

# Towards Operator 4.0, Increasing Production Efficiency and Reducing Operator Workload by Process Mining of Alarm Data

Gyula Dörgő<sup>a</sup>, Kristof Varga<sup>a</sup>, Máté Haragovics<sup>b</sup>, Tibor Szabó<sup>b</sup>, János Abonyi<sup>a,\*</sup>

<sup>a</sup>MTA-PE "Lendület" Complex Systems Monitoring Research Group, University of Pannonia, Egyetem u. 10, H-8200 Veszprém, Hungary

<sup>b</sup>MOL Danube Refinery, Olajmunkás u. 2, H-2443 Százhalombatta, Hungary  
[janos@abonyilab.com](mailto:janos@abonyilab.com)

A methodology to extract temporal patterns of alarm sequences and operator actions from the log files of alarm management systems is proposed. Firstly, time-segments that are informative from the viewpoint of operator interventions are identified by the algorithm. These segments include series of alarms that initialize operator actions, sets of operator actions, and a period that potentially covers the effects of the corrective actions of the operators. In the second step of the methodology, the sets of operator actions that are frequently applied in the same situations are determined. For this purpose, the FP-Growth Algorithm, which is one of the fastest tools of frequent item-set mining and generates well-structured action trees that are not only suitable for the visualization of interventions but lend themselves to build association rules that could be directly applied in decision support systems, is utilized. Finally, multi-temporal sequence mining is applied to reveal what alarms led to the sets of operator actions and what were the effects of these interventions. The applicability of the methodology is illustrated by presenting results connected to the analysis of the delayed coker plant at the Danube Refinery of the MOL Group.

## 1. Introduction

Plant operators can become overloaded when they are continuously exposed to a large number of alarms. The fundamental question concerning control system design is whether or not providing more information results in better operational decisions (Dadashi et al., 2017). A properly designed and operated alarm system should assist the work of operators by indicating the presence of abnormal situations. Recently, the reduction in operator workload by the rationalization of the number of annunciated process alarms has become a highly studied area of research. To achieve this goal, mostly conventional techniques, like alarm limit deadbands (Adnan et al., 2011), delay-timers (Zang et al., 2015) or filtering (Izadi et al., 2009), have been applied. It is believed that the information inefficiency of alarms results in operators becoming overloaded (Dadashi et al., 2017). The consequences of ignoring the complexity of human behaviour with regard to industrial safety and security have been discussed (Festag and Hartwig, 2016). According to our knowledge, only one publication has dealt with operator actions and alarm notifications together and addresses the problem concerning the effect of the action patterns of operators (Hu et al., 2016).

The challenge discussed in the present paper is twofold: first, we highlight that the sequences of industrial alarms cannot be handled by themselves since the interactions of the operator continuously change the underlying processes and thus the evolution path of the alarms. Following this concept, a novel methodology for the determination of alarms or alarm patterns that require interactions by the operators and lead to the intervention of operators is presented. On the other hand, not only is the work of the operators influenced by the alarms, their actions affect the evaluation of alarm patterns as well. Therefore, different operating strategies are compared by the proposed analysis and are qualified based on their effectiveness, which can ensure a consistent and efficient operation based on the best operational practices. In this way, not only can the workload

of operators be reduced, but the production efficiency can be significantly improved, e.g. energy consumption can be reduced (Challis et al., 2017).

