A General Framework for Action-Oriented Process Mining

Gyunam Park and W.M.P. van der Aalst

Process and Data Science (PADS),
RWTH Aachen University, Aachen, Germany

1 Event data

Definition 1 (Basic Notations). *Let X denote an arbitrary set. $\mathcal{P}(X)$ denote the power set of X, i.e., $\mathcal{P}(X) = \{X' | X' \subseteq X\}$. Let $\perp$ be the null element. We denote $\widehat{X} = X \cup \{\perp\}$ to additionally include the null element. Let $X_a, ..., X_z$ be arbitrary sets and $\bar{x}_1 = (x_{a_1}, \ldots, x_{z_1}) \in X_a \times \cdots \times X_z$ a tuple of these sets. Given a partial function $f \in X \nrightarrow Y$, $dom(f) = X'$ is the domain of f such that $X' \subseteq X$.*

Definition 2 (Universes). *We define the following universes ~~to be used~~ in the remaining of this paper:*

not consistent

- *$\mathcal{U}_{ei}$ is the universe of event identifiers*
- *$\mathcal{U}_{proc}$ is the universe of process identifiers,*
- *$\mathcal{U}_{act}$ is the universe of activities,*
- *$\mathcal{U}_{res}$ is the universe of resources,*
- *$\mathcal{U}_{time}$ is the universe of timestamps,*
- *$\mathcal{U}_{oc}$ is the universe of object classes,*
- *$\mathcal{U}_{oi}$ is the universe of object identifiers,*
- *$\mathcal{U}_{omap} = \mathcal{U}_{oc} \nrightarrow \mathcal{P}(\mathcal{U}_{oi})$ is the universe of object mappings where, for $omap \in \mathcal{U}_{omap}$, we define $omap(oc) = \emptyset$ if $oc \notin dom(omap)$,*
- *$\mathcal{U}_{attr}$ be the universe of attribute names,*
- *$\mathcal{U}_{val}$ the universe of attribute values,*
- *$\mathcal{U}_{vmap} = \mathcal{U}_{attr} \nrightarrow \mathcal{U}_{val}$ is the universe of value mappings where, for $vmap \in \mathcal{U}_{vmap}$, we define $vmap(attr) = \perp$ if $attr \notin dom(vmap)$.*
- *$\mathcal{U}_{event} = \mathcal{U}_{ei} \times \mathcal{U}_{proc} \times \mathcal{U}_{act} \times \mathcal{U}_{res} \times \mathcal{U}_{time} \times \mathcal{U}_{omap} \times \mathcal{U}_{vmap}$ is the universe of events.*

looks weird

but this? it is correct

We assume these universes are pairwise disjoint, e.g., $\mathcal{U}_{ei} \cap \mathcal{U}_{proc} = \emptyset$.

where is π from? which universe to which?

Definition 3 (Event Projection). *Given an event $e = (ei, proc, act, res, time, omap, vmap) \in \mathcal{U}_{event}$, $\pi_{ei}(e) = ei, \pi_{proc}(e) = proc, \pi_{act}(e) = act, \pi_{res}(e) = res, \pi_{time}(e) = time, \pi_{omap}(e) = omap$, and $\pi_{vmap}(e) = vmap$.*

Definition 4 (Event Stream). *An event stream S is a (possibly infinite) set of events, i.e., $S \subseteq \mathcal{U}_{event}$ such that $\forall_{e_1,e_2 \in S}\ \pi_{ei}(e_1) = \pi_{ei}(e_2) \implies e_1 = e_2$. We let $\mathcal{U}_{stream}$ denote the universe of all possible event streams.*

first define the universe?!

where
are they
from?

Definition 5 (Time Window). *A time window* $tw = (t_s, t_e) \in \mathcal{U}_{time} \times \mathcal{U}_{time}$ *is a pair of timestamps such that* $t_s \leq t_e$. *Given a time window* $tw = (t_s, t_e)$, $\pi_{start}(tw) = t_s$ *and* $\pi_{end}(tw) = t_e$. $\mathcal{U}_{tw}$ *is the set of all possible time windows.*

Definition 6 (Time Moment). *A time moment* $tm = (t, tw) \in \mathcal{U}_{time} \times \mathcal{U}_{tw}$ *is a pair of a timestamp* t *and a time window* tw *such that* $t \geq \pi_{end}(tw)$. *Given* $tm = (t, tw)$, *we indicate* $\pi_t(tm) = t$ *and* $\pi_{tw}(tm) = tw$. $\mathcal{U}_{tm}$ *is the set of all possible time moments.*

