Guided Interaction Exploration in Artifact-centric Process Models

van Eck, M.L.; Sidorova, N.; van der Aalst, W.M.P.

Published in:
Proceedings - 2017 IEEE 19th Conference on Business Informatics, CBI 2017

DOI:
10.1109/CBI.2017.42

Published: 14/08/2017

Document Version
Accepted manuscript including changes made at the peer-review stage

Guided Interaction Exploration in Artifact-centric Process Models

Maikel L. van Eck*, Natalia Sidorova, Wil M.P. van der Aalst
Eindhoven University of Technology, The Netherlands
Email: {m.l.v.eck,n.sidorova,w.m.p.v.d.aalst}@tue.nl

***Abstract*—Artifact-centric process models aim to describe complex processes as a collection of interacting artifacts. Recent development in process mining allow for the discovery of such models. However, the focus is often on the representation of the individual artifacts rather than their interactions. Based on event data we can automatically discover composite state machines representing artifact-centric processes. Moreover, we provide ways of visualizing and quantifying interactions among different artifacts. For example, we are able to highlight strongly correlated behaviours in different artifacts. The approach has been fully implemented as a ProM plug-in; the CSM Miner provides an interactive artifact-centric process discovery tool focussing on interactions. The approach has been evaluated using real life data sets, including the personal loan and overdraft process of a Dutch financial institution.**

I. Introduction

Process discovery is the automated creation of process models that explain the behaviour captured in event data [1]. These process models can be studied e.g. to identify interesting process flows that differ from the process behaviour expected by a process expert or analyst. However, complex process behaviour can result in unstructured process models, which makes them difficult and time-consuming to analyse. Furthermore, there are often multiple views on the same process, and analysts do not always know what they are looking for.

One of the sources of complexity of discovered process models is that many process discovery approaches produce models that provide a monolithic view on the real process [1], [2]. These models generally explain the behaviour of a process in terms of the life-cycle of a single process instance. However, in reality a process instance may involve several interacting process objects or artifacts, each with their own life-cycle [3], [4]. For example, a procurement process with order and invoice objects, the behavioural process of a smart product with sensors that detect the product's state for different physical aspects, or the status of a single resource in terms of its status in the different processes it is involved in.

Recently, it has become possible to automatically discover models for process artifacts and their behavioural interactions [2], [4], [5]. These techniques produce individual process models for each artifact or perspective similar to traditional process discovery approaches. The addition of artifact interaction enriches the individual models, connecting process elements from different artifact models. Such information highlights e.g. whether a specific state in one artifact coincides with the state of another artifact.

Artifact-centric techniques can provide more structured process models than traditional discovery approaches [2]. However, decomposing the behaviour of a process into interacting artifacts does not necessarily make the overall process easier to understand. Therefore, we present an approach to support the *analysis of behavioural interactions between process artifacts*. The goal is to find the most interesting or relevant interactions so that an analyst can inspect these first. This helps process analysts faced with complex processes featuring artifacts interacting in a bigger system.

There are different ways to interpret the interaction of artifacts [2], [5], [6]. We are interested in finding implications that given the occurrence of an element of one artifact-lifecycle provide information on the possible behaviour of other artifacts. Process data generally does not explicitly contain these interactions or causal relations between artifact behaviour, so instead, we use information on *correlations between artifact behaviour to obtain such insights*.

The analysis guidance involves the use of measures of interestingness to quantify artifact interactions. Such measures have been developed in the field of association rule learning to quantify the relevance of relations between sets of items [7], [8]. In this work we show how these measures can be defined in the context of process artifact interaction. Based on these measures a ranking of artifact interactions can be presented to process analysts when inspecting process discovery results. We have extended our artifact-centric process discovery tool, the CSM Miner [9] in the ProM process mining framework, to support the explanation and analysis of interactions.

To evaluate the use of analysis guidance in practice we have used the developed tool with real life process data. We discuss the results of this analysis and compare it to insights obtained by other researchers using traditional process mining approaches on the same data. This evaluation shows that the analysis guidance provides insights into the overall process behaviour by highlighting interesting artifact interactions.

The remainder of this paper is structured as follows. First, in Section II we discuss related work on artifact-centric process mining and measures of interestingness. In Section III we introduce a way to model processes representing artifact systems and define artifact interactions. Then in Section IV we define measures of interestingness in the context of process artifacts. We present the implementation of the analysis guidance in the

*This research was performed in the IMPULS collaboration project of Eindhoven University of Technology and Philips: "Mine your own body".

CSM Miner in Section V. We evaluate the tool using real life process data in Section VI. Finally, in Section VII we present future work and conclusions.

II. Related Work

A plethora of algorithms and tools for automated process discovery emerged over the last decade [1]. These produce models in various process model notations. Several approaches have also been developed to take an object-oriented or artifact-centric view of process mining [3], [4]. However, the number of techniques that can automatically discover the interactions between artifact models is limited [2], [5].

There are different types of behavioural interaction between artifacts that can be mined from process execution data. Like in monolithic process discovery, it is possible to establish causal dependencies between events that occur in different artifacts [5]. It is also possible to link a stage in one artifact lifecycle to stages in related artifact lifecycles by discovering synchronization conditions [6]. Similarly, one can identify artifact interaction defined as the co-occurrence of states and transitions from different artifacts as part of the states and transitions of the entire process [2].

The goal of the analysis of process artifacts and their interaction is to help the user understand complex behaviour by providing additional structure to the process through decomposition. There are several other existing approaches in process mining to deal with model complexity. Most process discovery tools have filtering options or sliders to adjust which activities and dependencies between activities are shown, often based on frequency information [1]. For some types of processes it is also possible to discover hierarchical process models that allow the analysis of a process at different levels of detail [10]. Trace clustering is a technique to decompose the process data of flexible processes with many different process instance variants that share little overlap in behaviour [11]. The clustered process instances are used to mine a more limited model with fewer and stronger dependencies between activities. However, all these approaches simplify the real behaviour shown by the data and hide information instead of using the complete information to guide the analyst.

Understanding the relations between artifacts and their effect on the overall process behaviour is a challenge [5]. For complex processes this requires the analysis of large numbers of possible artifact interactions, many of which are not interesting. This problem is related to the problem in association rule learning that association rule mining algorithms produce large numbers of rules that are not equally relevant [7], [8], [12]. A solution in association rule learning for this problem involves the quantification of the interestingness of the association rules using specific measures of interestingness.

III. Modelling of Artifact Systems

In this work we use the notion of state machines to model processes representing artifact systems and the life-cycles of artifacts as presented in [2]. We developed the CSM Miner to support such models [9].

Regarding notation, we write σ_k for the k-th element of a sequence $\sigma \in S^*$ of elements from some set S, and $|\sigma|$ denotes the length of σ. We write $s \in \sigma$ if $s = \sigma_k$ for some k and $\sigma\langle s, \ldots, s'\rangle$ for the concatenation of σ with sequence $\langle s, \ldots, s'\rangle$. Additionally, for $s \in S_1 \times \cdots \times S_n$ we write $s(i)$ for the value of the i-th component of s ($i \in \{1, \ldots, n\}$).

