Looking for Meaning: Discovering Action-Response-Effect Patterns in Business Processes

Jelmer J. Koorn[1], Xixi Lu[1], Henrik Leopold[2,3], and Hajo A. Reijers[1]

[1] Utrecht University, Utrecht , The Netherlands
j.j.koorn|x.lu|h.a.reijers@uu.nl
[2] Khne Logistics University, Hamburg, Germany
[3] Hasso Plattner Institute, University of Potsdam, Germany
henrik.leopold@the-klu.org

Abstract. Process mining enables organizations to capture and improve their processes based on fact-based process execution data. A key question in the context of process improvement is how responses to an event (action) result in desired or undesired outcomes (effects). From a process perspective, this requires understanding the action-response patterns that occur. Current discovery techniques do not allow organizations to gain such insights. In this paper we present a novel approach to tackle this problem. We propose and formalize a technique to discover action-response-effect patterns. In this technique we use well-established statistical tests to uncover potential dependency relations between each response and its effects on the cases. The goal of this technique is to provide organizations with processes that are: (1) appropriately represented, and (2) effectively filtered to show meaningful relations. The approach is evaluated on a real-world data set from a Dutch healthcare facility in the context of aggressive behavior of clients and the responses of caretakers.

Keywords: Effect Measurement · Process Discovery · Healthcare · Patterns.

1 Introduction

The desire to improve organizational processes has led to the adoption of process mining in many industries [11,23]. One of the key advantages of process mining is that it enables organizations to understand, analyze, and improve their processes based on process execution data, so-called event logs. Such event logs capture how organizational processes are actually executed and can be extracted from various information systems that are used in organizations [1].

While the advantages of process mining have been demonstrated in many domains, the application of process mining is still associated with different challenges. One particularly important challenge is to provide the user with a process representation that a) is easy to understand and b) allows the user to obtain the required insights about the process execution. To this end, various process discovery algorithms have been proposed, including the heuristic miner [27], the fuzzy miner [15], and the inductive miner [17]. What all of these algorithms have in common is that they focus on discovering the control flow of a process, i.e., the order constraints among events.

In many scenarios, however, the control flow perspective is not sufficient for understanding and improving the process. As an example, consider the care process of a residential care facility supporting clients with their daily needs. The main goal of this process is to ensure the well-being of clients. One of the main factors negatively affecting the well-being of clients are incidents of aggressive behavior, e.g. when clients verbally or physically attack other clients or staff. Staff responds to aggressive incidents with one or multiple measures ranging from verbal warnings to seclusion. A key question in the context of process improvement is which of these measures lead to desired (i.e., de-escalation of aggressive behavior) or undesired (i.e., escalation of aggressive behavior) outcomes.

From a process perspective, this requires understanding the *action-response-effect* patterns. In the healthcare process, we consider the aggressive incidents as *actions*, the countermeasures taken to the incident as *responses*, and the follow-up incidents as *effects*. Action-response-effect patterns are not accounted for in existing discovery algorithms. As a result, their application to such event logs leads to a process representation that is either hard to read (because it contains too many connections) or it does not allow the user to obtain actual insights about the process (because it does not show the effect of behavior).

Recognizing the limitation of existing algorithms with respect to showing meaningful insights into action-response-effect patterns, we use this paper to propose a novel discovery technique. We leverage well-established statistical tests to analyze event logs in order to discover simplified graphical representations of business processes. We simplify the resulting models by highlighting the statistically significant dependency relations according to statistical tests, while insignificant relations are hidden. We conduct an evaluation with an event log from a Dutch residential care facility containing a total of 21,706 aggression incidents related to 1,115 clients. We show that our technique allows to obtain important insights that existing discovery algorithms cannot reveal.

The rest of the paper is organized as follows. Section 2 describes and exemplifies the problem of discovering *action-response-effect* patterns. Section 3 introduces the formal preliminaries for our work. Section 4 describes our proposed technique for discovering *action-response-effect* patterns. Section 5 evaluates our technique by applying it to a real-world data set. Section 6 discusses related work and Section 7 concludes the paper.

2 Problem statement

Many processes contain action-response-effect patterns. As examples consider healthcare processes where doctors respond to medical conditions with a number of treatments, service processes where service desk employees respond to issues with technical solutions, and marketing processes where customers may

EID	CID	Timestamp	action	Response(s)
1	1	12-05 09:53	VA	Warning
2	1	13-05 13:35	PO	Distract Client, Seclusion
3	1	26-05 09:32	VA	Warning
4	1	26-05 11:02	PP	Distract Client
5	2	21-06 14:51	VA	Distract Client
6	1	23-06 21:23	VA	Distract Client
7	2	24-06 17:02	VA	-
8	3	29-08 11:22	VA	Warning
9	3	31-08 08:13	PO	Warning, Seclusion
10	3	31-08 10:48	PP	Distract Client

Legend: EID = Event identifier, CID = Client identifier, VA = Verbal Aggression, PP = Physical Aggression (People), PO = Physical Aggression (Objects),

