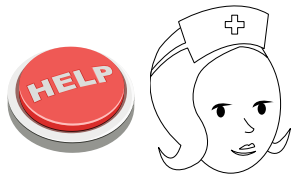
Where innovation starts

Scope: Analysis of event data



Problem: Events ≠ Recognizable activities

Alarm



records

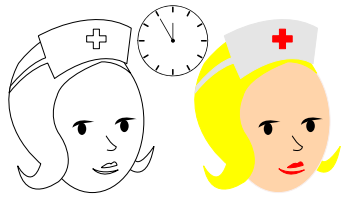
Event	Time
Red Button	20:08:00
Green Button	20:10:00
Gray Button	20:16:00

Visit

records

Event	Time
Green Button	22:02:00
Gray Button	22:10:00

Handover



records

Event	Time
Nurse Changed	07:02:00
Green Button	07:15:00
Gray Button	07:18:00

Goal 1: From low-level to high-level events (Supervised)

Low-level Event	Time
Red Button	20:08:00
Green Button	20:10:00
Gray Button	20:16:00
Green Button	22:02:00
Gray Button	22:10:00
Nurse Changed	07:02:00
Green Button	07:15:00
Gray Button	07:18:00

High-level Event	Start	Complete
Alarm	20:08:00	20:16:00
Visit	22:02:00	22:10:00
Shift Change	07:02:00	07:18:00

Goal 2: Deal with shared labels, concurrency and noise

Low-level Event	Time
Red Button	20:08:00
Green Button	20:10:00
~~**Gray Button**~~	~~**20:16:00**~~
Green Button	22:02:00
Nurse Changed	**22:09:00**
Gray Button	22:10:00
Nurse Changed	**22:15:00**
Nurse Changed	07:02:00
Green Button	07:15:00
Gray Button	07:18:00

Missing events

Unexpected events

High-level Event	Start	Complete
Alarm	20:08:00	**20:10:00**
Visit	22:02:00	22:10:00
Handover	07:02:00	07:18:00

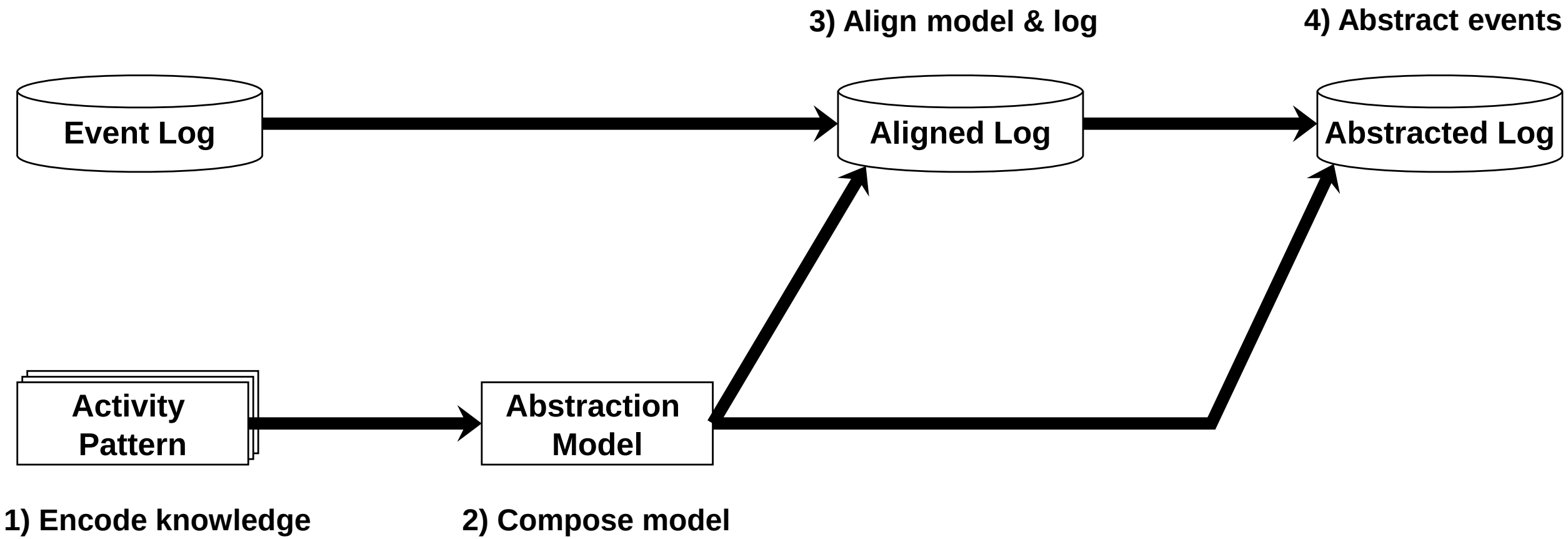
Related work

- Unsupervised event abstraction (Ferreira et al., Folino et al., …)
  - Does not take domain knowledge into account

- Supervised event abstraction (Thomas Baier et al., …)
  - Assumes knowledge of a complete process model
  - Semi-automatic discovery of the mapping between events and activities
  - Uses clustering and constraint satisfaction to determine the mapping

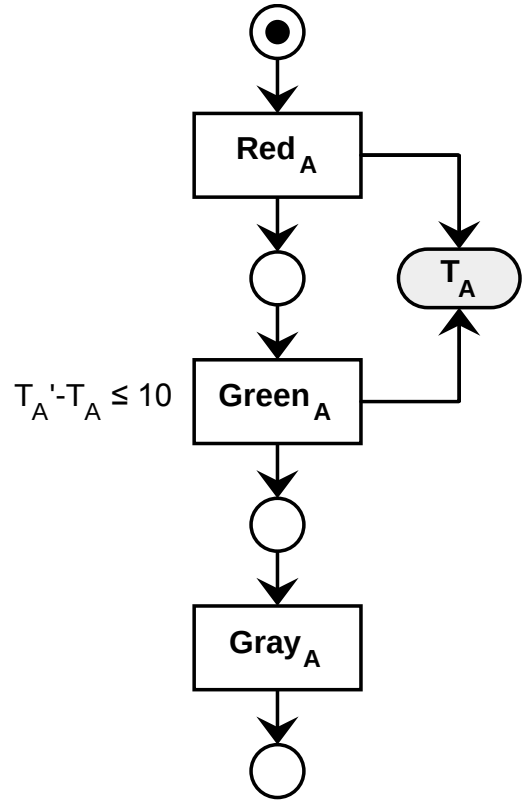
- Complex event processing
  - Focus on detection of event patterns in data streams
  - No concept of process instance / trace