A. Composite State Machines

A process consisting of a number of interacting artifacts is called an artifact system, and we model its behaviour as a *Composite State Machine* (CSM). The state of a CSM is defined as the composition of the states of its artifacts, i.e. it is a vector of states. The set of all possible states of a CSM is a subset of the cartesian product of the sets of states of its artifacts, as not all combinations of artifact states are necessarily possible. Each transition in a CSM represents a change in the state of at least one artifact; we do not allow self loops. Formally:

Definition 1. *A* Composite State Machine $\mathcal{M} = (S, T, b, f)$ *is a model of a process with* n *artifacts where* $S \subseteq (S_1 \times \cdots \times S_n)$ *is a set of states, with* $S_1, \ldots, S_n$ *the sets of artifact states,* $b = (b_1, \ldots, b_n)$ *is the initial source state,* $f = (f_1, \ldots, f_n)$ *is the final sink state,* $T \subseteq (S \cup \{b\}) \times (S \cup \{f\})$ *is the set of transitions, and* $\forall (s, s') \in T : s \neq s'$*. We define* $\overline{S} = S \cup \{b, f\}$ *and* $\overline{S}_i = S_i \cup \{b_i, f_i\}$ *for* $i \in \{1, \ldots, n\}$.

The explicit initial and final states have no incoming and outgoing transitions, respectively. They are not true states: they only mark the points in time where a process instance begins and finishes. As a special case, we call a CSM with only one artifact an *Artifact Model*, which represents the behaviour of the artifact in isolation.

We can project a CSM onto a specific subset of its artifacts to focus only on their behaviour. A *CSM Projection* is obtained by reducing the cartesian product of each state to the given subset of artifacts, merging the identical states, and omitting unnecessary transitions and self loops. As transitions represent state changes, two states of a projection are only connected by a transition if there is a transition in the CSM whose source and target are reduced to these different states.

Definition 2. *Given a CSM* $\mathcal{M}$ *and an ordered subset of indices* $\Pi = \{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$*, with* $i_1 < i_2 < \cdots < i_m$*, we define the state projection function* $\pi_\Pi : (\overline{S}_1 \times \cdots \times \overline{S}_n) \to (\overline{S}_{i_1} \times \cdots \times \overline{S}_{i_m})$ *as follows:* $\forall s \in \overline{S}, i_j \in \Pi : (\pi_\Pi(s))(j) = s(i_j)$*. A* CSM Projection *of* $\mathcal{M}$ *on* Π*,* $\mathcal{M}^\Pi = (S^\Pi, T^\Pi, b^\Pi, f^\Pi)$*, is defined as:*

$$
\begin{aligned}
S^\Pi &= \{\pi_\Pi(s) | s \in S\},\\
T^\Pi &= \{(\pi_\Pi(s), \pi_\Pi(s')) | (s, s') \in T \wedge \pi_\Pi(s) \neq \pi_\Pi(s')\},\\
b^\Pi &= \pi_\Pi(b),\\
f^\Pi &= \pi_\Pi(f).
\end{aligned}
$$

The Artifact Model $\mathcal{A}_i$ *is defined as the projection* $\mathcal{M}^{\{i\}}$ *of* $\mathcal{M}$ *on* $\{i\}$.

Note that the projection of a CSM is itself again a CSM, modelling only the behaviour of the artifacts projected on.

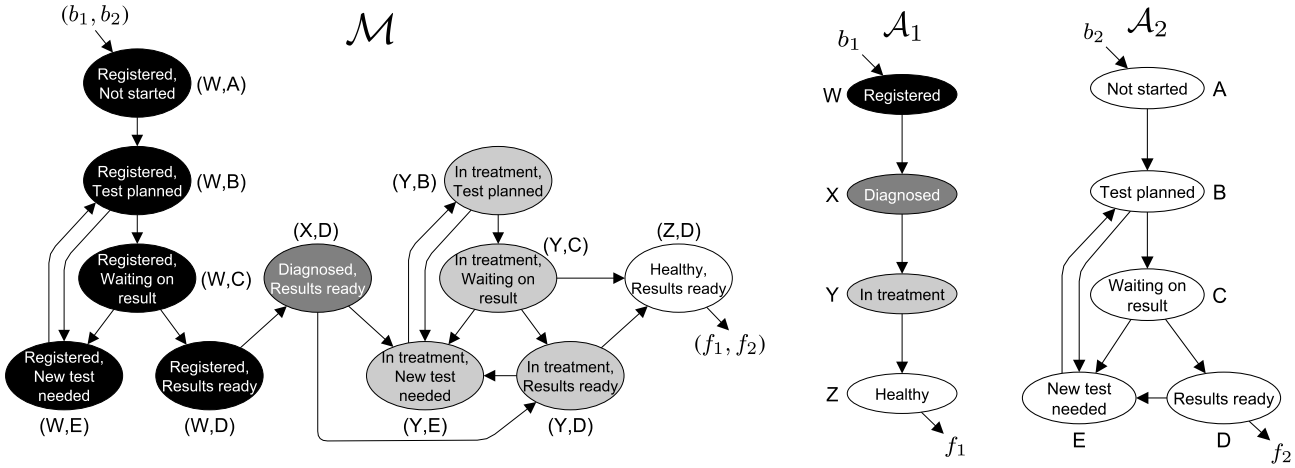

Fig. 1: A model $\mathcal{M}$ of a simple healthcare process and its two artifact models $\mathcal{A}_1$ and $\mathcal{A}_2$. Every state in the process is a combination of a state from each artifact.

In Fig. 1 we present a simple healthcare process, which we use as a running example. This process (model $\mathcal{M}$) has two distinct perspectives or artifacts: the status of the patient being treated (model $\mathcal{A}_1$), and the status of lab tests of the patient (model $\mathcal{A}_2$). The artificial initial and finial states are marked without border.

The healthcare process starts when the patient is registered, after which a lab test is planned to diagnose the patient. If the patient misses their appointment or if the results are inconclusive, then a new test is planned, but if the test results are ready then the treatment can proceed. During the treatment additional tests may be required, until the patient is healthy again and the process ends. Note that the composite process is smaller than the cartesian product of the artifacts ($4 \times 5 = 20$ states) because not all state combinations can be observed due to interdependencies. For example, once the patient is healthy no extra lab tests are needed. Such dependencies between artifacts can be interesting to analyse.

B. Process Execution Data

The CSM models as introduced above provide only limited insights into the dependencies and interaction between the artifacts whose behaviour makes up the process of the artifact system. There are no expected sojourn times for the different states or frequencies for transitions. For the process in Fig. 1 an analyst could be interested e.g. in the average time spent *Waiting on result (C)* while the patient is *In treatment (Y)* or the difference in probability of transitioning to *New test needed (E)* before and after the patient is *Diagnosed (X)*. To enrich the model with such information, we require a collection of process execution data.

In this work we assume the availability of both a CSM of the process of interest and a matching collection of process instance data consisting of execution sequences of the process. Each *State Entry* in an *Execution Sequence*, or trace, specifies the new state of the artifact system at a certain point in time. A collection of execution sequences together form a *Log*. Given a log, a CSM can be discovered that matches the execution sequences in the log [2].

Definition 3. *Let $\mathcal{M}$ be a CSM and $\mathbb{T}$ a time domain. We call $e \in (\overline{S} \times \mathbb{T})$ a* State Entry. *Function $\mathit{state}(e)$ returns the state, $\mathit{time}(e)$ returns the time, and $\mathit{state}_i(e) = \pi_{\{i\}}(\mathit{state}(e))$ returns the state projection of the state entry e.*

$\sigma \in (\overline{S} \times \mathbb{T})^*$ is an Execution Sequence *of $\mathcal{M}$ iff:*

- $\mathit{state}(\sigma_1) = b$,
- $\mathit{state}(\sigma_{|\sigma|}) = f$,
- $(\mathit{state}(\sigma_k), \mathit{state}(\sigma_{k+1})) \in T$ *for* $k \in \{1, \ldots, |\sigma| - 1\}$,
- $\mathit{time}(\sigma_1) = \mathit{time}(\sigma_2)$, *and*
- $\mathit{time}(\sigma_k) < \mathit{time}(\sigma_{k+1})$ *for* $k \in \{2, \ldots, |\sigma| - 1\}$.