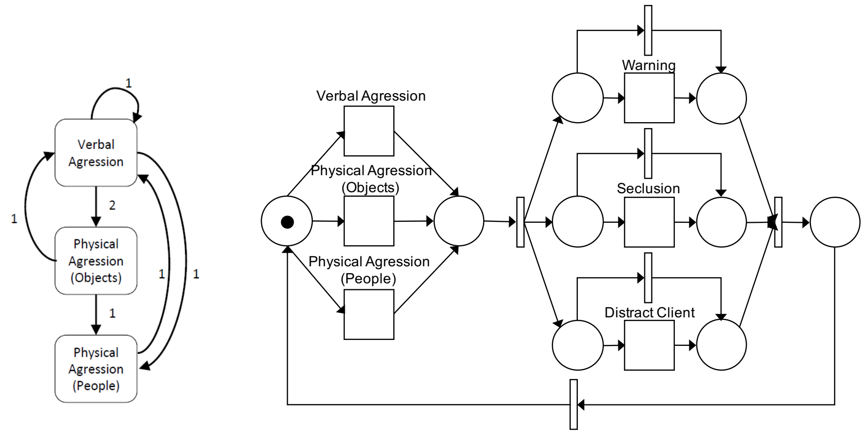
Table 1: Excerpt from an action-response log of a care process

respond to certain stimuli such as ad e-mails with increased demand. Let us reconsider the example of the healthcare process in a residential care facility in order to illustrate the challenge of discovering an understandable and informing process representation from an event log containing action-response relations. The particular aspect of interest are incidents of aggressive behavior from the clients and how these are handled by staff. Table 1 shows an excerpt from a respective event log. Each entry consists of an event identifier EID (which, in this case, is equal to the incident number), a case identifier CID (which, in this case, is equal to the client identifier), a timestamp, an aggressive incident (action), and one or more responses to this event.

Figure 1 a) shows the directlyfollows-graph that can be derived from the events of this log. It does not suggest any clear structure in the process. Although this graph is only based on twelve events belonging to three different event classes, it seems that almost any behavior is possible. In addition, this representation does not provide any insights into certain hidden patterns [2]. However, if we take a closer look, we can see that there are effects to a certain response. For instance, we can see that over time the aggressive incidents related to client 1 escalate from verbal aggression to physical aggression against objects and people. The verbal aggression event in June (EID = 6) is probably unrelated to the previous pattern since it occurs several weeks after. To gain an even deeper understanding, we need to take both the response and its effect into account. Both client 1 and 2 escalate from verbal aggression to physical aggression after the verbal aggression was only countered with a warning.

These examples illustrates that explicitly including the responses and effects in the discovery process is important for answering the question of how to possibly respond to an action when a certain effect (e.g. de-escalating aggressive behavior) is desired. Therefore, our objective is to discover a model that: (1) shows the action-response-effect process, and (2) reveals the dependency patterns of which responses lead to a desired or undesired outcomes (effect). There are two main challenges associated with accomplishing this:

1. *Graphical representation*: From a control-flow perspective, action-response relations are a loop consisting of a choice between all actions and a subse-



(a) Directly-follows graph (b) Petri net

Fig. 1: Representations resulting from the action-response log from Table 1

quent and-split that allows to execute or skip each responses. Figure 1 b) illustrates this by showing the Petri net representing the behavior from the log in Table 1. Obviously, this representation does not allow to understand which responses lead to a desired or undesired effect.

2. *Effective filtering mechanism*: The possible number of responses calls for a filtering mechanisms that allows to infer meaningful insights from the model. In the example above, we only have three event classes and three event response classes (plus the "no response"). This results in eight possible responses. In case of 5 response event classes, we already face 32 (=2^5 possible responses. Including all these response arcs in a model will likely lead to an unreadable model that does not allow to infer the desired insights.

In the next sections, we propose a technique that creates graphical representations of dependency patterns in action-response effect logs.

3 Preliminaries

In this section, we formalize the concept of action-response-effect event logs.

Definition 1 (Action-Response-Effect Log). *Let $\mathcal{E}$ be the universe of event identifiers. Let $\mathcal{C}$ be the universe of case identifiers. Let $d_1, ..., d_n$ be the set of attribute names (e.g., timestamp, resource, location). Let A be the set of actions and R a finite set of responses. An action-response log L is defined as $L = (E, \pi_c, \pi_l, \pi_r, \pi_{d_1}, ..., \pi_{d_n}, <)$, where*

- *$E \subseteq \mathcal{E}$ is the set of events,*
- *$\pi_c : E \to \mathcal{C}$ is a surjective function linking events to cases,*

ID	Timestamp	Action	Response(s)	Effect
1	12-05 09:53	VA	Warning	PO
1	13-05 13:35	PO	Distract Client, Seclusion	τ
1	26-05 09:32	VA	Warning	PP
1	26-05 11:02	PP	Distract Client	τ
2	21-06 14:51	VA	Distract Client	VA
1	23-06 21:23	VA	Distract Client	τ
2	24-06 17:02	VA	-	τ
3	29-07 11:22	VA	Warning	PO
3	31-07 08:13	PO	Warning, Seclusion	PP
3	31-07 10:48	PP	Distract Client	τ

Legend: VA = Verbal Aggression, PP = Physical Aggression (People), PO = Physical Aggression (Objects)

Table 2: Excerpt of the event log action-response-effect

- $\pi_l : E \rightarrow A$ *is a surjective function linking events to actions,*
- $\pi_r : E \rightarrow 2^R$ *is a surjective function linking events to a set of responses,*
- $\pi_{next} : E \rightarrow C$ *is a surjective function linking events to the effects,*
- $\pi_{d_i} : E \rightarrow \mathcal{U}$ *is a surjective function linking the attribute d_i of each event to its value,*
- $< \subseteq E \times E$ *is a strict total ordering over the events.*

Given an action-response log L according to Definition 1, we shall use the shorthand notation $\sigma = \langle e_1, \ldots, e_n \rangle$ in the remainder of this paper to refer to an event trace that consists of n events with an identical case identifier. Furthermore, for any pair of events e_i and e_j with $i < j$, it holds that $e_i < e_j$ according to the strict total ordering of the events in log L.