Overview: From Low-level Events to Activities

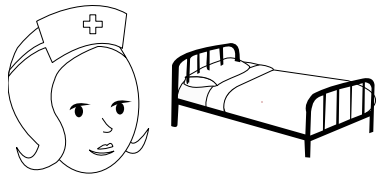


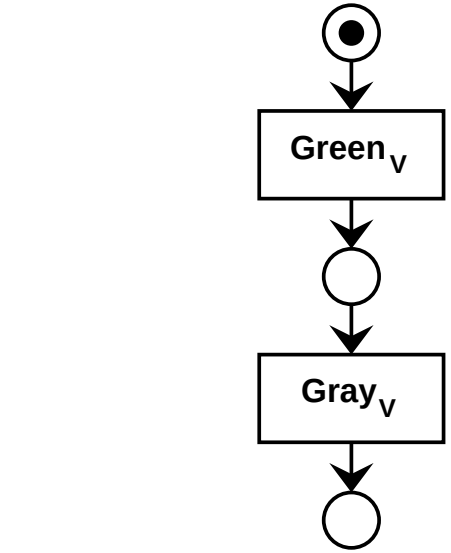
1) Encode knowledge on activities as activity patterns

Alarm



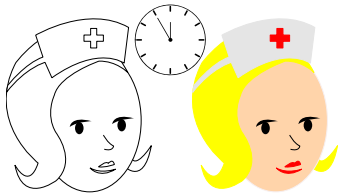
Visit



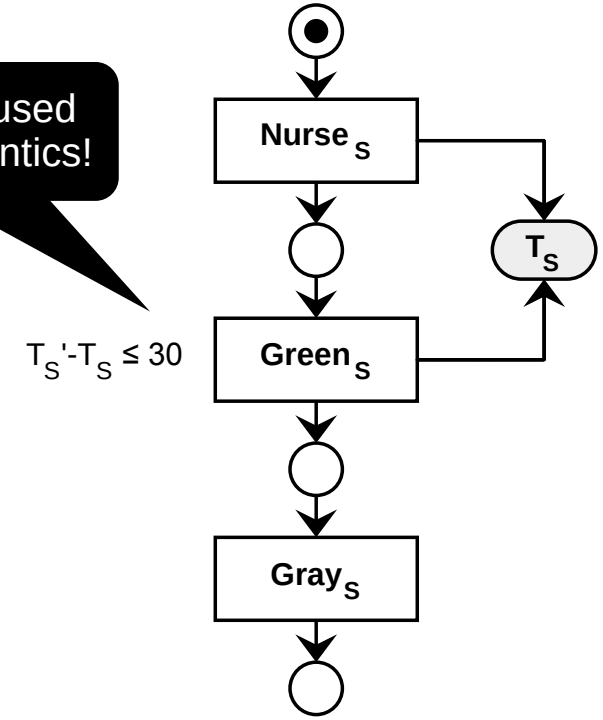


Pattern defines traces expected for one activity instance!