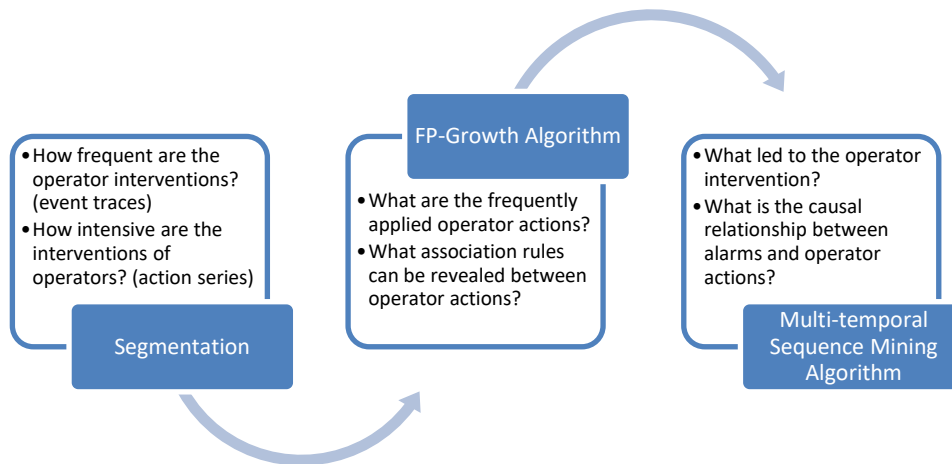


Figure 1: The workflow of the methodology presented for the detection of frequently applied operator actions and their causal relationship to the occurring alarms.

The structure of the paper follows our methodology illustrated in Figure 1. The applied algorithm is presented in the blue text boxes, while the questions to be answered by their application are illustrated in the associated white text boxes. Initially, the alarm and event-log database is segmented into event traces and action series, then operation periods for further investigation are selected. In Section 2, this process is discussed, while in Section 3 the details of the algorithms applied for the extraction of frequent operator actions are presented. In this section, it is also shown that what led to these operator action patterns and their effect on the process can be studied by multi-temporal analysis.

## 2. The structure and segmentation of the alarm and event-log database

In this section the structure of an industrial alarm and event-log database is described and the method for its segmentation proposed. Since frequent item-set and temporal pattern mining algorithms are utilized, a brief mathematical formalization of the temporal patterns of event databases is also provided.

### 2.1 Structure of the alarm and event-log database

Files of industrial alarms and event logs are usually composed of time-stamped events of alarms, operator actions, system messages and any further temporal information. Each event that occurs is labeled with a tag name. Every alarm and operator action can be considered as the states of the process. Each state (denoted by  $s$ ) is represented by  $\langle pv, a \rangle$  data couples, where  $pv$  compresses the tag name of the process variable and all the above-mentioned information, and  $a$  is the attribute showing the type of action on the process variable, for example, low or high alarms in the case of alarms, or open or close in the case of operator actions. An *event* is the time interval in which the defined state occurs, denoted by  $e$ . Each event,  $e$ , of the database is represented by  $\langle s, st, et \rangle$  triplets, where  $s$  is the state that occurs in the time interval between the start time,  $st$ , and the end time,  $et$ . It should be noted that the same attribute of a state can correspond to multiple events as well. This representation is applicable to both frequent item-set mining and pattern mining, which is the core concept of the present paper concerning the determination of operational practices.

The methodology is presented through the analysis of the delayed coker plant at the Danube Refinery of the MOL Group. An insight into how the intensity of alarms and operator actions varies with time in the case of the 25 most frequent tags is given in Figure 2. In the figure, all the numbers, tag names and identifiers are masked due to confidentiality. During the preprocessing of the event-log database, the operator actions were assumed to be point-like events without any temporal extension, while alarms lasted from their announcement to their associated return to normal records. An approximately 4-month-long operational period was analyzed with more than 2000 process tag names so that our example of application can be considered as a realistic and challenging case study.

## 2.2 Segmentation of the alarm and event-log database

Since the operator action patterns and their anticipation and consequential alarms are our main concerns, the alarm and event-log database was filtered for the segments containing operator actions.