The set of response events $\{r_1^e, \ldots, r_n^e\}$ of an event e is given by the function π_r, we write $\pi_r(e) = \{r_1^e, \ldots, r_n^e\}$. For each trace $\sigma = \langle e_1, \ldots, e_n \rangle$, the sequence of responses is $\langle \pi_r(e_1), \ldots, \pi_r(e_n) \rangle$. For example, in the action-response log listed in Table 1, for event e_1: $\pi_c(e_1) = 1$ is the case of event e_1, $\pi_l(e_1) =$ "Verbal Aggression" is the action of e_1, and $\pi_r(e_1) = \{$"Warning"$\}$ is the set of responses of e_1.

Effects of Responses. As we discussed, we aim to investigate whether a certain response to an action has an effect on the follow-up event. As such, we measure the effectiveness of a response to an action by studying the effect. For this aim, we first define the effects of events by using function π_{next} and introducing parameter ϵ for elapsed time. For each trace $\sigma = \langle e_1, ..., e_n \rangle$, we define the effect for each e_i, where $1 \leq i < n$ as follows: if the elapsed time to the next event e_{i+1} is less than ϵ, the effect $\pi_{next}(e_i)$ of e_i is the action of e_{i+1}, else we say that the effect is a silent action τ. Formally, if $\pi_{time}(e_{i+1}) - \pi_{time}(e_i) \leq \epsilon$, then $\pi_{next}(e_i) := \pi_{action}(e_{i+1})$, else $\pi_{next}(e_i) := \tau$.

To test the hypothesis whether an effect is independent of the response to an action, the number of observed events is compared to the number of expected events of different responses and effects. To calculate the number of observed events, we create a matrix (table) where each cell is filled with the number of observed events of a response and an effect. Let $a \in A$ be an action, $R = \{r_1, \cdots, r_m\}$ be a set of responses, and $C = \{c_1, \cdots, c_n\}$ a set of effects. We define a $|R| \times |C|$ matrix, where each row represents a response r_i, each column represents an effect c_j, and each cell counts the number of observed events that have response r_i and effect c_j. We have

$$freq_{a,R,C} = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,n} \end{pmatrix}$$

where

$$f_{i,j} = freq_L(a, r_i, c_j) = \quad |\{e \in L \mid \pi_l(e) = a \wedge r_i \in \pi_r(e) \wedge \pi_{next}(e) = c_j\}| \quad (1)$$

For instance, given a log L as listed in Table 2, $freq_L$(“VA”, “Warning”, “PO”) $=|\{e_1, e_8\}|= 2$. Considering Table 3 and omitting the column totals and row totals, it exemplifies a matrix $freq_{a,R,C}$. If the effects are independent of responses, then we should observe that the distribution of effects of a response is similar to the *total distribution*.

Each row r_i presents the distribution of effects $c_1, ..., c_k$ to the response r_i. To test whether each individual response r_i has an influence on the effects, we define $freq_{a,r,C}$ as a $2\times$ $|C|$ matrix:

$$freq_{a,r,C} = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,n} \end{pmatrix}$$

where $f_{1,j} = freq_L(a, r, c_j)$ and

$$f_{2,j} = \quad |\{e \in L \mid \pi_l(e) = a \wedge r \notin \pi_r(e) \wedge \pi_{next}(e) = c_j\}| \quad (2)$$

An example of $freq_{a,r,C}$ where r is “Terminate contact” is listed in Table 4. In the following section, our approach first performs a chi-squared test which allows us to calculate the expected values and test the dependency between responses and effects. The chi-square test compares the observed frequencies to the expected frequencies. If they differ significantly, then the null hypothesis is rejected, which means we cannot rule out that there is a dependency relation between the response and the effect.

The complete event logs containing action-response-effect are used in the technique proposed in this paper. The next section elaborates on this.

4 Discovery technique for action-response logs

In this section, we propose an algorithm to implement a discovery technique, see Algorithm 1. This technique builds on the formalization introduced previously. The goal is to create understandable process models that provide the user with the required insights into the execution of the process. First, we describe the preprocessing that needs to take place (Input for Algorithm 1). Then, we elaborate

Observed	**PO**	**PP**	**VA**	τ	*Total*
Warning	250	400	200	50	*900*
Held with force	20	50	50	10	*130*
Seclusion	30	50	20	10	*110*
Terminate contact	100	100	90	10	*300*
Distract client	100	150	40	10	*310*
Total	*500*	*750*	*400*	*100*	**1750**
Expected	**PO**	**PP**	**VA**	τ	*Total*
Warning	257.1	385.7	205.7	51.4	*900*
Held with force	37.1	55.7	29.7	7.4	*130*
Seclusion	31.4	47.1	25.1	6.3	*110*
Terminate contact	85.7	128.6	68.6	17.1	*300*
Distract client	88.6	132.9	70.9	17.7	*310*
Total	*500*	*750*	*400*	*100*	**1750**