The set $\mathit{Traces}_\mathcal{M}$ is the set of all possible execution sequences of $\mathcal{M}$. A Log *$\mathcal{L}_\mathcal{M} : \mathit{Traces}_\mathcal{M} \to \mathbb{N}$ is a multiset of execution sequences.*

An example of an execution sequence for the CSMs from Fig. 1 is provided in Fig. 2. Note that no time is spent in the artificial initial state b, representing the beginning of the known execution, but it is included in execution sequences to enable the calculation of the frequency of the different possible ways to start a process. Artificial final state f represents the point in time after which the process instance finished and the state is unknown.

We can use the time information in an execution sequence to calculate the time spent in a given state. By aggregating the durations of state entries over a log the models can be enriched with sojourn time statistics for each state. Similar to state sojourn times, we can also count the number of transitions occurring in a log. These numbers can be used to annotate the transitions in the process models with frequency statistics.

Definition 4. *Let σ_k be a state entry of an execution sequence $\sigma \in \mathit{Traces}_\mathcal{M}$ of CSM $\mathcal{M}$. The state entry's duration is given by:*

$$\delta(\sigma_k) = \begin{cases} \mathit{time}(\sigma_{k+1}) - \mathit{time}(\sigma_k), & \text{if } 1 \leq k < |\sigma| \\ 0, & \text{if } k = |\sigma| \end{cases}$$

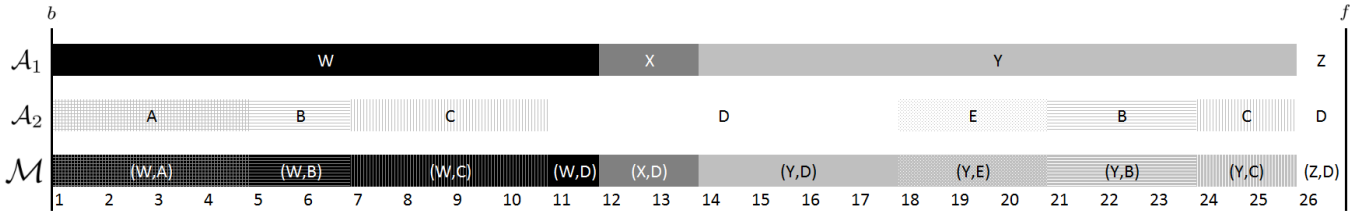

Fig. 2: An execution sequence for the running example process from Fig. 1.

TABLE I: The state entries of the execution sequence σ of $\mathcal{M}$ from Fig. 2, the sequence projected on the first artifact $\sigma' = \pi_{\{1\}}(\sigma)$, and the sequence projected on the second artifact $\sigma'' = \pi_{\{2\}}(\sigma)$.

k	σ_k	$\delta(\sigma_k)$
1	$((b_1, b_2)$, 1-1-'17)	0
2	((W,A), 1-1-'17)	4
3	((W,B), 5-1-'17)	2
4	((W,C), 7-1-'17)	4
5	((W,D), 11-1-'17)	1
6	((X,D), 12-1-'17)	2
7	((Y,D), 14-1-'17)	4
8	((Y,E), 18-1-'17)	3
9	((Y,B), 21-1-'17)	3
10	((Y,C), 24-1-'17)	2
11	((Z,D), 26-1-'17)	1
12	$((f_1, f_2)$, 27-1-'17)	0

l	σ'_l	$\delta(\sigma'_l)$
1	$(b_1$, 1-1-'17)	0
2	(W, 1-1-'17)	11
3	(X, 12-1-'17)	2
4	(Y, 14-1-'17)	12
5	(Z, 26-1-'17)	1
6	$(f_1$, 27-1-'17)	0

m	σ''_m	$\delta(\sigma''_m)$
1	$(b_2$, 1-1-'17)	0
2	(A, 1-1-'17)	4
3	(B, 5-1-'17)	2
4	(C, 7-1-'17)	4
5	(D, 11-1-'17)	7
6	(E, 18-1-'17)	3
7	(B, 21-1-'17)	3
8	(C, 24-1-'17)	2
9	(D, 26-1-'17)	1
10	$(f_2$, 27-1-'17)	0

The total sojourn time of a state $s \in S$ for a log $\mathcal{L}_\mathcal{M}$ is:

$$\mathit{soj}(s, \mathcal{L}_\mathcal{M}) = \sum_{\sigma \in \mathcal{L}_\mathcal{M}} \Big(\sum_{\{k | \mathit{state}(\sigma_k) = s\}} \delta(\sigma_k) \Big) * \mathcal{L}_\mathcal{M}(\sigma)$$

The frequency of a transition $(s, s') \in T$ for a log $\mathcal{L}_\mathcal{M}$ is:

$$\mathit{freq}_T((s, s'), \mathcal{L}_\mathcal{M}) = \sum_{\sigma \in \mathcal{L}_\mathcal{M}} \big|\{k | \mathit{state}(\sigma_k) = s \wedge \mathit{state}(\sigma_{k+1}) = s'\}\big| * \mathcal{L}_\mathcal{M}(\sigma)$$

An execution sequence of a CSM can also be projected onto a subset of its artifacts such that it is an execution sequence of the matching projected CSM. The projection abstracts from state entries where the state of the specified artifacts does not change from the previous state entry, as these entries no longer represent transitions in the projected process model. With such projections we can calculate sojourn and frequency statistics to enrich projected CSMs as before.

Definition 5. *Let $\mathcal{M}$ be a CSM, Π a set of artifact indices, and π_Π a state projection function. We lift the application of projection function π_Π to sequences $\sigma \in \mathit{Traces}_\mathcal{M}$ so that $\pi_\Pi(\sigma) \in \mathit{Traces}_{\mathcal{M}^\Pi}$. We define $\pi_\Pi(\sigma)$ recursively:*

If $\sigma = \langle\rangle$ then $\pi_\Pi(\sigma) = \langle\rangle$, and if $\sigma = \langle e \rangle$, with $e \in (\overline{S} \times \mathbb{T})$, then $\pi_\Pi(\sigma) = \langle(\pi_\Pi(\mathit{state}(e)), \mathit{time}(e))\rangle$. For an execution sequence $\sigma\langle e_1, e_2\rangle$,

$$\pi_\Pi(\sigma\langle e_1, e_2\rangle) = \begin{cases} \pi_\Pi(\sigma\langle e_1\rangle), & \text{if } \pi_\Pi(\mathit{state}(e_1)) = \pi_\Pi(\mathit{state}(e_2)) \\ \pi_\Pi(\sigma\langle e_1\rangle)\pi_\Pi(\langle e_2\rangle), & \text{otherwise} \end{cases}$$

A Log Projection $\mathcal{L}^\Pi_\mathcal{M} : \mathit{Traces}_{\mathcal{M}^\Pi} \to \mathbb{N}$ *of a log $\mathcal{L}_\mathcal{M}$ is a multiset of execution sequences such that: $\forall \varsigma \in \mathit{Traces}_{\mathcal{M}^\Pi}$: $\mathcal{L}^\Pi_\mathcal{M}(\varsigma) = \sum_{\sigma \in \mathcal{L}_\mathcal{M} : \varsigma = \pi_\Pi(\sigma)} \mathcal{L}_\mathcal{M}(\sigma)$.*

Table I shows an execution sequence σ of the running example process and its projections $\pi_\Pi(\sigma)$ for $\Pi = \{1\}$ and $\Pi = \{2\}$, together with their corresponding durations.