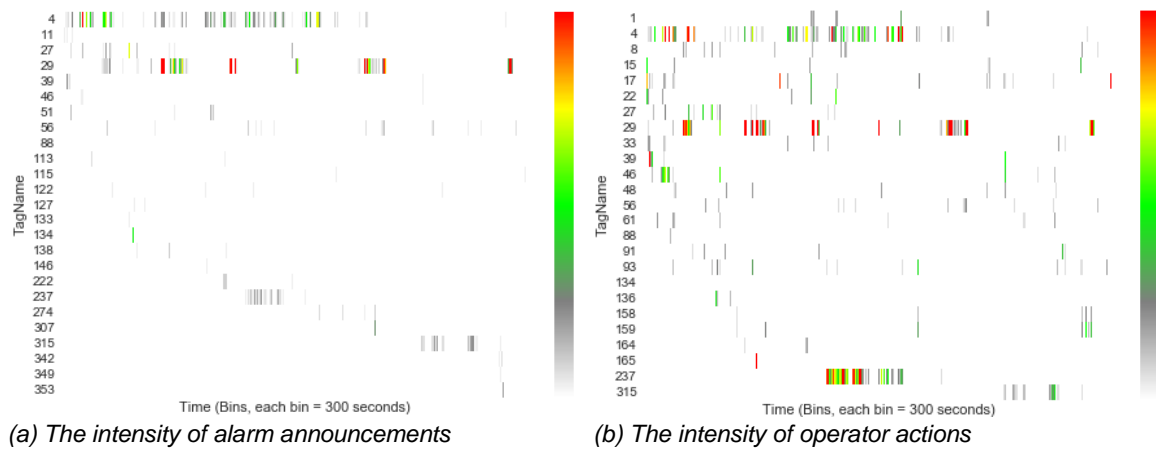


Figure 2: The intensity of alarm announcements (a) and operator actions (b) over a selected time period. The number of alarm announcements and operator actions in the case of the 25 most frequent tags are illustrated.

A schematic representation of the segmentation of the process into event traces is presented in Figure 3. Time is represented by the horizontal line, while the event series with and without operator actions is illustrated by the yellow and blue bars, respectively. The segmentation starts with the detection of well-separated segments of operator actions which are divided into periods that are not comprised of operator actions for longer than a defined *segmentation window*. To investigate the events that lead to the intervention of operators and to see the effect of these actions, a *trace window* based on the time constant of the process was set (green arrow in Figure 3.). Therefore, the time needed for a fault to develop and trigger operator actions and the time required for the corrective actions to drive back the process to its proper operation zone are covered by the window. Based on the expert knowledge of process engineers, to detect causal relationships between alarms and operator actions 240 seconds and 60 seconds seemed to be a preferable segmentation window and trace window for the definition of event traces, respectively. All the events of the alarm and event-log database that fall within the time constraint of the trace window are used to build an *event trace* in our investigations (red arrow in Figure 3.). Inside these event traces, sets of originally point-like operator actions that follow each other closely (within the *action series window*) are assumed to be one *action series* based on one intuition of the operator (yellow arrow in Figure 3.). A reaction time of approximately 20 seconds for chemical process operators was suggested (Buddaraju, 2011). Therefore, a ten-second-long action series window was defined which assumes that actions within this period can be considered as the results of the same stimulus. Based on this assumption these operator actions were aggregated into operator action series and their frequent itemsets were analyzed. Consequently, the remaining periods that do not contain operator actions are referred to as *alarm series*. It should be noted that according to the presented approach multiple action and alarm series can be contained in a single event trace, where the time difference between the operator actions inside the action series is less than the action series window, and between the action series is less than the segmentation window.

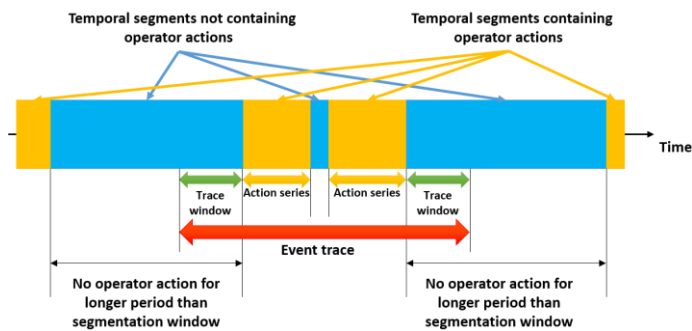


Figure 3: The segmentation of an alarm and event-log database into event traces of alarms and operator actions.

With the presented segmentation approach, more than 5,000 event traces with almost 10,000 operator action series were derived. The originally huge database could be significantly reduced to more informative segments, which is more convenient for further investigations of ours.

### 2.3 Multi-temporal representation of alarm and operator action sequences

The event traces were generated to extract frequently occurring temporal patterns within them. An illustrative representation of the alarms and operator actions of an exemplary event trace is presented in Figure 4. Time is represented by the horizontal axis, while the states of the process are denoted by the rows of the vertical axis. According to the description of the alarm and event-log database, the alarms are temporal events represented by the bars of  $s_{1-3}$ , where the length of the given event is indicated by the horizontal length of the bars. The operator actions are point-like in time, represented by the dotted lines of  $s_{4-5}$ . In order to reveal frequently occurring operation patterns of the process, a *time window* is required. The length of time in which two events are considered to be directly consequential is set by the value of this time window. Given two events,  $e_1$  and  $e_2$ , with the time intervals  $(st_1, et_1)$  and  $(st_2, et_2)$ , respectively, a temporal connection between them can be defined using the following temporal predicates:  $E, B, D$  and  $O$  which stand for the *equal, before, during* and *overlap*, respectively.