Legend: VA = Verbal Aggression, PP = Physical Aggression (People), PO = Physical Aggression (Objects)

Table 3: Excerpt of the tables used to perform high-level statistical tests; horizontal categories: effect, vertical categories: response

Observed	**PO**	**PP**	**VA**	τ	*Total*
Terminate contact = 0	300	500	210	90	*1100*
Terminate contact = 1	100	100	90	10	*300*
Total	*400*	*600*	*300*	*100*	*1400*
Expected	**PO**	**PP**	**VA**	τ	*Total*
Terminate contact = 0	314.3	471.4	235.7	78.6	*1100*
Terminate contact = 1	85.7	128.6	64.3	21.4	*300*
Total	*400*	*600*	*300*	*100*	*1400*

Legend: VA = Verbal Aggression, PP = Physical Aggression (People), PO = Physical Aggression (Objects)

Table 4: Excerpt of the tables for an individual response used to perform statistical tests; horizontal categories: effect

on the technique which consists of three main stages: (1) high-level statistics (line 1-5 in Algorithm 1), (2) detailed statistics (line 5-9), and (3) identifying influential points (line 9-14).

4.1 Pre-processing the Event Log

We first pre-process the log to obtain the effects of responses. As we are studying the effect of a response to an action, the duration between a response and its effect influences the likelihood of a dependency relation between the two. Let us return to our example: if there is an aggressive incident, there is a given response to this incident. However, if the next incident takes place after a long time (e.g. a year) we doubt that this new incident is still dependent on the response to the initial action. Thus, we defined the parameter epsilon (ϵ), see Section 3. ϵ represents the maximum duration between two events in which the first event is still considered to have an effect on the second event. For our specific example we define ϵ equaling seven days in line with the input of an expert. Based on

Algorithm 1 Compute graph

Input: Event log L
Output: Graph $G = (V, \prec)$
1: *{STAGE 1: High-Level Statistics}*
2: **for** $a \in A$ **do**
3: Initiate matrix $O[a] \leftarrow freq_{a,R,C}$ *{see Equation 2, calculate the observed values}*
4: Compute matrix $E[a]$ *{calculate the expected values by following the chi-square test, see [9]}*
5: Compute $\chi^2_a = \frac{(O[a]-E[a])^2}{E[a]}$ *{To test the dependence between responses R and effects $A \cup \{\tau\}$}*
6: **if** χ^2_a is significant **then**
7: *{$O[a]$ differs from $E[a]$, thus responses R have a statistically significant influence on the effects C}*
8: *{STAGE 2: Detailed Statistics}*
9: **for** response $r \in R$ **do**
10: Compute matrix $O[a]_r$, $E[a]_r$, and $\chi^2_{a,r}$
11: **if** $\chi^2_{a,r}$ is significant **then**
12: *{STAGE 3: Influential Points}*
13: Compute adjusted standardized residuals ASR_c *{see Sec. 4.4}*
14: **for** effect $c \in A \cup \{\tau\}$ **do**
15: **if** ASR_c is significant **then**
16: *{draw the arc from r to c}*
17: $V \leftarrow V \cup \{a_s\} \cup \{r\}$, $\prec \leftarrow \prec \cup \{(a_s, r)\} \cup \{(r, c)\}$
18: **end if**
19: **end for** *{effect}*
20: **else**
21: *{$\chi^2_{a,r}$ is insignificant, i.e., r has no significant influence on C. We do not draw node r or any arc from r to C}*
22: **end if**
23: **end for** *{response}*
24: **else**
25: *{Observed $O[a]$ follows the expected values $E[a]$, thus response R has no statistically significant influence on the effects C; thus, no arcs are drawn}*
26: **end if**
27: **end for** *{action}*
28: **return** G

the ϵ, we introduce state τ. It represents the state we reach if there is no next incident within the defined duration of ϵ. In Table 2 we can see, for example, that distracting the client seems to be related to τ.

4.2 Computing High-Level Statistics

After pre-processing the event log, we investigate for each action the significant relation between the responses and the effects. In our example, the client shows a certain type of aggressive behavior (the action). Given this, we are interested in how the response of a caretaker to that incident has an effect on the follow-up incident. Hence, we will explain the technique with a fixed initial action.

In Table 3, an example of the observed and calculated expected frequencies can be found given the action is physical aggression against objects (see line 2 & 3 in Algorithm 1). This allows us to perform a Pearson Chi-square test [9] (see line 4). Based on a confidence level α (usually 95%), the calculated χ^2 is compared to the Chi-square distribution to see if there is at least one pair of response-effect significantly different. If the chi-square score is insignificant, the action is excluded from the graphical representation (see line 22). If the Chi-square is significant (see line 5), this indicates that the effects may depend on the response. We then move to the second stage, see Sec. 4.3.

