

Article

# Conformance Checking of a Longwall Shearer Operation Based on Low-Level Events

Marcin Szpyrka <sup>1</sup>, Edyta Brzychczy <sup>2</sup>, Aneta Napieraj <sup>2</sup>, Jacek Korski <sup>3</sup>  
and Grzegorz J. Nalepa <sup>1,4,\*</sup>

<sup>1</sup> Department of Applied Computer Science, Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, AGH University of Science and Technology, 30-059 Krakow, Poland; mszpyrka@agh.edu.pl

<sup>2</sup> Department of Economics and Management in Industry, Faculty of Mining and Geoengineering, AGH University of Science and Technology, 30-059 Krakow, Poland; brzych3@agh.edu.pl (E.B.); aneta.napieraj@agh.edu.pl (A.N.)

<sup>3</sup> Famur S.A., 40-698 Katowice, Poland; jkorski@famur.com.pl

<sup>4</sup> Jagiellonian Human-Centered Artificial Intelligence Laboratory (JAHCAI) and Institute of Applied Computer Science, Jagiellonian University, 30-348 Krakow, Poland

\* Correspondence: gjn@gjn.re

Received: 30 October 2020; Accepted: 9 December 2020; Published: 15 December 2020



**Abstract:** Conformance checking is a process mining technique that compares a process model with an event log of the same process to check whether the current execution stored in the log conforms to the model and vice versa. This paper deals with the conformance checking of a longwall shearer process. The approach uses place-transition Petri nets with inhibitor arcs for modeling purposes. We use event log files collected from a few coal mines located in Poland by Famur S.A., one of the global suppliers of coal mining machines. One of the main advantages of the approach is the possibility for both offline and online analysis of the log data. The paper presents a detailed description of the longwall process, an original formal model we developed, selected elements of the approach's implementation and the results of experiments.

**Keywords:** industrial process; sensor data; conformance checking; Petri nets; shearer; underground mining

## 1. Introduction

Intensive digitisation brings new challenges and opportunities for companies and users. Most of them are related to data gathering and processing for the acquisition of potentially new and useful knowledge about processes in enterprise. In the case of industrial companies, the main issue concerns the use of low-level data acquired from monitoring systems. The analysis of this data should ultimately lead to the optimisation of industrial processes and company management procedures. The heterogeneous industrial data are most commonly generated by a range of low-level sensors, monitoring the operation of the machinery involved in the execution of the processes. In order to be practically useful, these data need to be pre-processed and often augmented with certain domain knowledge regarding the process. Moreover, sensor data can be translated into some higher level representation, e.g., event log [1–4].

Furthermore, the proper use of such data understanding of industrial processes often requires the use of computer models of the process. Such models, together with the above-mentioned data, can later be used for the process monitoring and possibly enable its optimisation. There are a number of specific issues, such as detection of anomalies in the execution of the process [5–7] and predictive maintenance of the machinery [8]. More recently, there has been a lot of interest in the so-called digital

twins—realistic computer models are used for the simulation of the process execution in modified conditions, thereby saving costs of field tests [9–11].

A conceptual and practical challenge is related to the selection of the modeling method (language) and approach. Recently, one of the often considered perspectives has been business process management (BPM). It is concerned with designing, monitoring, executing, and evaluating business processes in organisations on a general level. The primary artifact in BPM is a process model, which is an abstract representation of a business process [12], or corresponds to the industrial process, e.g., manufacturing. A business process model could be built manually or be generated based on the pre-processed monitoring data in the form of event logs through the use of process mining (PM) techniques. However, in industrial practice, such an approach is often not feasible.

BPM use various sources of information about the processes. One of them, especially in industrial installation, is Internet of Things (IoT), defined as a network of connected devices collecting and exchanging data over the Internet [13]. Process management requires an adequate amount of information as a base. IoT can complement BPM with continuous data delivery and physical actuation to improve decision making. In changeable market conditions it is not enough to use only historical data from traditional repositories (e.g., databases and data warehouses), but there is a need to use and analyse the up-to-dated data. In this context, IoT data, such as events, delivered via in-memory databases or complex event processing can be useful [13]. On the other hand, it can be pointed that events generated from IoT sensors do not directly correspond to meaningful process activities or process instance, and IoT entities and processes are rarely represented in an explicit way in and through process models [14]. It results in the hindered analysis of conformance between process model and process data.