By using these temporal predicates, temporal instances can be defined as  $\varphi := e_1 \overset{R}{\Rightarrow} e_2$ , where  $R \in \{E, B, D, O\}$ , for example, in the case of the events of Figure 4:  $e_1 \overset{O}{\Rightarrow} e_2$ ,  $e_2 \overset{B}{\Rightarrow} e_6$ . A pattern of  $k + 1$  states connected by  $k$  temporal predicates, is referred to as a  $k$ -length temporal pattern (the number of temporal predicates is  $k$ ) and is denoted by  $\Phi_k$ . Assuming a temporal pattern, e.g.  $\Phi := s_i \overset{R}{\Rightarrow} s_j$ , and a temporal instance of events,  $\varphi := e_{ip} \overset{R}{\Rightarrow} e_{jq}$ , where the events  $e_{ip}$  and  $e_{jq}$  are related to  $s_i$  and  $s_j$ , respectively, then the temporal pattern  $\Phi$  is referred to as being supported by the temporal instance  $\varphi$ . By following this concept, the probability of a state can be determined, assuming that its probability is proportional to the support of the  $k = 0$ -length sequence as presented in Eq (1), where  $supp(s_i)$  represents the number of supporting events of each state:

$$P(s_i) \simeq supp(s_i) \quad (1)$$

According to the present approach, to characterize the frequency of occurrences of the given state, the support of the pattern, which is the number of temporal instances normalized by the number of temporal instances of the most frequent state, is introduced:

$$support(\Phi) = supp(\Phi)/|E| \quad (2)$$

where  $|E| = \max_{j=1,2,\dots,N}(|E_j|)$ ,  $N$  stands for the number of states in our database,  $E_j$  is the set of events supporting the state  $s_j$ , and  $|E_j|$  represents the number of events in  $E_j$ . For further explanation and a detailed description of the frequent temporal sequence mining algorithm utilized see (Dörgő and Abonyi, 2018).

### 3. Detection of frequent operational strategies and their temporal causality to the process

The aim of this section is to discover how the operators interact with the process through the analysis of the operator actions and action series. First, the detection of frequently applied operator actions is described, and then the causal relationship between alarms and a frequently applied action pair investigated.

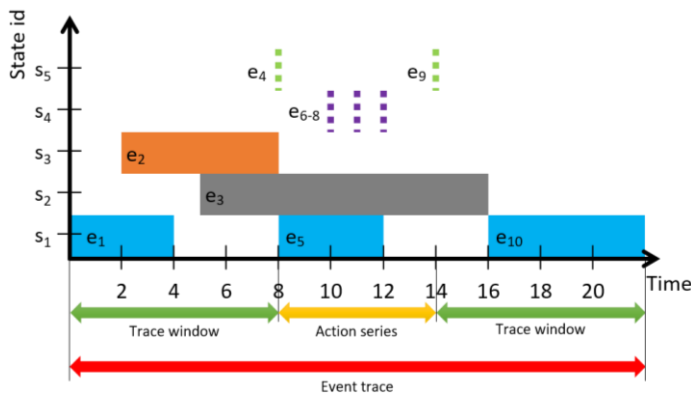


Figure 4: The multi-temporal representation of events. The states are denoted by the rows of the vertical axis. The alarms and operator actions are represented by the vertical bars and dotted lines, respectively.