We demonstrate this first stage by applying it to a designed example based on our case study presented in Table 3. Based on the observed values, we can calculate the expected values in the table, for example, the expected value for the first cell: response *Terminate Contact* and effect $VA = \frac{\mathrm{N_r} \times \mathrm{N_c}}{E[\mathrm{a}][\mathrm{r}][\mathrm{c}]} = \frac{300 \times 400}{1750} = 68.6$. We know from Table 3 that there are five response classes and four effect classes, so the degrees of freedom: $c = (5-1) \times (4-1) = 12$. Given all this, we can calculate the Chi-square score for the overall table: $\chi^2_c = \sum_{i=1}^{5} \frac{(O_{\text{Warning,PO}} - E_{\text{Warning,PO}})^2}{E_{\text{Warning,PO}}} + \ldots + \frac{(O_{\text{Distract client},\tau} - E_{\text{Distract client},\tau})^2}{E_{\text{Distract client},\tau}}$ $\chi^2_{12} = \frac{(250-257.1)^2}{257.1} + \ldots + \frac{(10-17.7)^2}{17.7} = 63.47$. Now we need to determine if this score is significantly different from the mean of the Chi-distribution [14]. The formula for calculating the p-value is complex and will thus not be discussed in detail in this paper. For more details we refer to [14]. In our case the p-value (< 0.001) corresponding to our Chi-square score is significant. This shows that for at least one pair of response-effect given action PO there is a significant difference from the expected frequency. Thus, we perform a Chi-square test for each individual response.

4.3 Computing Detailed Statistics

In the second stage of the algorithm, we perform the Chi-square test again on each response class to determine for which response we need to perform post-hoc statistical tests (see line 6 - 8 in Algorithm 1). For this purpose we create dummy variables. A dummy variable is made for each individual response, which takes the value of 0 or 1. The new table we create is a 2 x 4 table where the rows represent the response either taking a 0 or 1 value, see Table 4. Note that the degrees of freedom changes to three now. The same formulas are used to calculate the individual response Chi-square score and the corresponding p-value. A Bonferroni correction [16] is made to correct the critical value for the fact that on the same table multiple sets of analyses are performed. The Chi-square test identifies for which responses there is at least one effect that is significantly different from the expected frequency. If the Chi-square score is significant, we create a node for the response and perform post-hoc tests to identify the exact pairs of response-effect that are significant (see line 9).

We will demonstrate this stage on our designed example. We test five times (one for each response). Thus, we apply the Bonferroni correction [16] on confidence level of 95% (meaning $\alpha = 0.05$): $\frac{0.05}{5} = 0.01$. If we take Table 4, we can use the same formulas as presented in the previous section to calculate the expected values. Note that we assume independence of responses. Thus, if there are two responses, the action is counted twice: once for response 1 and once for response 2. Therefore, the observed frequencies in Table 3 are not necessarily equal to those in Table 4. If we perform the Chi-square test for the response *terminate contact* we get a Chi-square score of 31.96 with a p-value < 0.001. Thus, for the response *terminate contract* there is at least one effect that is significantly different from the expected frequency. A post-hoc test will identify the exact pairs for which this is true.

4.4 Identifying Influential Points

In the last stage, the post-hoc tests are performed to test which exact pairs of response-effect have a significant contribution to the Chi-square test score. For this, the adjusted standardized residuals (ASR) [3] are calculated (see line 10 in Algorithm 1). They represent a normalization of the residuals (observed - expected frequency). As the residuals can take either a positive or negative value we use two-sided testing. In order to improve the interpretability, we transform the α level into a critical value. We refer to [14] for details on this. If $|ASR| > criticalscore$ the difference between observed and expected frequency is significant. A significant score means that a specific pair of response-effect has a significant impact on the overall test score. We will refer to these as *influential points*. If the score is insignificant, no arc is drawn for that pair of response-effect

For each influential point, arcs are drawn in the graphical representation (see line 11-14). We first draw an arc from the action to the responses. On this arc, we indicate the observed frequency of the behavior. Then, we draw an arc from the response to the effect(s) for which we found a significant relation. On the arc we display the observed frequency followed by the expected frequency in brackets. If the observed frequency is larger than the expected frequency, i.e. the response leads to an increase in frequency of effect, we draw a thick arc. Correspondingly, if the observed frequency is lower than the expected frequency we draw a thin arc. The total number of graphical representations created equals the number of actions for which a significant Chi-square score is found (see line 25).

Now, we turn to the designed example. From the previous section we know that the response *Terminate contract* results in a significant Chi-square score. To calculate which points are influential points we calculate the adjusted standardized residuals for each pair. To exemplify, we show the calculation of the ASR for the pair *Terminate contact = 1* and *VA*: $ASR = \frac{90-64.3}{\sqrt{64.3*(1-\frac{64.3}{300})*(1-\frac{64.3}{300})}} = 4.08$ Given our Bonferroni correction gave us an alpha of 0.01 (see previous section), we need to test on the 99 % confidence level. The critical absolute value for this is 2.57. Thus, if our ASR value is $> |2.57|$ we mark it as influential point and draw an arc in the graphical representation. In the example of the pair Terminate contract = 1 and VA the ASR is larger than the critical score ($4.08 > 2.57$). Therefore, we draw a thick arc in the graphical representation of this example.

After conducting the above-described calculations for all actions, responses, and effects from the designed example, we obtain a total of three graphical representations (one for each action). In the next section, we evaluate the technique by applying it on a real-world data set.