Commonly the existing industrial installations, well understood by their designers, are augmented with sensors gathering data. Analysing the machinery operation, we have to handle streams of low-level measurements and events (e.g., measures of currents, temperature, oil levels, two-state signals). Thus high-level modelling techniques can not be applied directly. To overcome this issue, one can translate low-level events into high-level activities and analyse process behaviour with the standard conformance checking techniques. However, industrial processes can be complex and highly variable (e.g., there are no two similar traces in the log—the number of process variants can be equal to a number of traces). Moreover, in real-life scenarios, the differences between an event log and a process model can be large [15].

In the case of the analysed process of shearer operation in longwall mining, there is difficulty in creating a prescribed high-level model. There is a theoretical model of shearer operation (described in the following section); however, depending on changeable mining conditions, other behaviour is also allowed (e.g., some repetitions denoted as normal behaviour). Moreover, some activities can exist but are not prescribed in the process model (e.g., moving without cutting). Additionally, the spatial context of the mining process influences the assessment of normal/abnormal behaviour (some of the activities are normal in a specific location of their occurrence). Even though we translate low-level events into high-level events, resultant Petri net remains a spaghetti-like model, thus making conformance checking complicated. It should also be emphasised that in the online settings, to detect an anomaly, each low-level event should be translated in-the-fly into a high-level event, making the whole conformance checking process ineffective.

In this context, our paper presents an original approach that combines numerical analysis of low-level data collected from sensors with formal modelling of the industrial process under consideration. We focus on the model of the most important machinery executing the process. The use of a formal language for the given system representation allows for a better understanding of its features and for defining its behaviour unambiguously. The former advantage is crucial for researchers who want to understand the process in details but are not the corresponding domain experts, the latter is necessary for formal model analysis e.g., using model checking techniques [16]. Our model can be effectively used to validate data from sensors monitoring the machines to detect possible errors in

readings, or miss-operation of the equipment. Considering the used formalism, it can also be used for realistic simulation and data generation.

The developed approach supports process users with better, more complete information about the execution of the process and warnings about deviations from the standard process model. What is crucial, we can receive this information in a short time, enabling a quick response of the dispatcher or the person managing the process. In addition, we provide the synergy of information from various sensors, so far presented in visualisation systems on different charts, into one specific message about irregularities in the process. Until now, the supervisor has had to assemble and pre-process various information (e.g., the shearer location, currents of the drums and haulages and moving direction of the shearer) into identification of the process state and notice deviations.

The paper presents results of research work conducted in the CHIST-ERA Pacmel project. (See the project webpage at <http://PACMEL.geist.re>). The project is oriented toward the development of novel methods of artificial intelligence for knowledge modelling and intelligent data analysis in Industry 4.0. In the project, we are closely collaborating with several industrial companies providing us with expert knowledge regarding the machinery and industrial installations, and large data samples from industrial sensors. In this use case, we cooperate with both the mining company, and with the producer of the industrial machinery. Our partner, Famur S.A. (see the company webpage at <http://www.FAMUR.com.pl>), is one of the global suppliers of mining machines for a number of countries all over the world.

The paper is structured as follows: Section 2 presents a review of relevant related works. Then, Section 3 introduces the main concepts related to the longwall mining process and the main machinery used in it. A general description of our approach considered in this paper is given in Section 4. Section 5 introduces the original formal model of a longwall shearer. The results of the application of our approach for the conformance checking of the shearer operation process are presented in Section 6. The final section contains a summary of our work and presents our future research plans.

## 2. Related Works

It is anticipated that the factories of the future will turn into smart factories with autonomous and adaptive production processes. However, the growing complexity of production process networks is expected to complicate the rapid detection of real-time process anomalies and optimisation of industrial processes.

Detection of process deviation from the model is one of the main tasks of process mining [17], namely, conformance checking. A conformance checking technique's input usually consists of an event log and a process model, and its output includes a set of differences between the model and the log. This set can contain specific metrics or measurements and a verbal description.