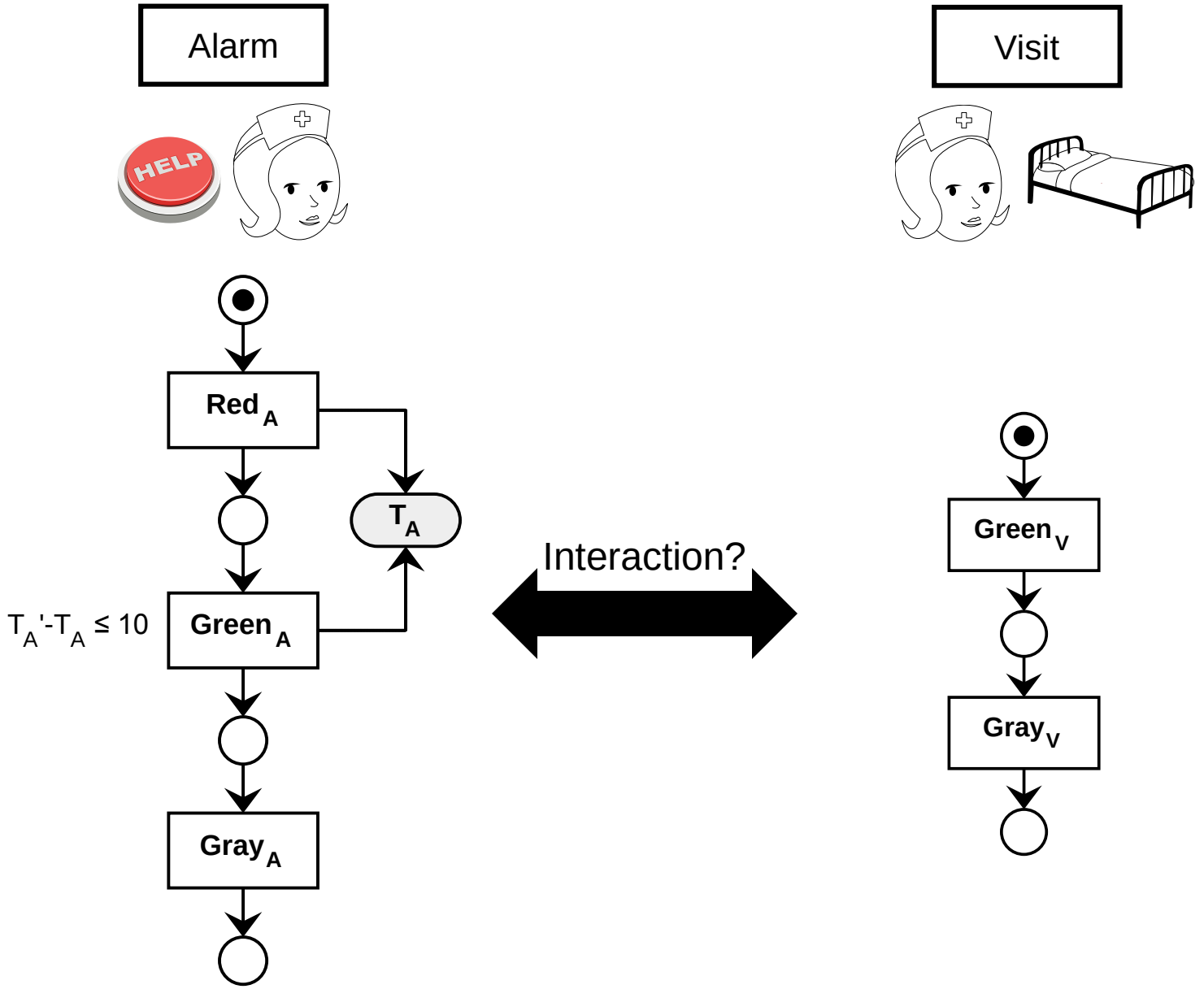
Handover



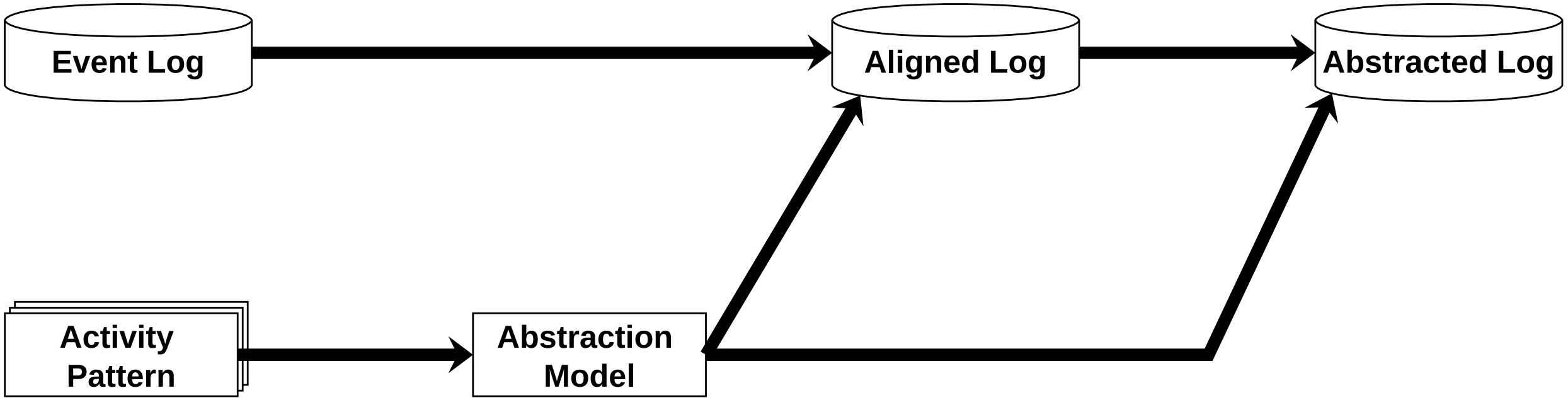
Data Petri net used for clear semantics!



What about interaction between activity patterns?

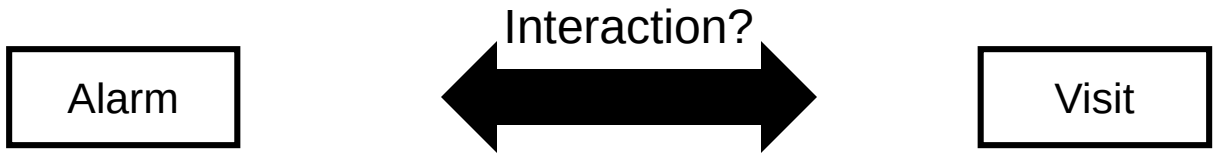


From Low-level Events to Activities

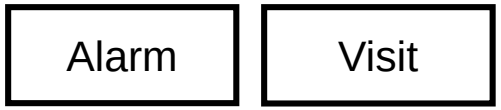


2) Compose model

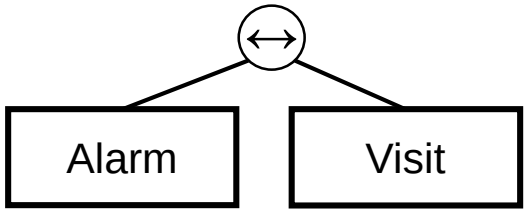
2) Build an integrated abstraction model



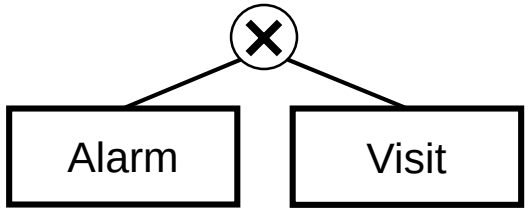
Parallel



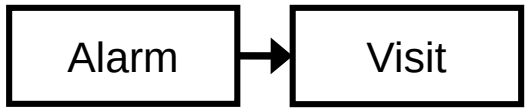
Interleaving



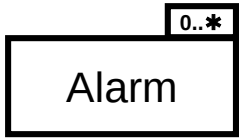
Choice



Sequence

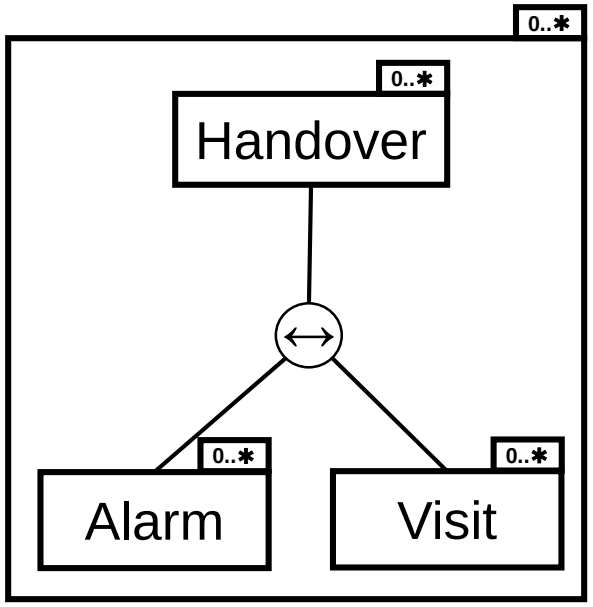


Repetition



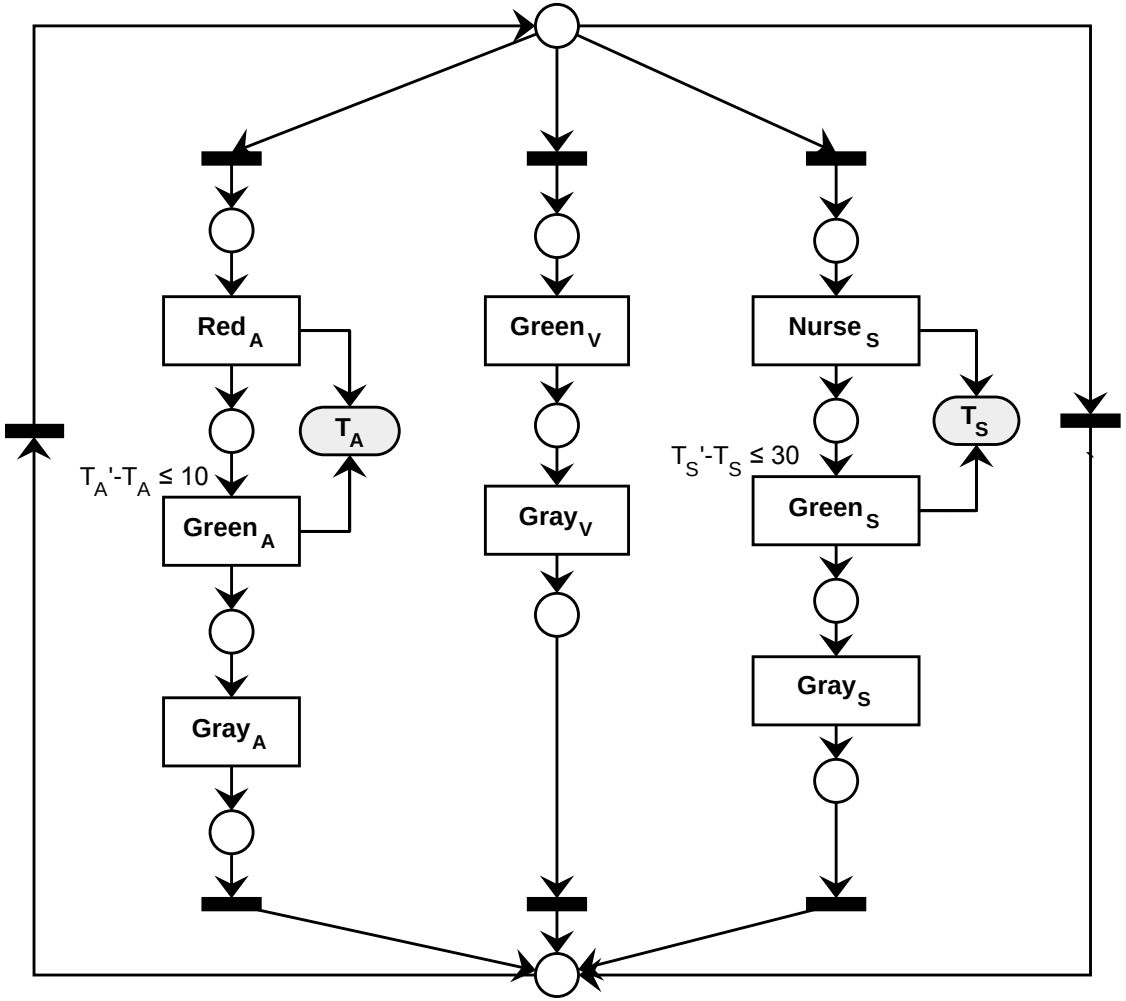
2) Build an integrated abstraction model