5 Evaluation

The goal of this section is to demonstrate the capability of our technique to discover models that allow to obtain meaningful insights into action-response-

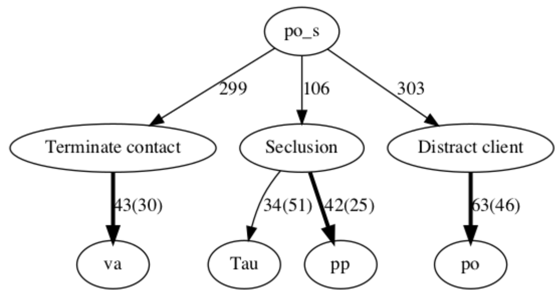


Fig. 2: Graphical representation of applying our technique on the action-response-effect event log. The initial action is physical aggression against objects.

effect patterns. To this end, we implemented our technique in Python and applied it to real-world data set. The scripts are publicly available for reproducibility[4].

5.1 Data set

To evaluate our technique, we use a real-world data set related to the care process of a Dutch residential care facility. The event log contains 21,706 recordings of aggressive incidents spread over 1,115 clients. The process captured in this log concerns the aggressive behavior of clients in their facilities and the way client caretakers respond to these incidents. In the log we can find an aggressive incident of the client, which fits in one of five action classes. This is followed by some measures taken by the staff as response to this incident, which fits in one of nine response classes. In line with the description of our technique, we transformed this log into suitable triples by adding the next aggressive incident of a client as the effect, given it took place within our ϵ. Thus, the effect can be one of five classes. As there are four different classes of actions, our technique will return four different graphical representations. Below, we present and discuss the results for one class of action: physical aggression against objects.

5.2 Results

Healthcare Case Results. After applying our technique to the data set, we obtain four graphs (one for each class of action). In Figure 2 we show the resulting graph when the initial action is physical aggression against objects ("po_s" in the figure). What we can see in the graph is the observed frequencies of the responses. For example, *terminate contact* has been observed 299 times in our data as a response to physical aggression against objects. Following this, the graph shows that in 43 events the effect to this response class is verbal aggression. From the data we know that, in total, there are four action classes, nine response classes,

[4] Source code and results: github.com/xxlu/ActionEffectDiscovery.

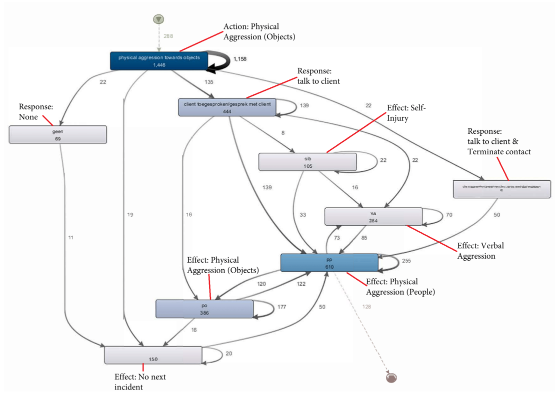


Fig. 3: Directly-follows process model of the real-world event log for the initial action physical aggression against objects. This shows the process filtered on 5% of the possible activities and paths. The model is created with Disco [5].

and five effect classes. As such, the representation for one action could potentially contain 81 $(9 * 4 + 9 * 5)$ arcs. In our graphical representation, we do not draw all these arcs, we only draw seven of them.

Each arc represents a significantly higher (thick arc) or lower (thin arc) amount of observed compared to expected frequencies of interactions between the response and effect. As can be seen in the graph, this reduces the number of arcs substantially such that the impact of each individual response to a physical aggression against objects event can be studied.

Focusing on the insights we can obtain from the graphical representation in Figure 2. The figure shows that responding to a physical aggression against objects event with *seclusion* results in a significantly higher amount of physical aggression against people ("pp" in the figure). This can be seen by the thicker arc or by comparing the observed frequency (42) with the expected frequency (25). Studying the frequencies we can conclude that we observe that the response *seclusion* is almost 1.7 times as likely to have the effect equaling physical aggression against people compared to what is expected. In similar fashion, the response *terminate contract* and *distract client* lead to a higher likelihood of one class of effect. However, the response *seclusion* leads to a significantly lower likelihood of the effect being no next aggression incident (τ).

Comparison to Control-Flow Based Discovery. Figure 3 illustrates that a control-flow based discovery approach, such as the directly-follows approach, cannot provide such insights in the context of action-response-effect logs. The process model contains a large number of arcs. The number of arcs here increase exponentially with the number of responses observed. A possible solution to this could be to add information to the control-flow based representation, such as the observed frequencies of the arcs or nodes. However, filtering based on the frequencies does not always have the desired result. This can also be seen in

Figure 3. It could even be misleading since the data set is imbalanced. In this real-world scenario, a high frequency does not imply a significant pattern. This becomes obvious if we compare the approaches. From the figures we can see that none of the significant response-effect pairs from Figure 2 are displayed in Figure 3. In order to understand the relations in the representation, we have to account for the relative frequencies. These reveal the meaningful insights that are hidden in the representation of a discovery technique such as the control-flow. Hence, even after applying filtering mechanisms, Figure 3 does not provide the insights that are required to answer a question such as: If a client displays aggressive behavior of class X, which response is likely to lead to an (de-)escalation or future aggression?