The information in a collection of execution sequences can be used to enrich a CSM and its projections with state sojourn statistics and transition frequencies as described above. Fig. 3 shows the running example process of Fig. 1 annotated with frequency and average sojourn time information. Process execution data can also be used for the identification of relations between artifact model elements and the calculation of measures of interestingness for such relations.

C. Artifact Interaction

Given a CSM $\mathcal{M}$ with multiple artifacts and a log $\mathcal{L}_\mathcal{M}$, we want to find interesting artifact interactions that are a part of the artifact system behaviour. For example, if the state of an artifact cannot be advanced until a certain state in a different artifact has been reached then this may represent a bottleneck in the overall process. Similarly, the probability of making specific choices at a decision point in one artifact may be affected by the state of another artifact. The executions in a log do not explicitly describe such causal dependencies between the behaviour of different artifacts, but we can infer *correlations* between sets of artifact states or transitions. Based on this, we distinguish three types of artifact interaction: *state co-occurrence*, *transition co-occurrence* and *forward-looking co-occurrence*.

We focus here only on the interaction between pairs of artifacts, but the interaction definitions can be generalised to involve sets of artifacts. We formulate each interaction as an *implication* $(X \Rightarrow Y)$ between two statements regarding the states or execution behaviour of the artifacts.

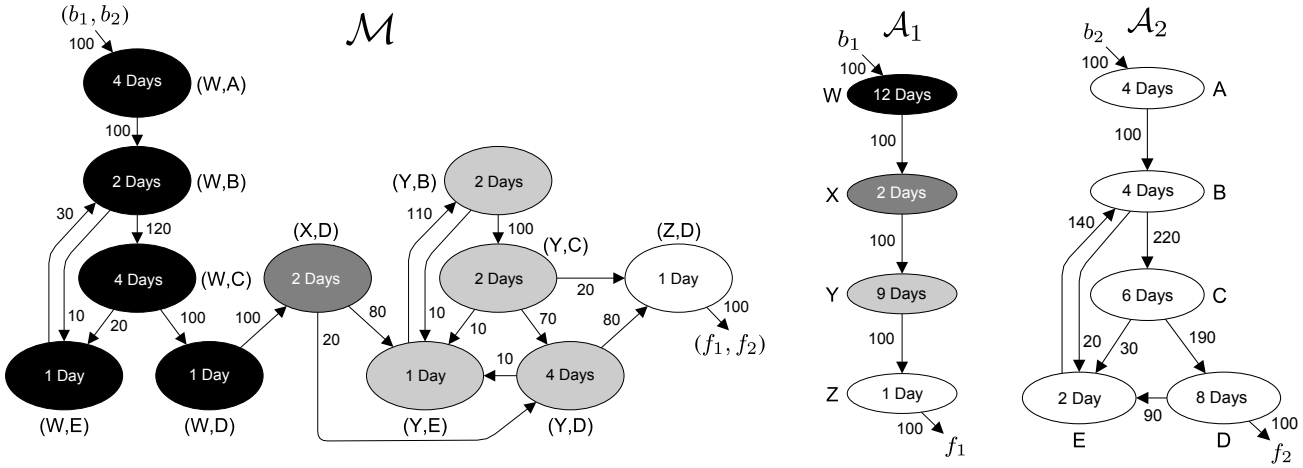

Fig. 3: The models of the healthcare process from Fig. 1 annotated with transition frequencies and average state sojourn times per trace.

State co-occurrence $(s_i \Rightarrow_S s_j)$ is defined as the conditional probability that artifact model $\mathcal{A}_j$ is in state s_j given that artifact model $\mathcal{A}_i$ is in state s_i. From the execution sequences in a log we can determine the strength of this interaction in the observed data. It is calculated as the amount of time the system state contains both states compared to the total time spent in s_i.

Definition 6. *Let $\mathcal{M}$ be a CSM with artifacts i and j, $s_i \in S_i$ and $s_j \in S_j$. The strength of the state co-occurrence $(s_i \Rightarrow_S s_j)$ is defined as:*

$$\hat{P}_S(s_i \Rightarrow_S s_j) = \frac{\mathit{soj}((s_i, s_j), \mathcal{L}_{\mathcal{M}}^{\{i,j\}})}{\mathit{soj}(s_i, \mathcal{L}_{\mathcal{M}}^{\{i\}})}$$

In Fig. 3 we can see that the average time spent *In treatment (Y)* given that the lab is *Waiting on result (C)* is 2 days, while the average time spent *Registered (W)* given that the lab is *Waiting on result (C)* is 4 days. So, the interaction $(C \Rightarrow_S W)$ is a stronger co-occurrence ($\frac{4}{6}$) than the interaction $(C \Rightarrow_S Y)$ ($\frac{2}{6}$).

Transition co-occurrence $((s_i, s_i') \Rightarrow_T (s_j, s_j'))$ is defined as the conditional probability that, given that $\mathcal{A}_i$ is in a transition from s_i to s_i', $\mathcal{A}_j$ has a state s_j before and a state s_j' after the transition. If $s_j = s_j'$ this co-occurrence specifies the state of $\mathcal{A}_j$ during the given transition in $\mathcal{A}_i$, but if they differ then it specifies a transition in $\mathcal{A}_j$ that co-occurs with the transition in $\mathcal{A}_i$. The strength of this interaction is calculated as the number of times we observe transitions for which both the condition and the consequence hold divided by the total number of observed transitions for which the condition holds.

Definition 7. *Let $\mathcal{M}$ be a CSM with artifacts i and j, $s_i, s_i' \in \overline{S}_i$, $s_i \neq s_i'$, and $s_j, s_j' \in \overline{S}_j$. The strength of the transition co-occurrence $((s_i, s_i') \Rightarrow_T (s_j, s_j'))$ is defined as:*

$$\hat{P}_T((s_i, s_i') \Rightarrow_T (s_j, s_j')) = \frac{\mathit{freq}_T(((s_i, s_j), (s_i', s_j')), \mathcal{L}_{\mathcal{M}}^{\{i,j\}})}{\mathit{freq}_T((s_i, s_i'), \mathcal{L}_{\mathcal{M}}^{\{i\}})}$$

In Fig. 3 there are three types of transitions from *Waiting on result (C)* to *Result ready (D)*: while the patient is *Registered (W)* (100 times), while the patient is *In Treatment (Y)* (70 times), and simultaneously together with a transition from *In treatment (Y)* to *Healthy (Z)* (20 times). Therefore, the strength of the transition co-occurrence $((C, D) \Rightarrow_T (W, W))$ is $\frac{100}{190}$.

Forward-looking co-occurrence $(s_i \wedge s_j \Rightarrow_{\mathcal{F}} (s_j, s_j'))$ is defined as the conditional probability that the next transition executed in $\mathcal{A}_j$ goes to state s_j', given that $\mathcal{A}_j$ is in state s_j and that $\mathcal{A}_i$ is in state s_i during and after the next transition in $\mathcal{A}_j$. The strength of this interaction is calculated as the number of times we observe a transition from s_j to s_j' while $\mathcal{A}_i$ has the specified state s_i divided by the total number of outgoing transitions from s_j while $\mathcal{A}_i$ is in s_i.