Abstraction Model

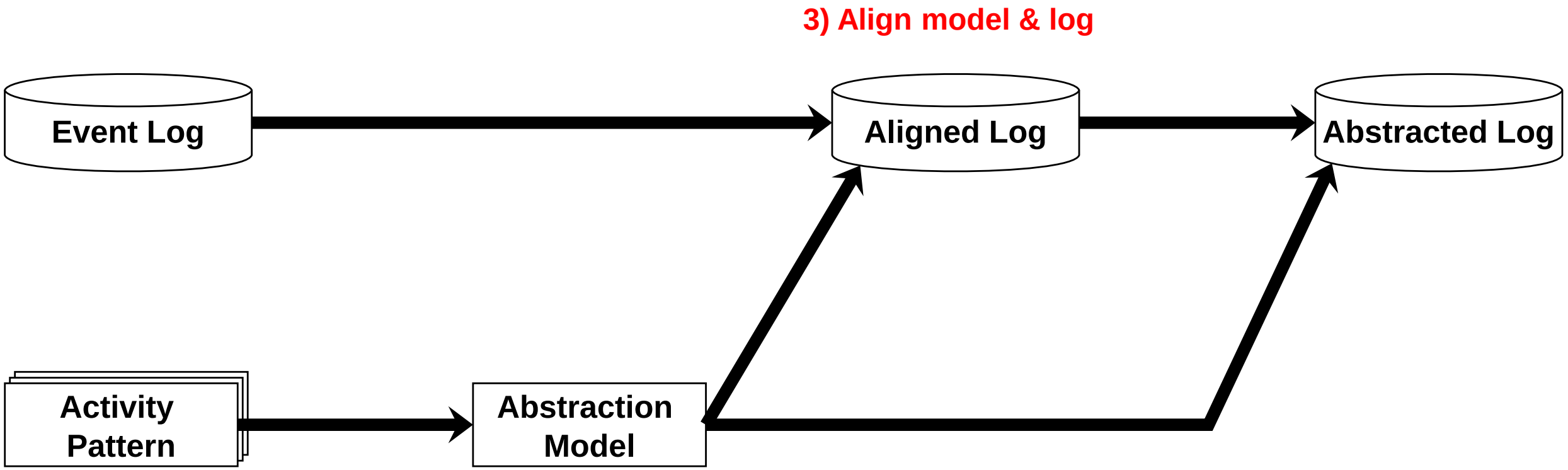


Automatic compilation

Compiled Abstraction Model (DPN)



From Low-level Events to Activities



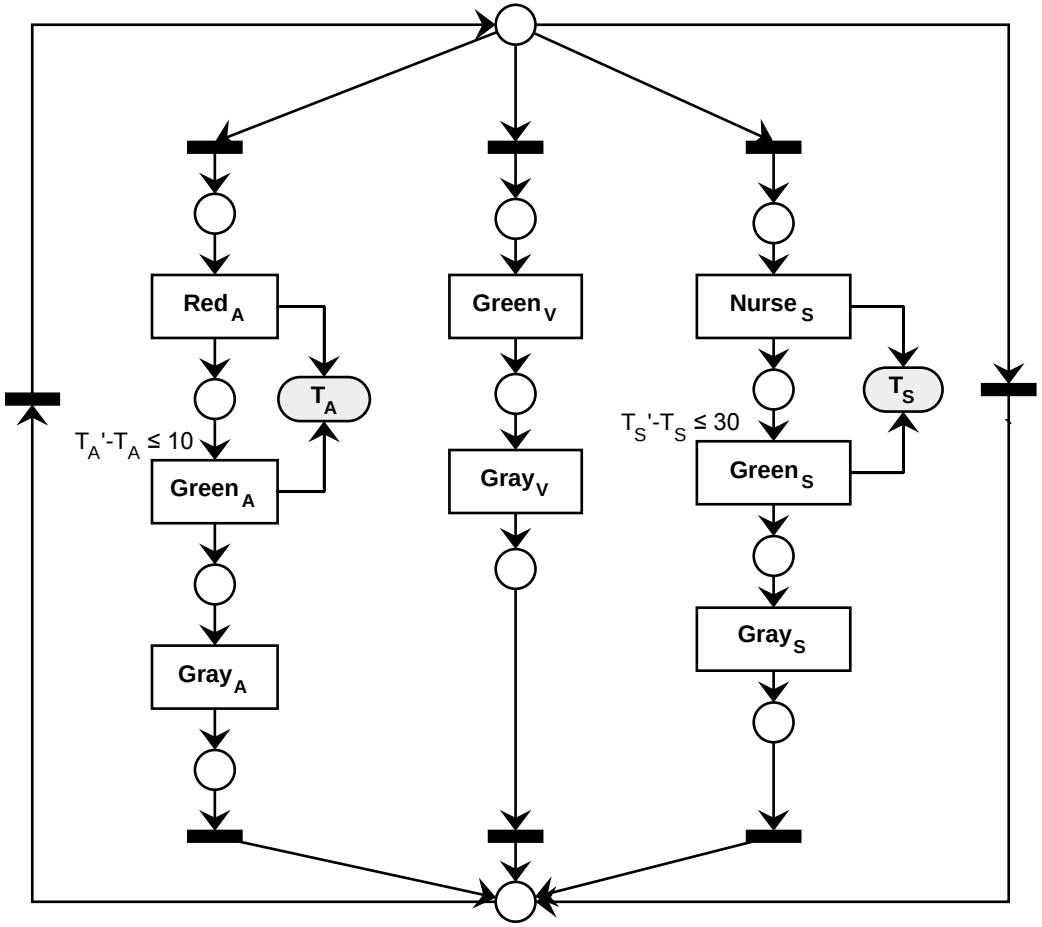
3) Align event log to abstraction model

Event Log

Low-level Event	Time
Red Button	20:08:00
Green Button	20:10:00
Green Button	22:02:00
Nurse Changed	22:09:00
Gray Button	22:10:00
Nurse Changed	22:15:00
Nurse Changed	07:02:00
Green Button	07:15:00