One of the key challenges of conformance checking in the industrial setting is to identify relevant event data among a pool of data in IoT systems; then one extracts and transforms the data to the process-centric event log format for process mining techniques. This action requires a great amount of domain knowledge and it mostly involves manual and ad hoc solutions. The challenge mentioned gave rise to a number of solutions and techniques for event log creation, presented, among others, in [14,18–21].

Conformance checking techniques detect two types of deviations between a process model and a log [15]: (1) behaviour observed in the log but disallowed by the model (unfitting behaviour), and (2) behaviour allowed by the model but not observed in the log (additional behaviour). The most recognised conformance checking techniques include token-based replay [22], alignments techniques [23], casual footprints [17] and behaviour profiles [24]. A comprehensive review of the topic is presented in [25].

One of the proposals for online conformance checking is a framework that relies on the notion of behavioural patterns in the process [26]. The proposed approach enables analysis also in the so-called warm start mode (when an analysed case is only a part of real execution). Therefore,

authors accompanied a framework with a corresponding instantiation that builds on top of the notion of Petri net unfoldings. The four-step approach includes the creation of reverse Petri net for a given model; then, unfolded models of original and reverse nets are created; after that, transformation of unfolded models to so-called process model for online conformance (PMOC) occurs (via reachability graphs); and at last, online conformance checking with the use of matrices describing possible relations between pairs of activities in the process model. The evaluation of conformance is based on three metrics: actual conformance, completeness and confidence.

In [27] an approach to computing conformance for online data streams with the replay approach is presented. The authors assumed as input a Petri net, which is converted into a transition system. The transition system is decorated with additional arcs allowing deviation, and a cost function associated with transitions representing deviations. The two-step approach starts with the creation of an augmented model based on region theory and state distance, and cost configuration for deviated behaviour. The second step includes online analysis consisting of replaying the events from the stream on the model and accumulating the costs associated with each execution. All process instances with costs higher than 0 are deviating from the prescribed model.

A distinct approach for conformance checking is presented in [28]. The authors proposed usage of a procedural process model—a workflow net with time series (TSWF-net) in which fulfilment of the process task can be expressed as a set of temporal conditions about the values of the variables in the log. It is the first conformance checking technique considering the evolution of data over time, which is of the most importance for the analysis of the real-time systems wherein process tasks depend on how one or several variables of the process variate.

The above-mentioned techniques support conformance checking with formal process models (Petri nets). However, other models can also be used.

One of the automaton-based techniques is presented in [15]. With this technique, an event log is encoded as sequences of words and compressed into a minimal deterministic acyclic finite state automaton (DAFSA), and the process model is transformed into another automaton (reachability graph—RG). After comparison of two automata, an error-correcting synchronised product automaton (called a partial synchronised product) is created, in which each state is a pair—a state in the RG and a state in the DAFSA. Based on this result, a set of optimal trace alignments is enumerated, or a set of behavioural difference statements can be formulated.

The automaton-based approach is also presented in [29]. In the configurable divide-and-conquer Projected Conformance Checking (PCC) framework, decomposing a log into subsets of activities and partial conformance checking are performed instead of the comparing of complete behaviour over all activities. For each subset, main measures are computed as recall, fitness or precision, and the final measures are provided by the averages over these subsets. The PCC framework handles any formalism which can be expressed in the form of deterministic finite automata (DFAs), including, e.g., BPMN, UML-ADs or Petri nets.

Petri nets, in past years, have been implemented for the modelling and analysis of mining processes (mainly in regard to the operation processes of machinery). In [30] the process model of mechanised roof support operation was presented. A more extensive model of concurrent operation of longwall shearer and mechanised roof supports was presented in [31]. However, these models were not used in the conformance checking task.

### 3. Characteristics of the Mining Process in an Underground Coal Mine

Longwall mining is one of the most important mining techniques used in underground coal mines [32]. Mining of coal is most often carried out in longwall faces which are equipped with a longwall complex, i.e., a set of machines and devices appropriately selected in terms of efficiency and mutual cooperation. The modern mechanised longwall complex consists of several basic elements, such as the cutting machine (a shearer or a plow); an armoured face conveyor (articulated chain

conveyor, AFC); a mechanised roof support; a chain (overhead) conveyor with a crusher (beam stage loader, BSL); a hydraulic pump unit feeding the roof support; and a set of electrical power devices.