5.3 Discussion

Insights. The key question identified at the start of this research addressed the desire to express insights into how a response to an action can lead to a desired or undesired outcome (effect). In our problem statement we identified two main challenges associated with this that need to be overcome: (1) graphical representation, and (2) effective filtering mechanism. Studying the example of aggressive behavior highlights how the proposed technique addresses both these challenges. Figure 2 shows that our technique creates simple graphical representation that allow for insights into dependency relations that cannot be obtained using Figure 3. In addition, comparing the same figures we can see that the use of statistics reduces the number of arcs substantially. The filtering mechanisms is effective in the sense that it filters those arcs that are meaningful, opposed to those that are merely frequent.

As a result in Figure 2 we can see the effects of responses to aggressive incidents. Important to note is that physical aggression against people is seen as the most and verbal aggression as the least severe form of aggressive behavior. The figure shows that responding to a physical aggression against objects event with *seclusion* increases the likelihood of the next event being physical aggression against people. In other words, this response leads to an undesired outcome: escalation of the aggressive behavior of the client. In contrast, we observe that the response *terminate contact* is more likely to lead to a verbal aggressive incident. Thus, this represents a desired outcome: de-escalation of future violence. Finally, the response *distract client* has the effect that the client is more likely to repeat the same class of action ("po" in the figure), indicating a circular relation.

Implications. One interesting implication of our technique is that the generated insights can be used to support decision making processes. In our example, Figure 2 can be used to train existing and new staff members to ensure that appropriate responses are taken. Placing this technique in a broader medical context, the technique could help make informed decisions when different treatment options are considered. In a different domain, the technique could help a marketing organization understand the effectiveness of marketing strategies in terms of response of potential customers. In short, the discovery technique provides insights into action-response-effect patterns where the objective of analyzing the process is to understand possible underlying dependency patterns.

Limitations. It is worth mentioning that, in our technique, we assume the independence of the responses. This means that each response has a unique effect on the effect and there is no interfering effect when responses are combined. For example, if response r_1 is more likely to lead to c_1 and r_2 to c_2, then performing r_1 and r_2 are more likely to lead to follow-up effects c_1 or c_2, but not a different effect c_3. Statistical pre-tests can be performed to verify this assumption. A basic approach is to create a correlation matrix for the dummies of the responses. In our example, this matrix shows that the assumption holds. In other words, no responses are strongly and significantly correlated. If the assumption is violated then the technique should consider R' as input. R' is a set of all independent classes including those groups of responses that have a potential interfering effect.

6 Related Work

Over the last two decades a plethora of process discovery algorithms were proposed [5]. The majority of these approaches generate procedural models such as Petri nets [25,26], causal nets [20,28], BPMN models [4,7] or process trees [8,17]. Some approaches also discover declarative models [6,24] or hybrid models (i.e. a combination of procedural and declarative models) [10,19]. What all these techniques have in common is that they aim to discover the control flow of a business process, that is, the execution constraints among the process activities. Our approach clearly differs from these traditional process discovery approaches by focusing on action-response patterns instead of the general control flow.

There are, however, also alternative approaches to process discovery. Most prominently, several authors addressed the problem of artifact-centric process discovery [18,21,22]. The core idea of artifact-centric process discovery is to consider a process as a set of interacting artifacts that evolve throughout process execution. The goal of artifact-centric discovery, therefore, is to discover the lifecyles associated with these artifacts and the respective interactions among them. While artifact-centric discovery techniques move away from solely considering the control-flow of the process' activities, the main goal is still control-flow oriented. A related, yet different approach to process discovery was proposed in [12,13]. This approach focuses on the different perspectives of a process and discovers and captures how their relations using Composite State Machines.

While the technique from [12,13] is potentially useful in many scenarios we address with our technique, the insights that can be obtained with our technique differ substantially. The technique from [12,13] allows to understand how different artifact lifecycle states are related. For example, it reveals that a patient in the state "*healthy*" does no longer require a "*lab test*". The goal of our technique is to show what actually needs to be done (or should not be done) to make sure a patient ends up in the state "*healthy*". To the best of our knowledge, we are the first to propose a technique that discovers such action-response-effect patterns and allows the reader to develop an understanding of why certain events occur.

7 Conclusion

This paper presented a technique to discover action-response-effect patterns. We identified two main challenges that we addressed in this research: (1) graphical representation, and (2) effective filtering mechanism. In order to address these challenges, we proposed a novel discovery technique that builds on filtering influential relations using statistical tests. We evaluated our technique on a real-world data set from the healthcare domain. More specifically, we used our technique to study aggressive behavior and show that we can gain valuable and novel insights from the representations discovered by our technique. The representations also show that the technique can tackle both challenges by providing an easy-to-interpret representation that only displays meaningful relations.

In future work, we plan to further test the approach on real-world cases. In addition, we plan to extend this work in two ways: (1) by introducing more complex statistical tests to provide flexibility in the assumption of independence of the responses, and (2) by introducing statistical tests to approximate the optimal configuration of ϵ.

Acknowledgments This research was supported by the NWO TACTICS project (628.011.004) and Lunet Zorg in the Netherlands. We would also like to thank the experts from the Lunet Zorg for their valuable assistance and feedback.