### 3.1 The detection of frequently applied corrective actions

The incoming alarms of an industrial system are mainly characterized by the physics of the underlying processes, which provide the opportunity for the determination of frequently occurring alarm patterns. However, in the case of operator actions, not only the hectic attitude of each operator, but the presence of actions given by multiple operators can accumulate in random action patterns as well. However, it can be assumed that the corrective actions triggered by a malfunction do not change in a completely random fashion, just the order of execution can be interchanged, providing the opportunity to identify the set of frequently applied operator actions. To find such action sets, a frequent itemset mining algorithm was applied, referred to as the FP-Growth Algorithm (Han et al., 2000). The core idea of the algorithm is the utilization of the frequent-pattern tree (FP-tree) data structure, which is a highly efficient and transparent solution for the representation of association rules. In the following, the structure of the FP-tree and the methodology behind the algorithm focusing on its application in the mining of frequently occurring action sets in action series is discussed.

The tree consists of a root with the subtrees of nodes representing the operator actions. Each node of this tree consists of four fields: the *action-name* for the identification of the action that the node represents, the *support count* to register the number of action series represented by the portion of the path that reaches this node, the *node-link* to the next node in the FP-tree which represents the same operator action, and the *parent-link* to the parent node. The inputs of the FP-Growth Algorithm are the event-log database, in the present context the list of unique actions in each action series, and the support threshold, that is the minimum number of action series in which the given event occurs for it to be considered as a frequently occurring action analogous to the measure presented in Section 2.3 for multi-temporal patterns. The resultant FP-tree is constructed in two runs over the dataset: in the first, the support of each action in the input database is calculated by the algorithm (the number of action series in which the action occurs), which discards the infrequent actions based on the given support threshold and generates a support-descending list of frequent actions. In the second run, each action series is revisited by the algorithm sorting the frequent actions according to the determined support-descending list and attaching the listed actions of the particular action series to the tree starting from its root: either by incrementing the support count of an existing node or by creating a new node with its count initialized to one.

The FP-Growth Algorithm helps to identify the operator actions that frequently occur in action series and provides an easily interpretable visualization of association rules. Based on the results of the FP-Growth Algorithm, frequent actions can be selected for further investigations and the anticipated and following alarms of the action series that contain the selected actions examined.

The results of the mining of the defined action series of the alarm and event-log database of the analysed delayed coker plant are presented in Figure 5. On each node, the code of the tag name is written together with its percentage of occurrence in the action series separated by a colon. The structure is a transparent representation of association rules: by following the nodes starting from the root of the tree, the percentage of occurrences of the operator actions represented by the visited nodes can be determined. Consider the bottom subtree for an exemplary interpretation: 17.06 % of the action series contained operator actions with tag 271 and 9.21 % of the action series contained the tags 271 and 675 altogether. By following this approach, each subtree can be easily interpreted.

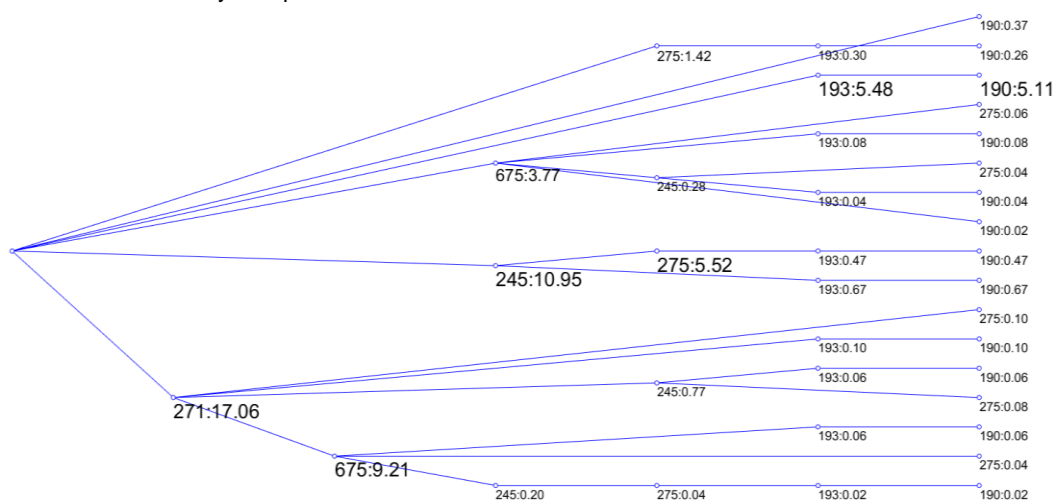


Figure 5: The FP-tree structure of frequent operator actions. The operator actions are represented by the nodes with the tag of the operator action and its percentage of occurrence separated by a colon.