Don't get the
point of the
following

2 A General Framework for Action-oriented Process Mining

2.1 Overview

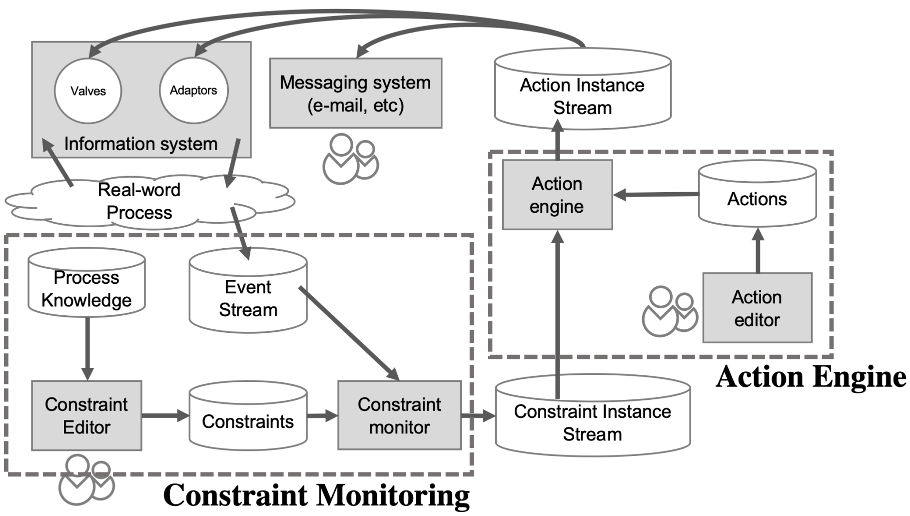

Fig. 1. Overview of a general framework for action-oriented process mining (AOPM)

a or the?

2.2 Constraint Monitoring

set theory

above you used A why the difference? Don't know A

Definition 7 (Constraint Formula). *We define* $\mathcal{U}_{outc} = \{OK, NOK\}$ *to be the universe of outcomes.* $cf \in (\mathcal{U}_{stream} \times \mathcal{U}_{tw}) \to \widehat{\mathcal{U}}_{proc} \times \widehat{\mathcal{U}}_{act} \times \widehat{\mathcal{U}}_{res} \times \mathcal{U}_{omap} \times \mathcal{U}_{vmap} \times \mathcal{U}_{outc})$ *is a constraint formula such that, for* ~~*any*~~ $S \in \mathcal{U}_{stream}$ ~~*and*~~ $tw \in \mathcal{U}_{tw}$,

– $\nexists_{(proc,res,act,omap,vmap,outc) \in cf(S,tw)} (proc = \perp \wedge res = \perp \wedge act = \perp \wedge dom(omap) = \emptyset \wedge dom(vmap) = \emptyset)$, *and*

space space space

– $\forall_{(proc,res,act,omap,vmap,outc),(proc',res',act',omap',vmap',outc')\in cf(S,tw)}.proc = proc' \wedge act = act' \wedge res = res' \wedge omap = omap' \wedge vmap = vmap' \implies outc = outc'$.

universe?

$\mathcal{U}_{cf}$ *is the set of all possible constraint formulas.*

Definition 8 (Constraint). *A constraint* $c = (cf, TM) \in \mathcal{U}_{cf} \times \mathcal{P}(\mathcal{U}_{tm})$ *is a pair of a constraint formula* cf *and a set of time moments* TM. $\mathcal{U}_c$ *is the set of all possible constraints.*

universe?

Definition 9 (Constraint Instance). *A constraint instance* $ci \in \mathcal{U}_{cf} \times \widehat{\mathcal{U}}_{proc} \times \widehat{\mathcal{U}}_{act} \times \widehat{\mathcal{U}}_{res} \times \mathcal{U}_{omap} \times \mathcal{U}_{vmap} \times \mathcal{U}_{time} \times \mathcal{U}_{outc}$ *is a tuple of a constraint formula* cf*, a process identifier* $proc$*, an activity* act*, a resource* res*, a object mapping* $omap$*, a value mapping* $vmap$*, a timestamp* $time$*, and an outcome* $outc$. $\mathcal{U}_{ci}$ *is the set of all possible constraint instances.*

ci = (cf, act, res, omap, ...

write it out if you have the variable parts later

Definition 10 (Constraint Instance Projection). *Given a constraint instance* $ci = (cf, proc, act, res, omap, vmap, time, outc) \in \mathcal{U}_{ci}$, $\pi_{cf}(ci) = cf, \pi_{proc}(ci) = proc, \pi_{act}(ci) = act, \pi_{res}(ci) = res, \pi_{omap}(ci) = omap, \pi_{vmap}(ci) = vmap, \pi_{time}(ci) = time,$ *and* $\pi_{outc}(ci) = outc$. again where is π coming from?