The longwall shearer is basically used for coal cutting and loading on the armoured face conveyor. AFC is used as a roadway for the shearer and for coal transport. The coal from AFC goes to BSL; then to a belt conveyor, and by a system of transportation to the main tank; and then to the surface by the vertical transport. In the case of high longwalls, crushers are sometimes used on a shearer in order to crush large lumps of coal. The next part of the longwall complex is mechanised roof support. It is used to prevent the roof from collapsing. From the technical point of view, it is also necessary for adjusting the AFC to the cut surface. Figure 1 presents a view along a longwall face showing the longwall complex with connections between the mechanised roof supports on the right and AFC on the left.

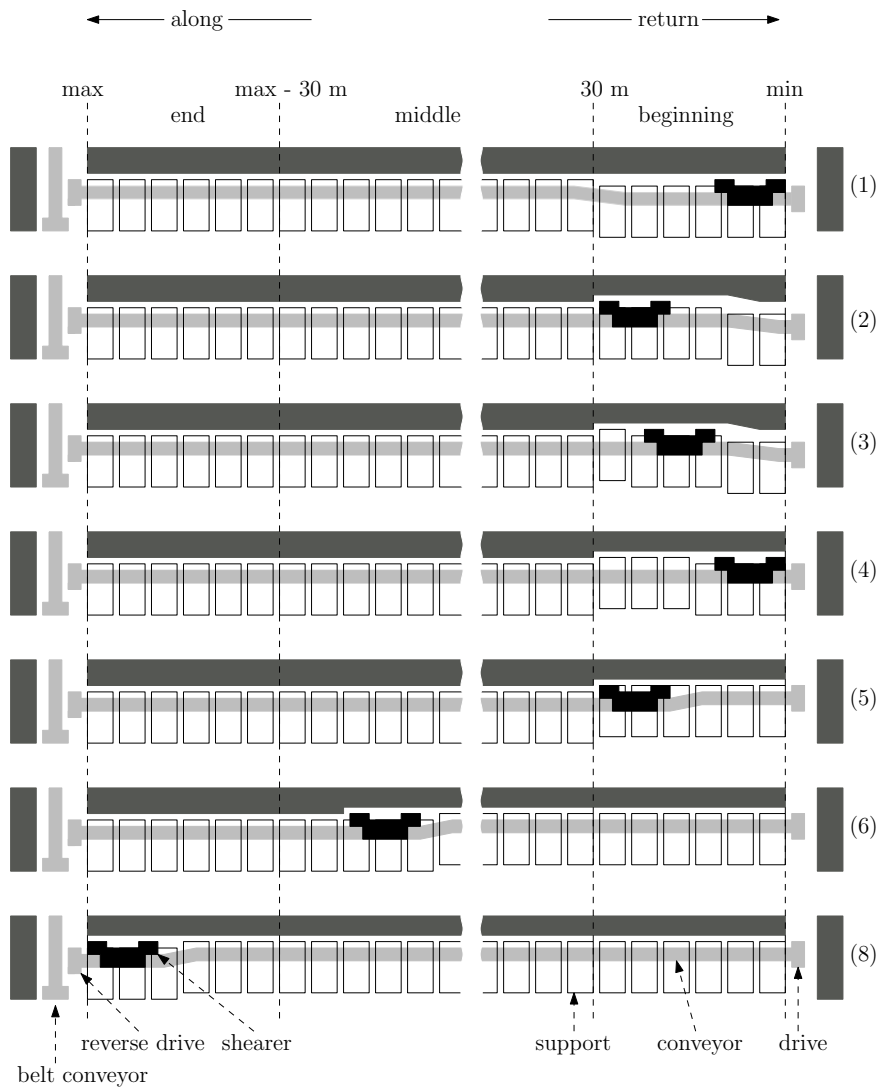


**Figure 1.** Standard longwall complex (source: Famur).

To reduce the model size and the subset of attributes for the paper purposes, we focus on modelling the shearer behaviour only. A shearer consists of a machine body containing electric motors (haulages), hydraulic equipment and controls which are mounted over the AFC. Horizontal cutting drums (organs) are mounted on the face side of the machine, laced with cutting picks and rotate in a plane parallel to the longwall face.