Alignment

Compiled abstraction model

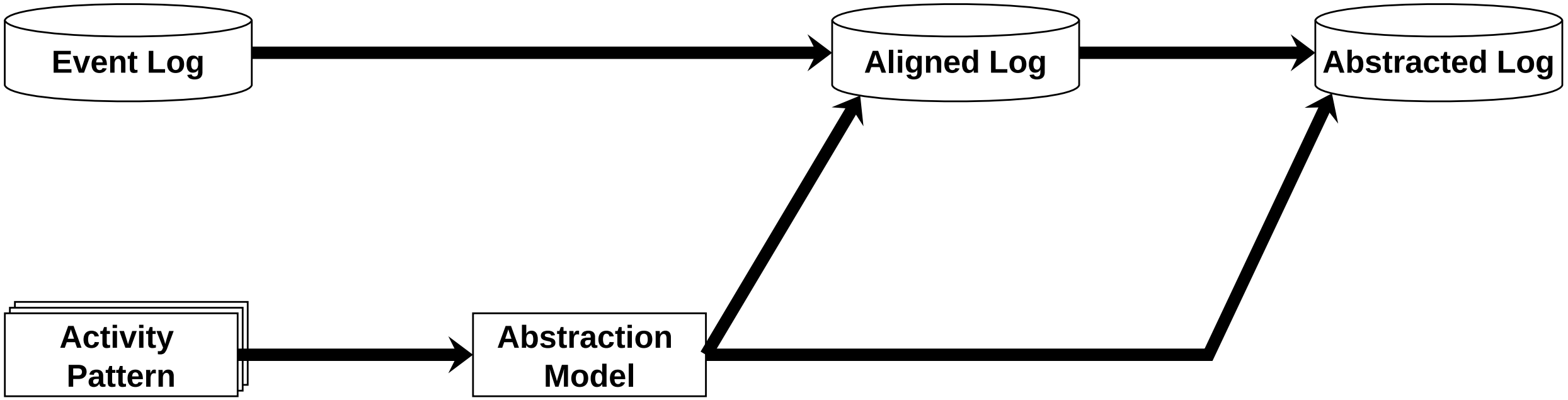


3) Align event log to abstraction model

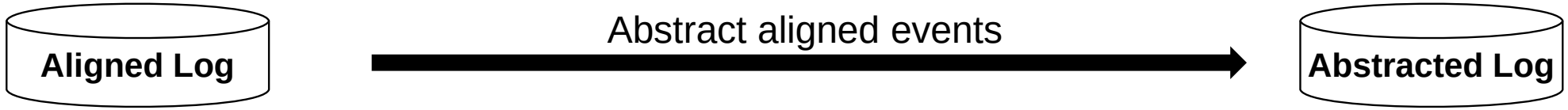
Low-level Event	Time		Process Step	Time
Red Button	20:08:00		Red_A	20:08:00
Green Button	20:10:00		$Green_A$	20:10:00
			$Gray_A$	
		Model only		
Green Button	22:02:00		$Green_V$	22:02:00
Nurse Changed	22:09:00			
		Log only		
Gray Button	22:10:00		$Gray_V$	22:10:00
		...		
Nurse Changed	22:15:00			
Nurse Changed	07:02:00		$Nurse_S$	07:02:00
Green Button	07:15:00		$Green_S$	07:15:00

From Low-level Events to Activities

4) Abstract events



4) Create the abstracted event log with high-level events



Process Step	Time
Red_A	20:08:00
$Green_A$	20:10:00
$Gray_A$	
$Green_V$	22:02:00
$Gray_V$	22:10:00
$Nurse_S$	07:02:00
$Green_S$	07:15:00
$Gray_S$	07:18:00

Alarm

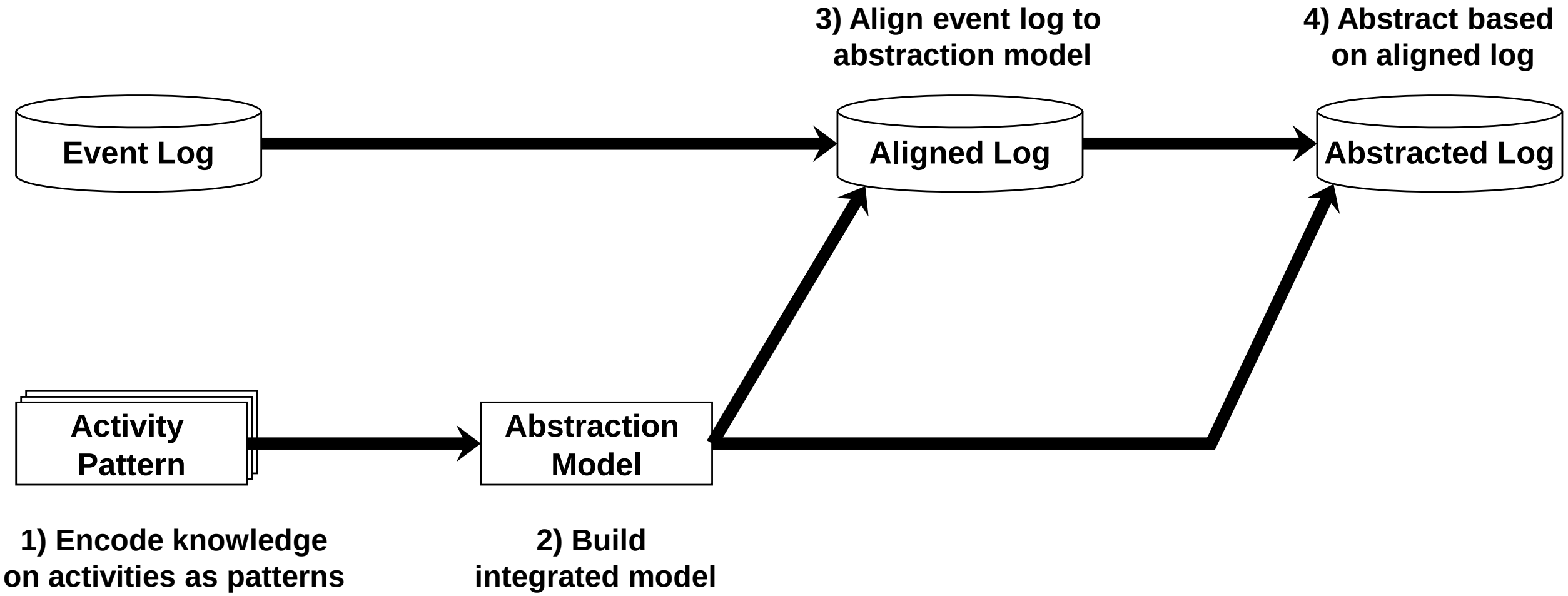
Visit

Handover

High-l		Time
Alarm		20:08:00
Alarm	complete	20:10:00
Visit	start	22:02:00
Visit	complete	22:10:00
Handover	start	07:02:00
Handover	complete	07:18:00