Definition 8. *Let $\mathcal{M}$ be a CSM with artifacts i and j, $s_i \in S_i$, and $s_j, s_j' \in \overline{S}_j$, $s_j \neq s_j'$. The strength of the forward-looking co-occurrence $(s_i \wedge s_j \Rightarrow_{\mathcal{F}} (s_j, s_j'))$ is defined as:*

$$\hat{P}_{\mathcal{F}}(s_i \wedge s_j \Rightarrow_{\mathcal{F}} (s_j, s_j')) = \frac{\mathit{freq}_T(((s_i, s_j), (s_i, s_j')), \mathcal{L}_{\mathcal{M}}^{\{i,j\}})}{\sum_{s_j'' \in S_j} \mathit{freq}_T(((s_i, s_j), (s_i, s_j'')), \mathcal{L}_{\mathcal{M}}^{\{i,j\}})}$$

In Fig. 3 there are transitions from *Waiting on result (C)* to *New test needed (E)* that occur while the patient is *In treatment (Y)* (10 times). While *In treatment (Y)* and *Waiting on result (C)* there are also transitions to *Result ready (D)* (70 times). Therefore the interaction $(Y \wedge C \Rightarrow_{\mathcal{F}} (C, E))$ has a strength of $\frac{10}{80}$.

It is possible to calculate the artifact interactions defined above for all pairs of states and transitions of all pairs of artifacts. However, it is clear that this results in a very large number of interactions for a process analyst to inspect. One solution to this problem is to rank and filter the list of interactions to obtain the most interesting artifact relations and to present those to the analyst first.

IV. Artifact Interaction Interestingness

In order to rank and filter artifact interactions based on their interestingness it is necessary to be able to quantify "interestingness". As we discussed in Section II, work has been performed in the field of association rule learning to develop measures of interestingness to help with the analysis of large sets of association rules [7], [8]. We have selected a number of such measures and we discuss their meaning and applicability in the context of artifact interactions that represent process behaviour.

A. Probability Interpretation

The artifact interactions we defined in Section III-C are implications over binary stochastic variables representing statements of artifact behaviour. The implications are of the form $(X \Rightarrow Y)$. Each statement X or Y is either true or false, with a certain probability that can be estimated from process execution data. The measures of interestingness objectively score statistical correlations between the variables based on these probabilities. We discuss the probabilities and their interpretations for each type of artifact interaction.

State co-occurrence $(s_i \Rightarrow_S s_j)$ is an implication between stochastic variables of the form $(X_{s_i} \Rightarrow Y_{s_j})$ with X_{s_i} defined as $\mathcal{A}_i$ *has state* s_i and Y_{s_j} defined as $\mathcal{A}_j$ *has state* s_j. The probability of X_{s_i} can be estimated based on the total sojourn time over all states:

$$\hat{P}_S(X_{s_i}) = \frac{\mathsf{soj}(s_i, \mathcal{L}_{\mathcal{M}}^{\{i\}})}{\sum_{s \in S} \mathsf{soj}(s, \mathcal{L}_{\mathcal{M}})}$$

Transition co-occurrence $((s_i, s_i') \Rightarrow_T (s_j, s_j'))$ is either an implication of the form $(X_{(s_i,s_i')} \Rightarrow Y_{s_j})$ if $s_j = s_j'$, with $X_{(s_i,s_i')}$ defined as $\mathcal{A}_i$ *is in transition from* s_i *to* s_i', or it is an implication $(X_{(s_i,s_i')} \Rightarrow Y_{(s_j,s_j')})$ if $s_j \neq s_j'$. Strictly speaking, the probability of $X_{(s_i,s_i')}$ cannot be expressed because transitions are instantaneous and on a continuous time scale the probability to be in the specific point in time where the transition occurs is infinitesimal, i.e. not distinguishable from 0. As a result, a number of measures of interestingness would not be defined for transition co-occurrence. We express the probability based on the total frequency of transitions to avoid this issue:

$$\hat{P}_T(X_{(s_i,s_i')}) = \frac{\mathsf{freq}_T((s_i, s_i'), \mathcal{L}_{\mathcal{M}}^{\{i\}})}{\sum_{(s,s') \in T} \mathsf{freq}_T((s, s'), \mathcal{L}_{\mathcal{M}})}$$

Forward-looking co-occurrence $(s_i \wedge s_j \Rightarrow_{\mathcal{F}} (s_j, s_j'))$ is of the form $(X_{s_i \wedge s_j} \Rightarrow Y_{\mathcal{F}(s_j,s_j')})$ with $X_{s_i \wedge s_j}$ defined as $\mathcal{A}_j$ *has state* s_j *and* $\mathcal{A}_i$ *has state* s_i *during the next transition in* $\mathcal{A}_j$, and $Y_{\mathcal{F}(s_j,s_j')}$ defined as *the next transition in* $\mathcal{A}_j$ *is from* s_j *to* s_j'. The probability of $X_{s_i \wedge s_j}$ is estimated by the probability to be in s_j and the frequency of s_i in all possible transitions from s_j:

$$\hat{P}_{\mathcal{F}}(X_{s_i \wedge s_j}) = \frac{\mathsf{soj}(s_j, \mathcal{L}_{\mathcal{M}}^{\{j\}})}{\sum_{s \in S} \mathsf{soj}(s, \mathcal{L}_{\mathcal{M}})} * \frac{\sum_{s_j'' \in S_j} \mathsf{freq}_T(((s_i, s_j), (s_i, s_j'')), \mathcal{L}_{\mathcal{M}}^{\{i,j\}})}{\sum_{s_j'' \in S_j} \mathsf{freq}_T((s_j, s_j''), \mathcal{L}_{\mathcal{M}}^{\{j\}})}$$

Because $Y_{\mathcal{F}(s_j,s_j')}$ is only possible if $\mathcal{A}_j$ has state s_j we can estimate it with the probability to be in s_j and the frequency of each possible outgoing transition from s_j:

$$\hat{P}_{\mathcal{F}}(Y_{\mathcal{F}(s_j,s_j')}) = \frac{\mathsf{soj}(s_j, \mathcal{L}_{\mathcal{M}}^{\{j\}})}{\sum_{s \in S} \mathsf{soj}(s, \mathcal{L}_{\mathcal{M}})} * \frac{\mathsf{freq}_T((s_j, s_j'), \mathcal{L}_{\mathcal{M}}^{\{j\}})}{\sum_{s_j'' \in S_j} \mathsf{freq}_T((s_j, s_j''), \mathcal{L}_{\mathcal{M}}^{\{j\}})}$$

B. Measures of Interestingness

Below we present a selection of measures of interestingness that have been implemented in the CSM Miner to evaluate the interestingness of artifact interactions. The motivation for this selection is that each of these measures has an intuitive interpretation, and that evaluation studies in other application areas have shown that these measures have high predictive power and low collinearity with each other when used to approximate association rule interestingness [12].

For each measure we provide a definition, a short description of its intuitive meaning and its range. The measures are defined in terms of the probabilities of observing the conditions and consequences of the implications representing the different types of artifact interaction. Some measures are symmetric, i.e. their value for $X \Rightarrow Y$ is equal for $Y \Rightarrow X$.

Confidence: The confidence of an artifact interaction is also referred to as the strength of the prediction, which we introduced for each type of artifact interaction in Section III-C. It is defined as a conditional probability:

$$\mathsf{conf}(X \Rightarrow Y) = P(X \Rightarrow Y) = P(Y|X)$$

The range of conf is $[0, 1]$ and it is asymmetric, i.e. in general $\mathsf{conf}(X \Rightarrow Y) \neq \mathsf{conf}(Y \Rightarrow X)$.

Support: In the context of association rule learning the support measure is traditionally defined as the frequency with which items occur in a set of transactions, which is an estimate of their probability of occurrence. In the setting of artifact interaction the support of individual statements is their probability interpretation as defined in the section above, e.g. $\mathsf{supp}(X_{s_i}) = \hat{P}_S(X_{s_i})$ and $\mathsf{supp}(Y_{\mathcal{F}(s_j,s_j')}) = \hat{P}_{\mathcal{F}}(Y_{\mathcal{F}(s_j,s_j')})$. The support of an implication $X \Rightarrow Y$ is then the probability that the implication is true, multiplied by the probability of observing the condition of the implication:

$$\mathsf{supp}(X \Rightarrow Y) = P(Y|X)P(X) = P(X \wedge Y)$$

The range of supp is $[0, 1]$ and it is symmetric.