### 3.2 Causal relationship between alarms and operator actions

Instead of the enormous task of analyzing the circumstances of all the operator interventions, which would eventually lead to less informative patterns, a relatively frequently applied action pair based on the FP-tree in Figure 5, namely the action pair 271-675, was selected. By filtering the action series containing both of these actions, the multi-temporal patterns of alarms and operator actions (with 0.5 % of the support threshold) were generated, and the ones containing one item of the action pair selected. In this way nine different temporal connections between operator actions and two between operator actions and alarms were determined. In the case of longer patterns, operator actions which are performed by multiple presses in a row could be detected. This occurs, when an operator sets the new parameters step by step pressing the control panel multiple times in a short time period. This result provides the opportunity to analyse the causal relationship between alarms and operator actions with the incorporation of expert knowledge, and defines logic for automatization and decision support systems based on the conclusions of the analysis.

## 4. Conclusions

Although numerous publications have focused on frequently occurring patterns of alarms, the investigation of operator actions has scarcely been studied in the field of alarm management studies. In the present article, it is highlighted that alarm occurrences cannot be investigated by themselves since alarms and operator actions are in a close causal relationship with each other. The methodology presented in this paper provides a heuristic approach for the segmentation of alarm and event-log databases to obtain the event traces of operator interactions. Since operator interactions can follow random patterns, instead of temporal patterns only the co-occurrence of operator actions using the FP-Growth Algorithm were analysed, and how transparent the visualisation of the resultant association rules demonstrated. Furthermore, with the aggregation of the investigated action series into a single-process event, a multi-temporal frequent sequence mining algorithm was applied to reveal what alarm sequences led to the intervention of operators and what the effect of these actions was. The effectiveness of the proposed methodology is demonstrated through the analysis of the alarm and event-log database of the delayed coker plant at the Danube Refinery of the MOL Group. To stimulate further research, the resultant Python and MATLAB codes of the proposed alarm and event-log analysis algorithms are readily available from the website of the authors ([www.abonyilab.com](http://www.abonyilab.com)).

### Acknowledgments

This research has been supported by the National Research, Development and Innovation Office (NKFIH) through the project OTKA-116674 (Process mining and deep learning in the natural sciences and process development) and the EFOP-3.6.1- 16-2016- 00015 Smart Specialization Strategy (S3) Comprehensive Institutional Development Program. Gyula Dörgő was supported by the ÚNKP-17-3 New National Excellence Program of the Ministry of Human Capacities.

### References

- Adnan N.A., Izadi I., Chen T., 2011, On expected detection delays for alarm systems with deadbands and delay-timers, *Journal of Process Control*, 21, 1318-1331.
- Buddaraju D., 2011, Performance of control room operators in alarm management, Master Thesis, Louisiana State University and Agricultural and Mechanical College, Baton Rouge, USA.
- Challis C., Tierney M., Todd A., Wilson E., 2017, Human factors in dairy industry process control for energy reduction, *Journal of Cleaner Production*, 168, 1319-1334.
- Dadashi N., Golightly D., Sharples S., 2017, Seeing the woods for the trees: the problem of information inefficiency and information overload on operator performance, *Cognition, Technology & Work*, 19, 561-570.
- Dörgő G., Abonyi J., 2018, Sequence mining based alarm suppression, *IEEE Access*, 6, 15365-15379
- Festag F., Hartwig S., 2016, Consequences of ignoring the complexity of human behaviour for industrial safety and security, *Chemical Engineering Transactions*, 48, 919-924.
- Han J., Pei J., Yin Y., 2000, Mining frequent patterns without candidate generation, *SIGMOD Rec.*, 29, 1-12.
- Hu W., Al-Dabbagh A.W., Chen T., Shah S.L., 2016, Process discovery of operator actions in response to univariate alarms, *IFAC-PapersOnLine*, 49, 1026-1031.
- Izadi I., Shah S.L., Shook D.S., Kondaveeti S.R., Chen T., 2009, A frame-work for optimal design of alarm systems, *IFAC Proceedings Volumes*, 42, 651-656.
- Zang H., Yang F., Huang D., 2015, Design and analysis of improved alarm delay-timers, *IFAC-PapersOnLine*, 48, 669-674.