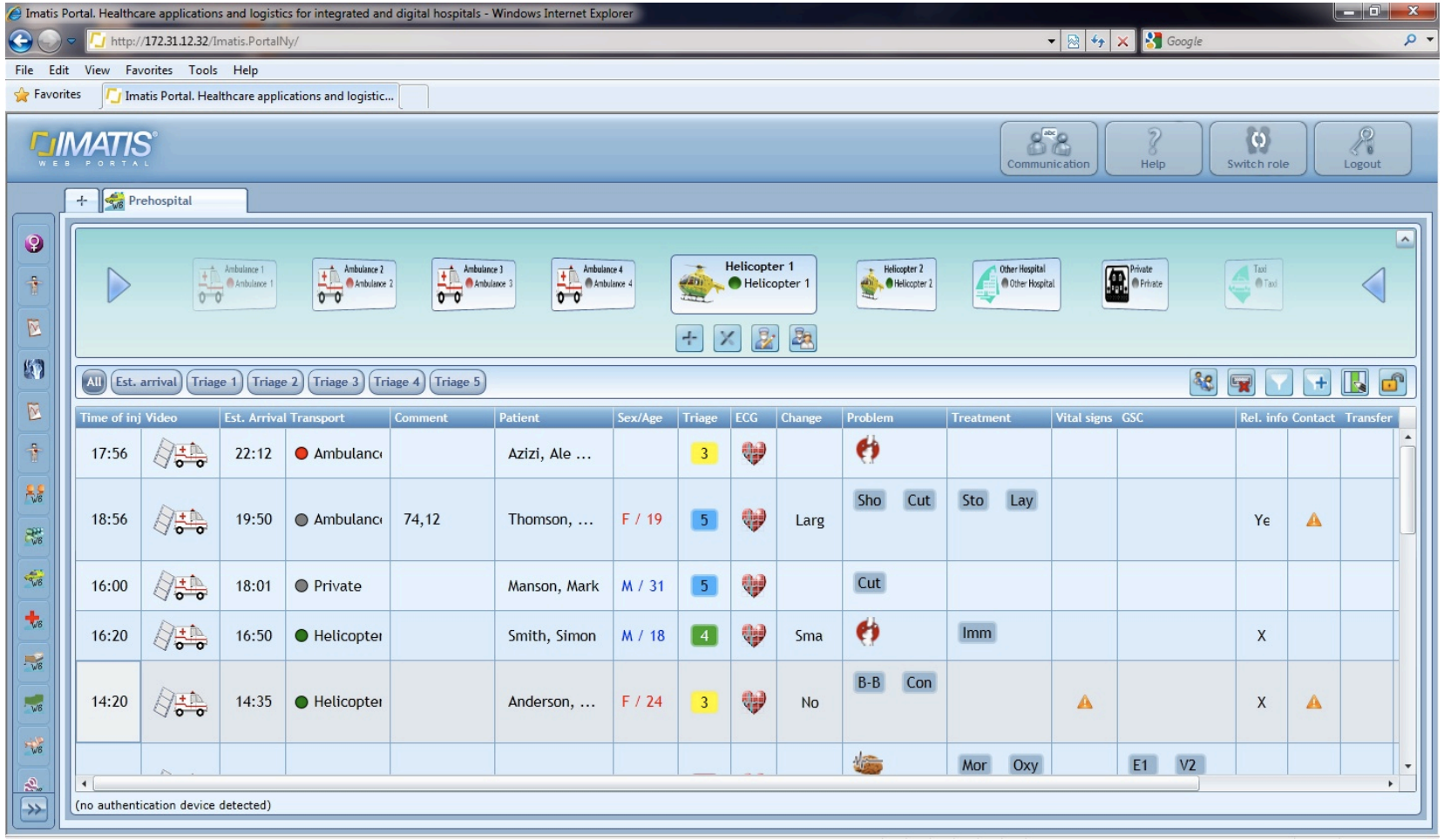
Use time of previous mapped event.

Recap: Abstraction Method



Evaluation: Digital whiteboard system in a hospital

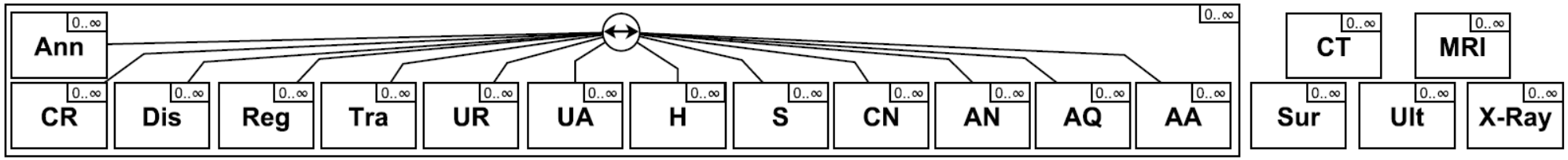
- **Information system**
  - Digital whiteboard
  - Supports work of nurses
  - Mixed clinical & logistic info
  - Flexible system
- **Dataset**
  - One year
  - > 8,000 cases
  - > 280,000 events
  - Event per changed cell