Lift: The lift of an interaction is defined as the ratio between the probabilitiy of co-occurrence and the expected co-occurrence under statistical independence:

$$\mathsf{lift}(X \Rightarrow Y) = \frac{\mathsf{conf}(X \Rightarrow Y)}{\mathsf{supp}(Y)} = \frac{P(Y|X)}{P(Y)} = \frac{P(X \wedge Y)}{P(X)P(Y)}$$

The range of lift is $[0, \infty]$ and it is symmetric. A lift measure of 0 indicates that they are never observed together, a value of 1 indicates that X and Y are independent, and a value above 1 indicates that X and Y are observed together more often than can be expected under conditions of statistical independence.

Conviction: The conviction of an interaction is similar to lift, but it is a directed measure. It looks at the expected probability of observing X without Y, i.e. the frequency of the implication being incorrect. It is defined as the ratio of the frequency of the implication being incorrect, if they were statistically independent, and the frequency of actual observations of the implication not holding:

$$\mathsf{convic}(X \Rightarrow Y) = \frac{1 - \mathsf{supp}(Y)}{1 - \mathsf{conf}(X \Rightarrow Y)} = \frac{P(X)P(\overline{Y})}{P(X \wedge \overline{Y})}$$

The range of convic is $(0, \infty]$ and it is asymmetric. A conviction measure of 1 indicates that X and Y are statistically independent, while a measure value of ∞ occurs for interactions that always hold in the observed data.

Cosine: The cosine measure is defined as the geometric mean of lift and support:

$$\mathsf{cosine}(X \Rightarrow Y) = \frac{\mathsf{supp}(X \Rightarrow Y)}{\sqrt{\mathsf{supp}(X)\mathsf{supp}(Y)}} = \frac{P(X \wedge Y)}{\sqrt{P(X)P(Y)}}$$

The range of cosine is $[0, 1]$ and it is symmetric. It is a null-invariant measurement, i.e. it is not affected by the number of observations involving neither X nor Y in the dataset, while e.g. the lift measure does not have this property.

Jaccard: The jaccard of an interaction is defined as the ratio between the probability of the co-occurrence of X and Y and the probability of observing either:

$$\begin{aligned}\mathsf{jaccard}(X \Rightarrow Y) &= \frac{\mathsf{supp}(X \Rightarrow Y)}{\mathsf{supp}(X) + \mathsf{supp}(Y) - \mathsf{supp}(X \Rightarrow Y)} \\ &= \frac{P(X \wedge Y)}{P(X \vee Y)}\end{aligned}$$

The range of jaccard is $[0, 1]$, it is symmetric and a null-invariant measurement. A jaccard measure of 0 means that items from X and Y are never observed together, and a value of 1 indicates that if they occur then they are always observed together.

Phi-coefficient: The ϕ-coefficient of an interaction is defined as the normalised difference between the probability of co-occurrence and the expected probability of co-occurrence under statistical independence:

$$\phi(X \Rightarrow Y) = \frac{P(X \wedge Y) - P(X)P(Y)}{\sqrt{P(X)P(Y)(1 - P(X))(1 - P(Y))}}$$

The range of ϕ is $[-1, 1]$ and it is symmetric. A value of 0 indicates that X and Y are statistically independent.

V. Analysis Guidance Implementation

In this section we discuss the implementation of the analysis guidance in the CSM Miner [9], a plug-in[1] in the process mining framework ProM.

The CSM Miner discovers a model of the artifact system and of each artifact in the input log, annotates them with sojourn times and frequencies, and presents them in an interactive visualisation. The interaction allows the user to click on a state or transition and this will highlight all other states and transitions for which $\mathsf{supp}(X \Rightarrow Y) > 0$, based on either $\hat{P}_S$, $\hat{P}_T$ or $\hat{P}_{\mathcal{F}}$. The colour of the highlighting is dependent on $\mathsf{conf}(X \Rightarrow Y)$.

The analysis guidance for the exploration of artifact interactions is provided below the interactive model visualisation, as shown in Fig. 4. It provides a list of artifact interactions and for each interaction the measures discussed in Section IV are calculated. The user can sort the interactions by the measure values and can set minimum values for each measure to filter the list.

When clicking on the artifact interactions in the list, the user is also presented with a textual interpretation based on four possible templates:

- "$\mathsf{conf}(s_i \Rightarrow_S s_j)$ of the total time spent in s_i is spent while being in s_j" (state co-occurrence)
- "Transitions from s_i to s'_i occur $\mathsf{conf}((s_i, s'_i) \Rightarrow_T s_j)$ of the times while being in s_j" (transition co-occurrence)
- "Transitions from s_i to s'_i occur $\mathsf{conf}((s_i, s'_i) \Rightarrow_T (s_j, s'_j))$ of the times together with a transition from s_j to s'_j" (transition co-occurrence)
- "A transition from s_j goes $\mathsf{conf}(s_i \wedge s_j \Rightarrow_{\mathcal{F}} (s_j, s'_j))$ of the times to s'_j while being in s_i (compared to $\hat{P}_{\mathcal{F}}((s_j, s'_j)|s_j)$ on average)" (forward-looking co-occurrence)

VI. Evaluation

To be able to create analysis guidance that suggests artifact interactions of interest to process analysts it is necessary to establish what qualifies as interesting or relevant. The measures introduced in Section IV are suggestions to quantify interest from the field of association rule learning, but it is unclear how these measures relate to the actual judgments of interest by process analysts. The aim of this evaluation is to show that the analysis guidance highlights behaviour in real life processes that is useful for understanding the process. Therefore, we compare the suggestions provided by the tool with insights obtained by other researchers using traditional process mining approaches on real life process data.

[1] Contained in the *CSMMiner* package of the ProM 6 nightly build, available at http://www.promtools.org/.

