FBK PDI Activity Report

Report period: June 2016 - January 2017
PDI Work Package involvement

- WP1 - Foundations of organizational modeling
- WP2 - Management of event logs
- WP3 - Hybrid monitoring
- WP4 - Inference of accurate predictions
- WP5 - Case studies
WP2

Management of event logs (M1 - M24)
Development of a comprehensive conceptual framework for the extraction of (possibly incomplete) multi-dimensional event logs from the legacy data sources of an organization.
Tasks: PDI current focus

- **T2.1** Definition of a language for the end-user modeling of log mappings. [D2.1, M6]

- **T2.2** Development of a combined framework for event log extraction by enriching OBDA with log mappings. [D2.1, M6; D2.2, M18]

- **T2.3** Management of incomplete logs. [D2.2, M18]

- **T2.4** Development of an architectural extension of the ProM OS backbone towards KOS. [D2.3, M24]
T2.3 incomplete logs: problems

![Diagram showing trace and reasoning service with labels for strong compliance, conditional compliance, model consistency, and runtime monitoring.](image)
Incomplete logs: approaches

- Abduction-based (ECAI ’16, AI*IA ’16);
- Planning-based.
An abductive logic program is a triple \((KB, A, IC)\) where:

- \(KB\) is a static knowledge base;
- \(A\) is a special set of predicates, called *abducibles*;
- \(IC\) is a set of integrity constraints.

Given a goal \(G\), the abductive reasoning looks for \(\Delta \subseteq A\) such that:

\[
KB \cup \Delta \models G \cup IC
\]

The set \(\Delta\) is referred as the *abductive explanation*.
Incomplete logs: abduction

\[ \text{KB } \cup \Delta \models \text{G } \cup \text{IC} \]

- Process model \(\Rightarrow\) IC
- Partial trace \(\Rightarrow\) KB
- \(\text{G } \Rightarrow \emptyset\)
- \(\Delta \Rightarrow\) trace completion

The (first-order logic based) framework is expressive enough to take into account \textbf{data} (currently: time).
An action language defines a dynamic domain $D$ by means of:

- **Fluents**;
- **actions**;
- **rules**;

A planning problem is a triple:

- dynamic domain $D$;
- initial state;
- goal, i.e., the final state.

The solution is a sequence of actions from initial to final state.
Incomplete logs: planning

- Fluents ⇒ process states;
- actions ⇒ process activities;
- rules ⇒ workflow + partial trace;
- initial state ⇒ process initial state;
- final state ⇒ process final state.

Data (variables) can be encoded as process states.
The role of data

Events in the trace can carry a complex payload.

We need an expressive and intuitive data-aware process model.
Relational-Aware SYStems

Features:
- Relational data model;
- tasks modifies data;
- conditions on data;
- “fresh” values can be introduced.
WP3

Hybrid Monitoring (M12 - M36)
WP3 objective (from DoW)

Definition and development of mechanisms for the hybrid monitoring (combining symbolic and statistical) of processes.
Tasks: PDI current focus

- **T3.1** Definition of a framework to combine symbolic and statistical approaches for process monitoring. [D3.1, M24]

- **T3.2** Analysis and tuning of existing machine learning techniques for hybrid monitoring approaches. [D3.2, M30]

- **T3.3** Extracting symbolic representations of properties from data. [D3.3, M36]

- **T3.4** Concept drift detection. [D3.3, M36]
T3.3: Spaghetti-like proc models
Declarative process discovery

A or B always occur but never together

A is always eventually followed by B

A and B never occur in sequence
What is already there?

Declare Miner

Apriori Algorithm

{a, b}

Automata-based constraint check

S0

S1

\[ \neg a \quad a \quad \neg b \quad b \]
A new approach

Faster Miner
(EDOC’16)
Apriori Algorithm

$\mathcal{L}$ on $\Sigma$

$\langle \text{a b c j b b d a} \rangle$
$\langle \text{a b b c d a} \rangle$
$\langle \text{a b b i i a c f} \rangle$
$\langle \text{a j j e e} \rangle$
$\langle \text{a b b c j e f b} \rangle$

Frequent activity sets

$A_1, A_2, ..., A_n \subseteq \Sigma^*$

$A_k = \{a, b\}$

- **Support**: percentage of traces containing the set of activities
- Frequent Activity Set $\rightarrow$ **high support** ($> \text{supp_threshold}$)
Sequence Analysis

Frequent activity set instantiacted on a Declare template

- An algorithm for each Declare template
- Further optimizations: Pruning and Multithreading
Open Points

- Deliverable format and submission
- ProM plugin for log incompleteness
References

References