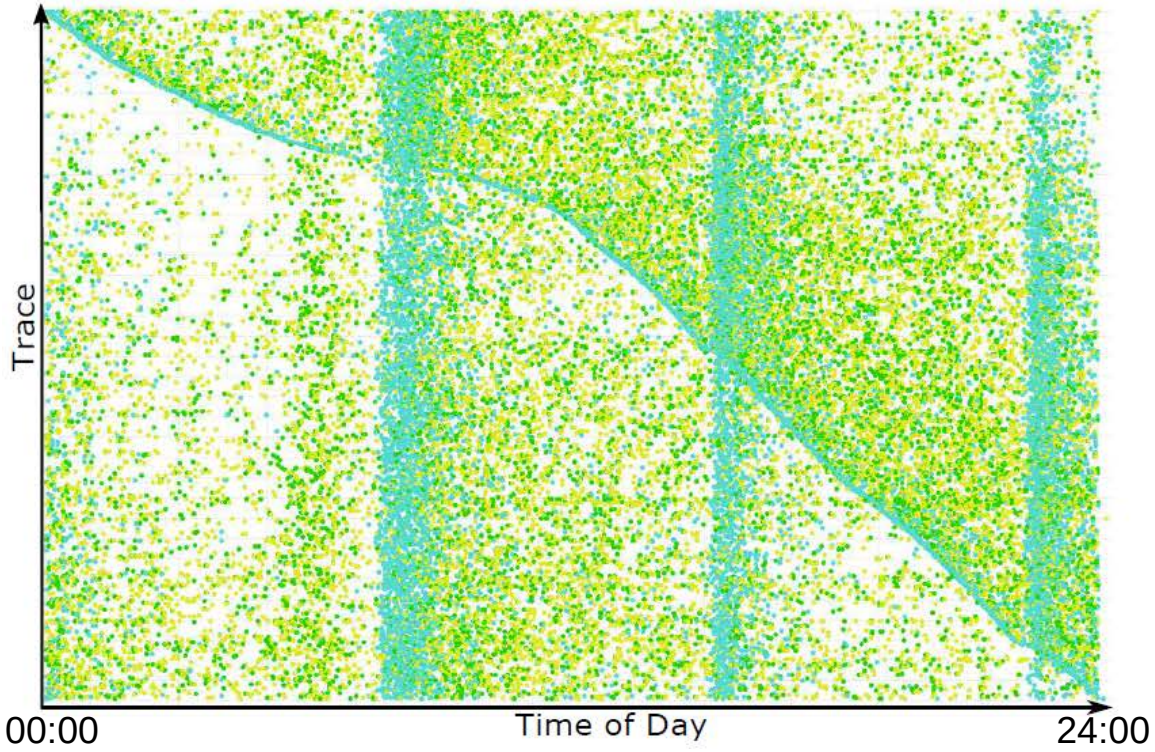
Evaluation: Activity Patterns

Activity Name	Transitions (Shared)
Announcement (Ann)	8 (6)
Change Room (CR)	5 (4)
Discharge (Dis)	7 (4)
Registration (Reg)	6 (6)
Transfer (Tra)	6 (6)
Update Report (UR)	4 (0)
Update Arrival (UA)	5 (1)
Handover (H)	1 (1)
Shift (S)	3 (3)
Call Nurse (CN)	2 (2)
Alarm Normal (AN)	3 (3)
Alarm Quick (AQ)	2 (2)
Alarm Assist (AA)	5 (3)
CT	4 (2)
MRI	4 (2)
Surgery (Sur)	3 (3)
Ultrasound (Ult)	5 (3)
X-Ray	4 (2)

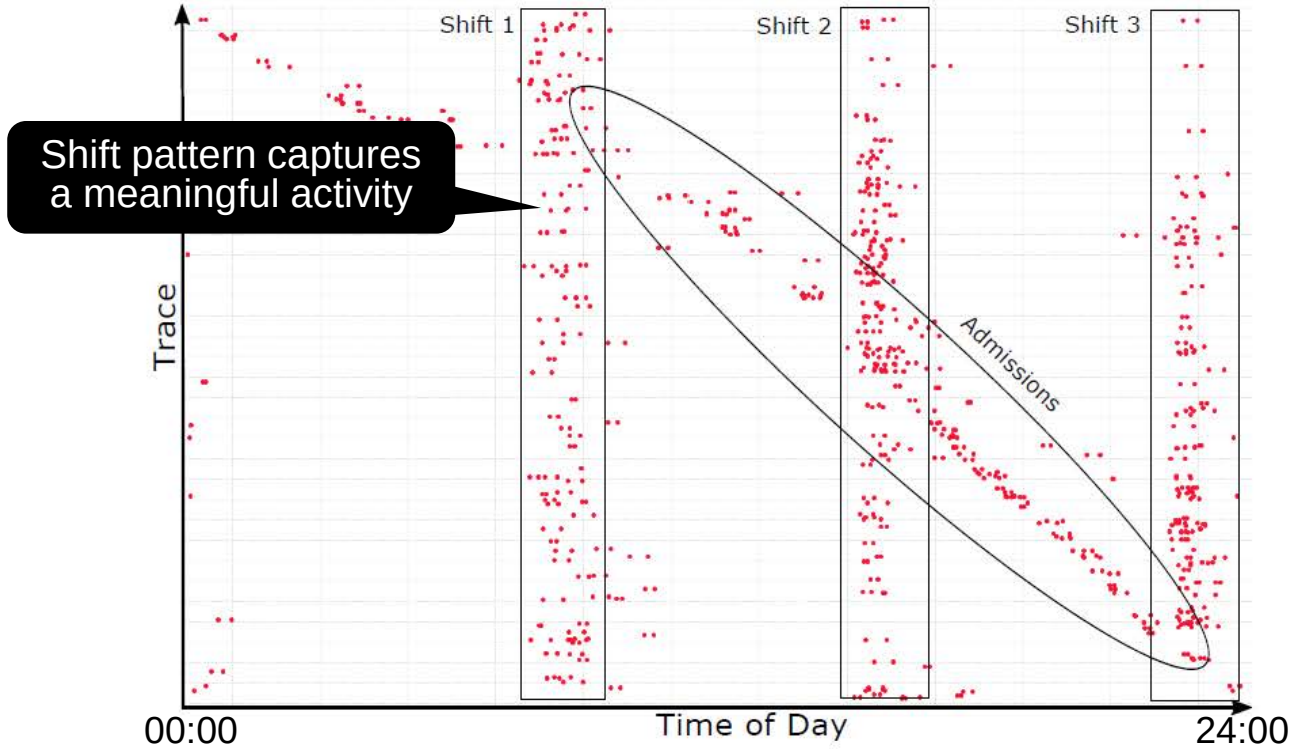
- **18 activity patterns**
  - Our assumptions
  - Interview with expert
- **Abstraction model**
  - Most interleaved & repeated
  - Five concurrent activities
- **Resulting abstraction**
  - Low average error rate
  - Successful abstraction