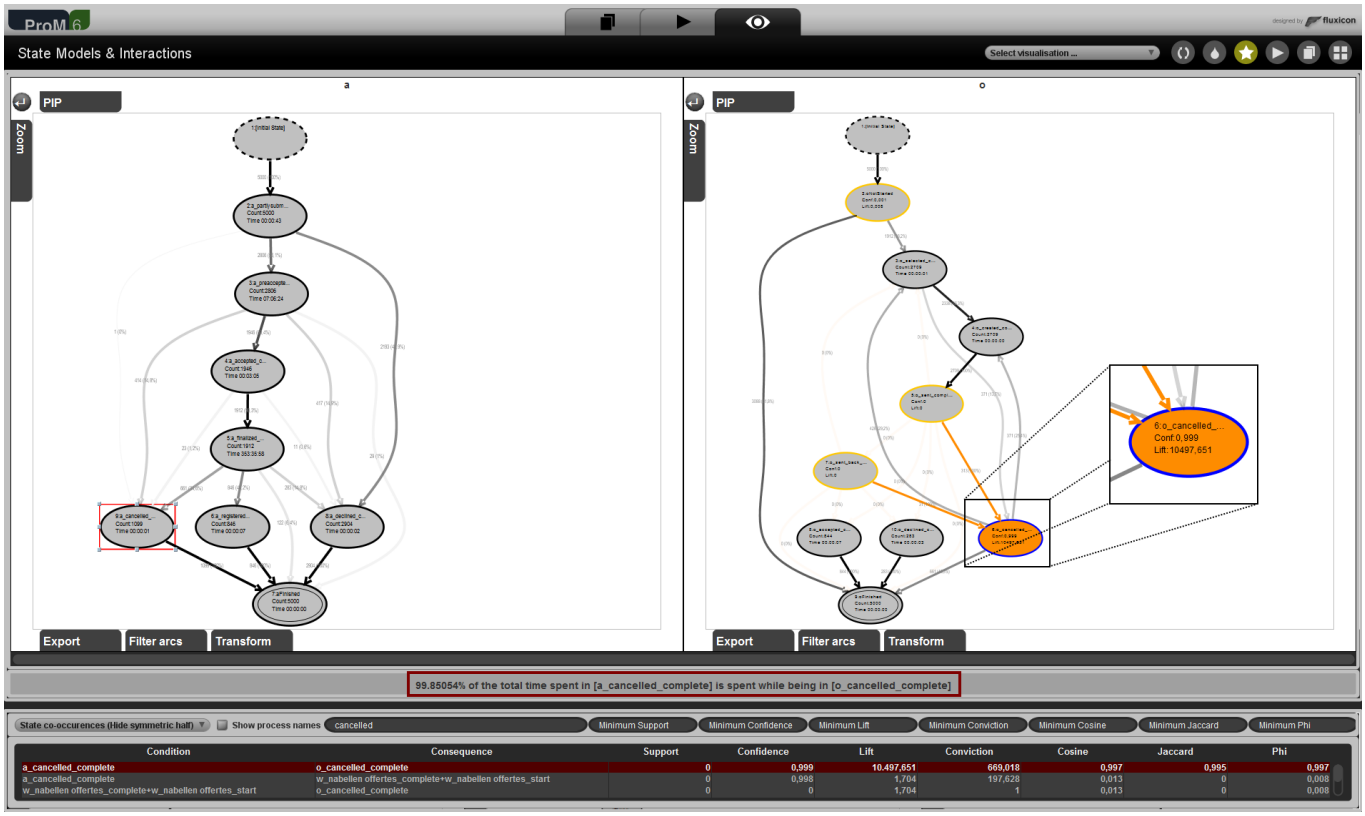

Fig. 4: The analysis guidance is shown below the process models. Users can sort and filter on the different measures of interestingness, and then click on an artifact interaction to highlight it. The highlighted interaction is also presented as a textual interpretation.

A. Process Description

The event data was taken from the BPI Challenge 2012 [13]. This dataset concerns process instances of a personal loan and overdraft application process at a Dutch financial institute. The events and activities in the log are related to three interrelated sub-processes, which can be considered as interacting process artifacts. The first artifact concerns the state of the application (*A*-states), the second relates to the work-items performed by the financial institute (*W*-states), and the third concerns the state of a potential offer that the institute can make to the applicant (*O*-states). This process has been analysed in several other papers [14], [15].

The overall process behavior is as follows. The process starts with the submission of the application. An unlogged check determines whether the application is pre-accepted or declined immediately. The application is accepted once all necessary information has been provided to complete the application. After the acceptance, the institute sends a concrete offer for the terms of the loan or overdraft to the applicant. When the response is returned, the application is validated and then accepted or declined. At any point in the process the applicant can decide to cancel their application and exit the process. In cases where the applicant does not respond in a timely manner, or if the application does not meet the criteria of the financial institute, then the application can be declined by the institute. In exceptional cases the financial institute checks the applications for fraud.

B. Results

The data of the above process was mined by the CSM Miner and then analysed by looking at the measures of interestingness. We present a list of the top artifact interactions for several of the measures from Section IV and explain their relevance for understanding the process behaviour. Such lists can be obtained in the tool by sorting on the desired measure.

Table II shows five examples of state co-occurrences with high conf scores. There are several state co-occurrences that have a conf score of 1, indicating that a given artifact state always co-occurs with a single state in another artifact. Not all of these are shown here because most are the result of the offer artifact not changing state from *o::notStarted* until after the application has been accepted. In general, the state co-occurrences with a high conf score indicate relations between artifact states that match the expected flow of the process as also described in other work [14]. For example, if the loan is activated then the offer has been accepted by the customer ($a::activated \Rightarrow_S o::accepted$), and if the application is approved then the application has been validated ($a::approved \Rightarrow_S w::validation.end$). Another example is that the financial institute only contacts the cus-

TABLE II: Top conf State Co-occurrence.

Condition	Consequence	conf
a::accepted	o::notStarted	1
w::processLeads.start	o::notStarted	1
a::activated	o::accepted	1
a::approved	w::validation.end	0.998
w::followupOffers.start	o::sent	0.986

TABLE III: Top 5 supp State Co-occurrence.

Condition	Consequence	supp
a::finalized	o::sent	0.657
a::finalized	w::followupOffers.end	0.577
w::followupOffers.end	o::sent	0.504
a::preaccepted	o::notStarted	0.191
w::completeApplication.end	o::notStarted	0.175

TABLE IV: Top 5 lift State Co-occurrence.

Condition	Consequence	lift
a::declined	o::declined	596207
a::cancelled	o::cancelled	10498
a::declined	w::processLeads.start	834
w::validation.start	o::declined	819
a::declined	w::fraudDetection.start	626

TABLE V: Top convic State Co-occurrence.

Condition	Consequence	convic
w::validation.end	o :sentBack	10.4
a::accepted	w::completeApplication.end	10.0
a::preaccepted	w::completeApplication.end	7.61
a::activated	w::validation.end	7.54
w::callIncompleteFiles.start	o::sentBack	6.70

tomer to follow-up on an offer after the offer has been sent (*w::followupOffers.start* $\Rightarrow_S$ *o::sent*). State co-occurrences highlighted with high conf scores can be compared to concurrent dependencies between events or activities in traditional process mining.

By contrast, a high supp measure indicates the co-occurring artifact states where a lot of time is spent. Table III shows the five pairs of artifact states with the highest supp scores; note that supp is a symmetric measure so condition and consequence are interchangeable. These results show that almost two thirds of the average time spent in a loan application is spent waiting for the customer to respond after the application has been finalised (*a::finalized* $\Rightarrow_S$ *o::sent*). During this period some time is spent calling the customer, but most of it is spent in between follow-ups (*a::finalized* $\Rightarrow_S$ *w::followupOffers.end*). Additionally, this measure shows that around 20% of the average total time is spent completing the application before an offer is sent out (*a::preaccepted* $\Rightarrow_S$ *o::notStarted*). These imbalances indicate a potential bottleneck at the customer. This shows that an initial overview of this measure can point out performance issues and encourage a process analyst to do a more thorough process performance and bottleneck analysis. The insights also match results from other process analyses [14], [15].

A top five of most interesting artifact interactions according to the lift measure is shown in Table IV. lift shows relations between artifacts that are statistically strong in the sense that they occur much more often than would be expected by chance under independence assumptions. These results indicate that there are different reasons for an application to be declined (*a::declined*): the offer may have been declined (*o::declined*), the lead may not have matched the required criteria (*a::declined* $\Rightarrow_S$ *w::processLeads.start*), or fraud may have been discovered (*a::declined* $\Rightarrow_S$ *w::fraudDetection.start*). It also highlights the synchronisation between the cancellation of the application and the offer (*a::cancelled* $\Rightarrow_S$ *o::cancelled*), as lift is a symmetric. Although the results are not surprising given the process description and semantic understanding of the state names, lift does provide understanding of the strongest connections between artifacts.