Definition 11 (Constraint Instance Stream). *A constraint instance stream* CIS *is a (possibly infinite) set of constraint instances, i.e.,* $CIS \subseteq \mathcal{U}_{ci}$. $\mathcal{U}_{CIS}$ *is the set of all possible constraint instance streams.*

Definition 12 (Constraint Monitoring). *Let* $C \subseteq \mathcal{U}_c$ *be a set of constraints to be used for monitoring.* $cm_C \in \mathcal{U}_{stream} \to \mathcal{U}_{CIS}$ *is a constraint monitoring such that, for any* $S \in \mathcal{U}_{stream}$, $cm_C(S) = \{(cf, proc, act, res, omap, vmap, time, outc) \in \mathcal{U}_{ci} | \exists_{TM,tm}(cf, TM) \in C \wedge tm \in TM \wedge time = \pi_t(tm) \wedge (proc, act, res, omap, vmap, outc) \in cf(S, \pi_{tw}(tm))\}$.

2.3 Action Engine

Definition 13 (Transaction). *Let* $\mathcal{U}_{op}$ *be the universe of operations that are executed by information systems (e.g., send emails). A transaction* $tr = (op, vmap) \in \mathcal{U}_{op} \times \mathcal{U}_{vmap}$ *is a pair of an operation* op *and a parameter mapping* $vmap$. $\mathcal{U}_{tr} \subseteq \mathcal{U}_{op} \times \mathcal{U}_{vmap}$ *denotes the set of all possible transactions.*

universe $\mathcal{U}_{af}$ missing

Definition 14 (Action Formula). *An action formula* $af \in (\mathcal{U}_{CIS} \times \mathcal{U}_{tw}) \to \mathcal{P}(\mathcal{U}_{tr})$ *is a function that generates a set of transactions by analyzing the constraint instances in a constraint instance stream that belong to a time window.* ? not mathematically clear to me

Definition 15 (Action). *An action* $\alpha = (af, TM) \in \mathcal{U}_{af} \times \mathcal{P}(\mathcal{U}_{tm})$ *is a pair of an action formula* af *and a set of time moments* TM. $\mathcal{U}_{\alpha}$ *denotes the set of all possible actions.*

should be close together if possible

Definition 16 (Action Instance). *An action instance* $ai = (af, tr, time) \in \mathcal{U}_{af} \times \mathcal{U}_{tr} \times \mathcal{U}_{time}$ *is a tuple of an action formula* af*, a transaction* tr*, and a timestamp* $time$. $\mathcal{U}_{ai}$ *is the set of all possible action instances. Given an action instance* $ai = (af, tr, time) \in \mathcal{U}_{ai}$, $\pi_{af}(ai) = af, \pi_{tr}(ai) = tr,$ *and* $\pi_{time}(ai) = time$. π?

You use ^ by universes above for identifier universes. It is not so nice to have it for instance, too.

You define this already above uncomplete.

Definition 17 (Action Instance Projection). *Given an action instance* $ai = (af, tr, time) \in \mathcal{U}_{ai}$, $\pi_{af}(ai) = af, \pi_{tr}(ai) = tr$, *and* $\pi_{time}(ai) = time$.

2

Definition 18 (Action Instance Stream). *An action instance stream* AIS *is a (possibly infinite) set of action instances, i.e.,* $AIS \subseteq \mathcal{U}_{ai}$. $\mathcal{U}_{AIS}$ *is the set of all possible action instance streams.*

Definition 19 (Action Engine). *Let* $A \subseteq \mathcal{U}_{\alpha}$ *be a set of actions to be used by an action engine.* $ae_A \in \mathcal{U}_{CIS} \rightarrow \mathcal{U}_{AIS}$ *is an action engine such that, for ~~any~~* $CIS \in \mathcal{U}_{CIS}$, $ae_A(CIS) = \{(af, tr, time) \in \mathcal{U}_{ai} | \exists_{TM,tm} \ (af, TM) \in A \wedge tm \in TM \wedge time = \pi_t(tm) \wedge tr \in af(CIS, \pi_{tw}(tm))\}$.

References

General note:
It would be a little bit easier if you would have an example showing what eg a constraint is etc. Your concept helps to combine the definitions, but still it is not completely clear from a real-life point of view.

- For the rest I think the definitions are good (will already accepted so they should be :))
There are some slight inconsistencies, but nothing really wrong

:)