References

1. Van der Aalst, W.: Data science in action. In: Process Mining, pp. 3–23. Springer (2016)
2. van der Aalst, W.M.: A practitioners guide to process mining: Limitations of the directly-follows graph. Procedia Computer Science **164**, 321 – 328 (2019)
3. Agresti, A.: Categorical data analysis, vol. 482. John Wiley & Sons (2003)
4. Augusto, A., Conforti, R., Dumas, M., La Rosa, M.: Split miner: Discovering accurate and simple business process models from event logs. In: 2017 IEEE International Conference on Data Mining (ICDM). pp. 1–10. IEEE (2017)
5. Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Maggi, F.M., Marrella, A., Mecella, M., Soo, A.: Automated discovery of process models from event logs: Review and benchmark. IEEE TKDE **31**(4), 686–705 (2018)
6. Bernardi, M.L., Cimitile, M., Di Francescomarino, C., Maggi, F.M.: Using discriminative rule mining to discover declarative process models with non-atomic activities. In: International Symposium on Rules and Rule Markup Languages for the Semantic Web. pp. 281–295. Springer (2014)
7. vanden Broucke, S.K., De Weerdt, J.: Fodina: A robust and flexible heuristic process discovery technique. decision support systems **100**, 109–118 (2017)
8. Buijs, J.C., van Dongen, B.F., van der Aalst, W.M.: A genetic algorithm for discovering process trees. In: 2012 IEEE Congress on Evolutionary Computation. pp. 1–8. IEEE (2012)
9. Cochran, W.G.: The $\chi 2$ test of goodness of fit. The Annals of Mathematical Statistics pp. 315–345 (1952)
10. De Smedt, J., De Weerdt, J., Vanthienen, J.: Fusion miner: Process discovery for mixed-paradigm models. Decision Support Systems **77**, 123–136 (2015)

11. De Weerdt, J., Schupp, A., Vanderloock, A., Baesens, B.: Process mining for the multi-faceted analysis of business processes—a case study in a financial services organization. Computers in Industry **64**(1), 57–67 (2013)
12. van Eck, M.L., Sidorova, N., van der Aalst, W.M.: Discovering and exploring state-based models for multi-perspective processes. In: International Conference on Business Process Management. pp. 142–157. Springer (2016)
13. van Eck, M.L., Sidorova, N., van der Aalst, W.M.: Guided interaction exploration in artifact-centric process models. In: 2017 IEEE 19th Conference on Business Informatics (CBI). vol. 1, pp. 109–118. IEEE (2017)
14. Fisher, R.A., Yates, F.: Statistical tables: For biological, agricultural and medical research. Oliver and Boyd (1938)
15. Günther, C.W., Van Der Aalst, W.M.: Fuzzy mining–adaptive process simplification based on multi-perspective metrics. In: International conference on business process management. pp. 328–343. Springer (2007)
16. Haynes, W.: Bonferroni Correction, pp. 154–154. Springer New York, New York, NY (2013)
17. Leemans, S.J., Fahland, D., van der Aalst, W.M.: Discovering block-structured process models from event logs-a constructive approach. In: International conference on applications and theory of Petri nets and concurrency. pp. 311–329. Springer (2013)
18. Lu, X., Nagelkerke, M., van de Wiel, D., Fahland, D.: Discovering interacting artifacts from erp systems. IEEE Transactions on Services Computing **8**(6), 861–873 (2015)
19. Maggi, F.M., Slaats, T., Reijers, H.A.: The automated discovery of hybrid processes. In: International Conference on Business Process Management. pp. 392–399. Springer (2014)
20. Nguyen, H., Dumas, M., ter Hofstede, A.H., La Rosa, M., Maggi, F.M.: Mining business process stages from event logs. In: International Conference on Advanced Information Systems Engineering. pp. 577–594. Springer (2017)
21. Nooijen, E.H., van Dongen, B.F., Fahland, D.: Automatic discovery of data-centric and artifact-centric processes. In: International Conference on Business Process Management. pp. 316–327. Springer (2012)
22. Popova, V., Fahland, D., Dumas, M.: Artifact lifecycle discovery. International Journal of Cooperative Information Systems **24**(01), 1550001 (2015)
23. Rojas, E., Munoz-Gama, J., Sepúlveda, M., Capurro, D.: Process mining in healthcare: A literature review. Journal of Biomedical Informatics **61**, 224 – 236 (2016)
24. Schönig, S., Rogge-Solti, A., Cabanillas, C., Jablonski, S., Mendling, J.: Efficient and customisable declarative process mining with sql. In: International Conference on Advanced Information Systems Engineering. pp. 290–305. Springer (2016)
25. Song, W., Jacobsen, H.A., Ye, C., Ma, X.: Process discovery from dependence-complete event logs. IEEE Transactions on Services Computing **9**(5), 714–727 (2015)
26. Verbeek, H., van der Aalst, W.M., Munoz-Gama, J.: Divide and conquer: A tool framework for supporting decomposed discovery in process mining. The Computer Journal **60**(11), 1649–1674 (2017)
27. Weijters, A.J., Van der Aalst, W.M.: Rediscovering workflow models from event-based data using little thumb. Integrated Computer-Aided Engineering **10**(2), 151–162 (2003)
28. Yahya, B.N., Song, M., Bae, H., Sul, S.o., Wu, J.Z.: Domain-driven actionable process model discovery. Computers & Industrial Engineering **99**, 382–400 (2016)