Similar to lift, convic also provides an overview of strong relations between artifacts, but this measure is asymmetric in condition and consequence. Table V shows several relations with high conviction. We have omitted relations that have even higher convic scores but that were also highlighted by the other measures. Again, the results show relations that are consistent with other analyses [14]. For example, given that the application has been validated we know that the offer must have been sent back, and given that the application has been accepted or preaccepted we know that the customer must have provided information to complete the application.

The top results for state co-occurrence in terms of cosine, jaccard and ϕ generally score high on at least one other measure. The exact order of the artifact interactions differs between the measures, but in general the state co-occurrence relations that are scored as most interesting are those that have a strong link to the overall behaviour of the application process.

Table VI shows several transitions that always co-occur with the application state *a::finalized*. This means that these transitions, such as the creation and sending of an offer, are only enabled if the application has been finalized ((*o::created,o::sent*) $\Rightarrow_T$ *a::finalized*), i.e. if all the required information has been provided. In general, there are many trivial transition co-occurrences that have a conf of 1, which means there are clear synchronization points in the interaction between the artifacts. Other examples are related to the start of the process that only involves the application artifact.

There are many transition co-occurrences with high lift metric scores due to the clear synchronisation between artifacts. Table VII shows a number of these, with a minimum support of 0.001 to filter out patterns that are the result of very rare transitions. Especially the strong links between

TABLE VI: Top conf Transition Co-occurrence.

Condition	Consequence	conf
from o::selected, to o::cancelled	a::finalized	1
from o::selected, to o::created	a::finalized	1
from o::created, to o::sent	a::finalized	1
from w::completeApplication.end, to w::followupOffers.start	a::finalized	1
from w::fraudDetection.end, to w::validation.start	a::finalized	1

TABLE VII: Top lift Transition Co-occurrence.

Condition	Consequence	lift
from o::sent, to o::declined	a::declined	130
from o::sent, to o::accepted	a::approved	54.0
from o::sent, to o::accepted	w::callIncompleteFiles.start	18.3
from w::followupOffers.end, to w::validation.start	o::sentBack	6.88
from o::sentBack, to o::cancelled	w::callIncompleteFiles.start	4.90

the outcome of the application and the state of the offer are very clear again. Interestingly, there are transitions from the sending of the offer directly to its acceptance, without receiving a reply to the offer (*o::sentBack*). The lift measure shows that these transitions co-occur significantly often while calling the customer for incomplete information ((*o::sent,o::accepted*) $\Rightarrow_T$ *w::callIncompleteFiles.start*). This shows that it appears that the institute also allows the offer to be verbally accepted by customers during contact by phone. Also, a significant number of offers that were sent back and then cancelled were cancelled during contact by phone ((*o::sentBack,o::cancelled*) $\Rightarrow_T$ *w::callIncompleteFiles.start*). These observations are not immediately clear when looking at the control flow using traditional approaches [14], [15]

The above discussion shows that the presented approach is able to highlight artifact interactions that provide insights into the behaviour of a real life process. The insights obtained are comparable with those provided by traditional process mining approaches, but they do not require an analysis of the control flow of a complex or unstructured process model. Sorting and filtering functionalities ensure that the size of the list of potentially interesting artifact interactions remains manageable. However, there are often interactions that score well on multiple measures and it currently remains up to the user to identify the overlap between the top scoring interactions for two or more measures.

VII. Conclusion & Future Work

In this paper we have presented an approach to objectively quantify the interestingness of interactions between artifacts in artifact-centric processes. This approach is based on measures of interestingness that have been defined in the context of process models. It highlights useful or surprising artifact interactions and thereby enables process analysts to deal with large or complex models. The approach has been implemented using an interactive process discovery tool, the CSM Miner, which has been shown to provide relevant insights on real life process execution data. Most of the insights discussed can also be obtained with traditional process mining techniques, but they require data preprocessing to obtain structured models and careful analysis of the behaviour of those complex models.

We aim to extend this work in several ways. The current evaluation is limited and provides only an indication of the usefulness of the approach in practice. We plan to conduct a user study to relate the objective measures of interestingness to the subjective interests of process analysts. Controlled experiments could also provide indications for cut-off or minimal values for the measures.

Extensions of the approach itself are also possible. Instead of only looking at pairs of artifacts, we can generalise artifact interaction to sets of artifacts, similar to association rule learning. In contrast to association rule learning, infrequent relations may also be interesting when analysing a process. There is also room to improve the transformation of execution sequences into observations of artifact interaction. For example, correlations based on time intervals could be used to handle noise or non-fitting executions in the process data.

References

[1] W. M. P. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.

[2] M. L. van Eck, N. Sidorova, and W. M. P. van der Aalst, "Discovering and exploring state-based models for multi-perspective processes," in *Business Process Management - 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings*, 2016, pp. 142–157.

[3] W. M. P. van der Aalst, P. Barthelmess, C. A. Ellis, and J. Wainer, "Proclets: A framework for lightweight interacting workflow processes," *Int. J. Cooperative Inf. Syst.*, vol. 10, no. 4, pp. 443–481, 2001.

[4] V. Popova, D. Fahland, and M. Dumas, "Artifact lifecycle discovery," *Int. J. Cooperative Inf. Syst.*, vol. 24, no. 1, 2015.

[5] X. Lu, M. Nagelkerke, D. van de Wiel, and D. Fahland, "Discovering interacting artifacts from ERP systems," *IEEE Trans. Services Computing*, vol. 8, no. 6, pp. 861–873, 2015.

[6] V. Popova and M. Dumas, "Discovering unbounded synchronization conditions in artifact-centric process models," in *Business Process Management Workshops - BPM 2013 International Workshops, Beijing, China, August 26, 2013, Revised Papers*, 2013, pp. 28–40.

[7] P. Tan, V. Kumar, and J. Srivastava, "Selecting the right objective measure for association analysis," *Inf. Syst.*, vol. 29, no. 4, pp. 293–313, 2004.

[8] B. Liu, W. Hsu, S. Chen, and Y. Ma, "Analyzing the subjective interestingness of association rules," *IEEE Intelligent Systems*, vol. 15, no. 5, pp. 47–55, 2000.

[9] M. L. van Eck, N. Sidorova, and W. M. P. van der Aalst, "Composite state machine miner: Discovering and exploring multi-perspective processes," in *Proceedings of the BPM Demo Track 2016 Co-located with the 14th International Conference on Business Process Management (BPM 2016), Rio de Janeiro, Brazil, September 21, 2016.*, 2016, pp. 73–77.

[10] R. P. J. C. Bose, H. M. W. E. Verbeek, and W. M. P. van der Aalst, "Discovering hierarchical process models using prom," in *IS Olympics: Information Systems in a Diverse World - CAiSE Forum 2011, London, UK, June 20-24, 2011, Selected Extended Papers*, 2011, pp. 33–48.

[11] J. D. Weerdt, S. K. L. M. vanden Broucke, J. Vanthienen, and B. Baesens, "Active trace clustering for improved process discovery," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2708–2720, 2013.

[12] D. L. Bazaldua, R. S. Baker, and M. O. S. Pedro, "Comparing expert and metric-based assessments of association rule interestingness," in *Proceedings of the 7th International Conference on Educational Data Mining, EDM 2014, London, UK, July 4-7, 2014*, 2014, pp. 44–51.

[13] B. F. van Dongen, "Bpi challenge 2012," 2012. [Online]. Available: http://dx.doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f

[14] A. D. Bautista, L. Wangikar, and S. M. K. Akbar, "Process mining-driven optimization of a consumer loan approvals process - the BPIC 2012 challenge case study," in *Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, 2012, pp. 219–220.

[15] A. Adriansyah and J. C. A. M. Buijs, "Mining process performance from event logs," in *Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, 2012, pp. 217–218.