Evaluation: Detected shift change pattern



Blue: Nurse Changed
Green: Call Signal Green
Yellow: Call Signal Gray

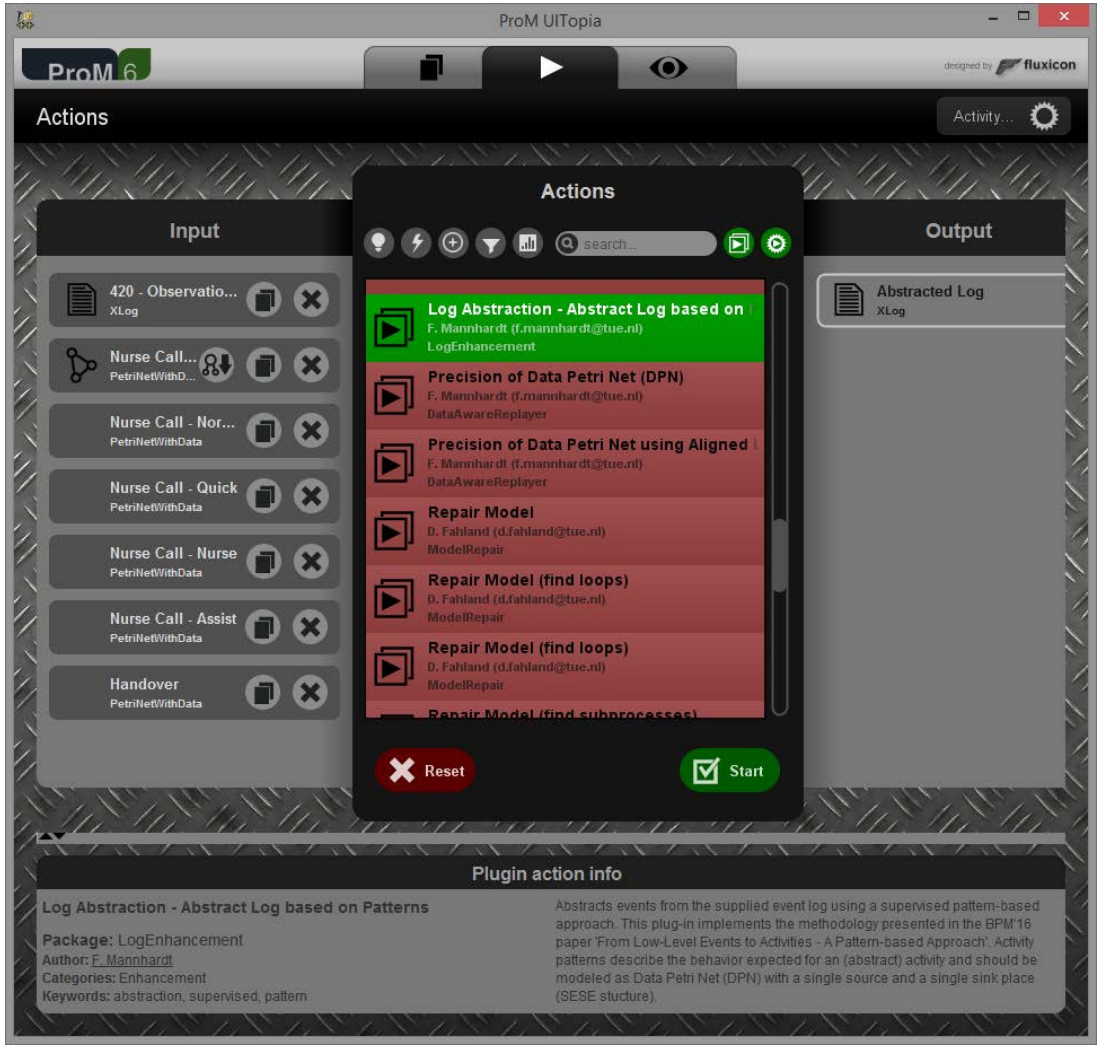


Handover (H)	1 (1)	24,228	0.0
Shift (S)	3 (3)	405	0.04
Call Nurse (CN)	2 (2)	12,416	0.08
		8,842	0.02
		12,730	0.0
		32	0.17

Relatively rare pattern!

Conclusion & Future work

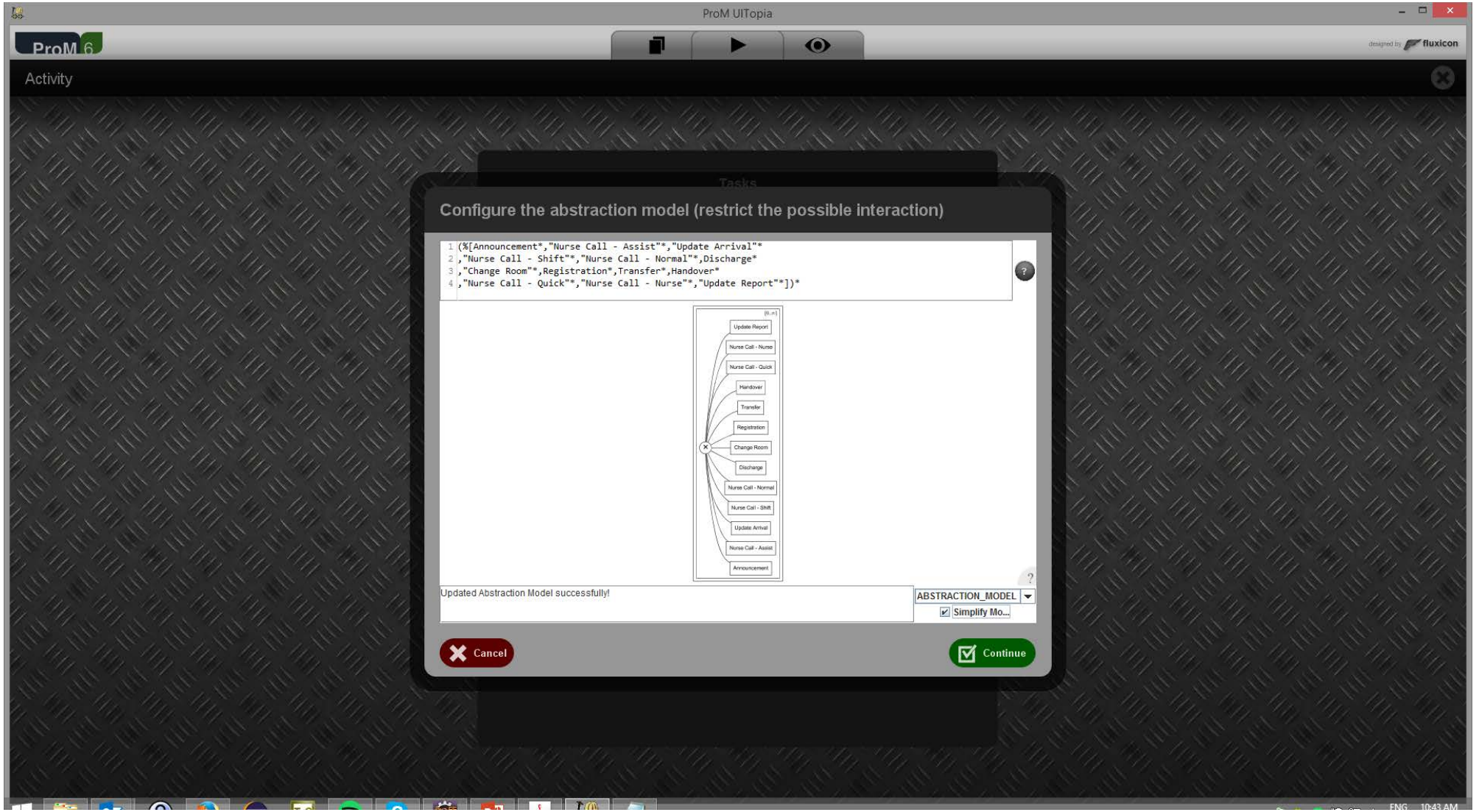
- **Method**
  - Pattern-based event abstraction
  - Knowledge encoded as activity patterns
  - Abstraction using alignment methods
- **Results**
  - Handles shared labels, concurrency and noise
  - Alignment gives reliability measure
  - Successfully used in a case study
- **Future work**
  - In-depth comparison to related methods
  - Prioritization among activity patterns
  - Decomposed of alignment methods



Implemented in ProM 6.6

Questions?

@fmannhardt - f.mannhardt@tue.nl



http://promtools.org - LogEnhancement package