The longwall mining process can be defined as a cyclical implementation of a set of operations (activities) repeated in a specific order and time in a longwall face. The set of these activities depends on technology, equipment and work organisation. However, most often (i.e., in the technology of two-way mining with a shearer) they cover the activities shown in Figure 2 [32,33]. Moving the support and conveyor follows the working shearer. Drive and reverse drive modes are performed at both ends of the longwall face.

The operations performed in the longwall face are monitored by various IT systems. The core information system in an underground mine is the dispatcher system [34]. It collects data from IT systems implemented in a mine, e.g., monitoring of specific mining machinery; telemetric systems monitoring the mine atmosphere and automatic shutdown of electric devices in case of hazards occurrence; and TV monitoring enabling observation of important mining machinery and excavations.



**Figure 2.** General scheme of activities carried out in a longwall face.

The data in the longwall monitoring pipeline are mostly low-level data with different granularity. For the purpose of the shearer modelling, we focus on the subset of attributes presented in Table 1.

**Table 1.** Characteristics of variables for shearer operation description.

Description	Unit	Domain	Attributes
Shearer main status	-	on/off	Status
Currents on organs (drums)	A	(0–1000)	LDC, RDC
Currents on tractors (haulages)	A	(0–200)	LHC, RHC
Shearer move direction	-	true/false	Move_Left, Move_Right
Shearer arms move event	-	true/false	Right_Arm_Up, Right_Arm_Down, Left_Arm_Up, Left_Arm_Down
Shearer speed	m/s	(0–20)	Speed
Shearer location	m	(0–300)	SM_Location
General direction	-	along/return	Direction

Useful variables for shearer process description are: the main status of the shearer, currents on cutting drums and haulages (values of currents are expressed as the averages of three-phase currents), the direction of shearer movement, shearer arm movement, shearer speed, location of the shearer in the longwall and general direction of shearer movement. The abbreviations LDC, RDC, LHC and RHC in the Table correspond to left/right, drum/haulage currents respectively.

Figure 3 presents a description of the ideal shearer cycle with the denotation of general shearer position (beginning, middle and end of the longwall face) and enumerated process stages.

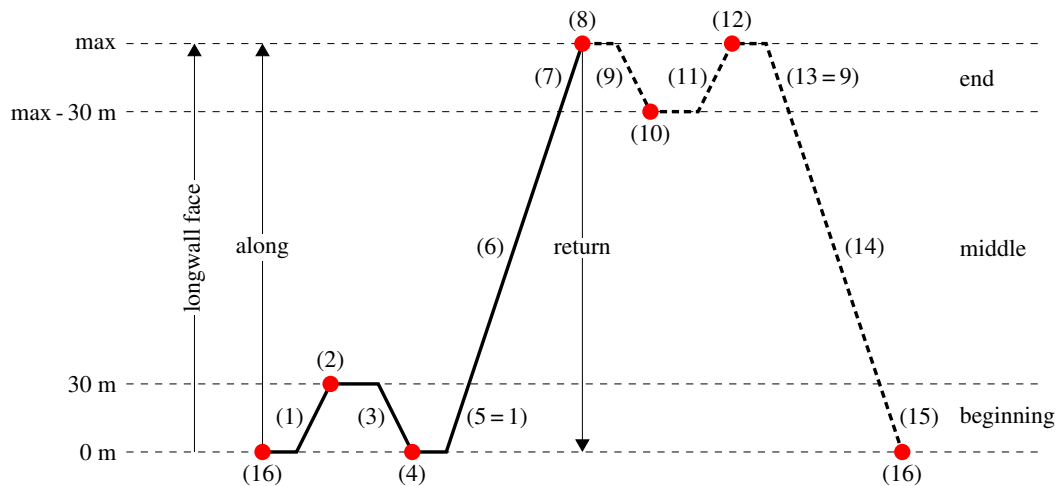


Figure 3. Model of the shearer cycle (see Table 2).

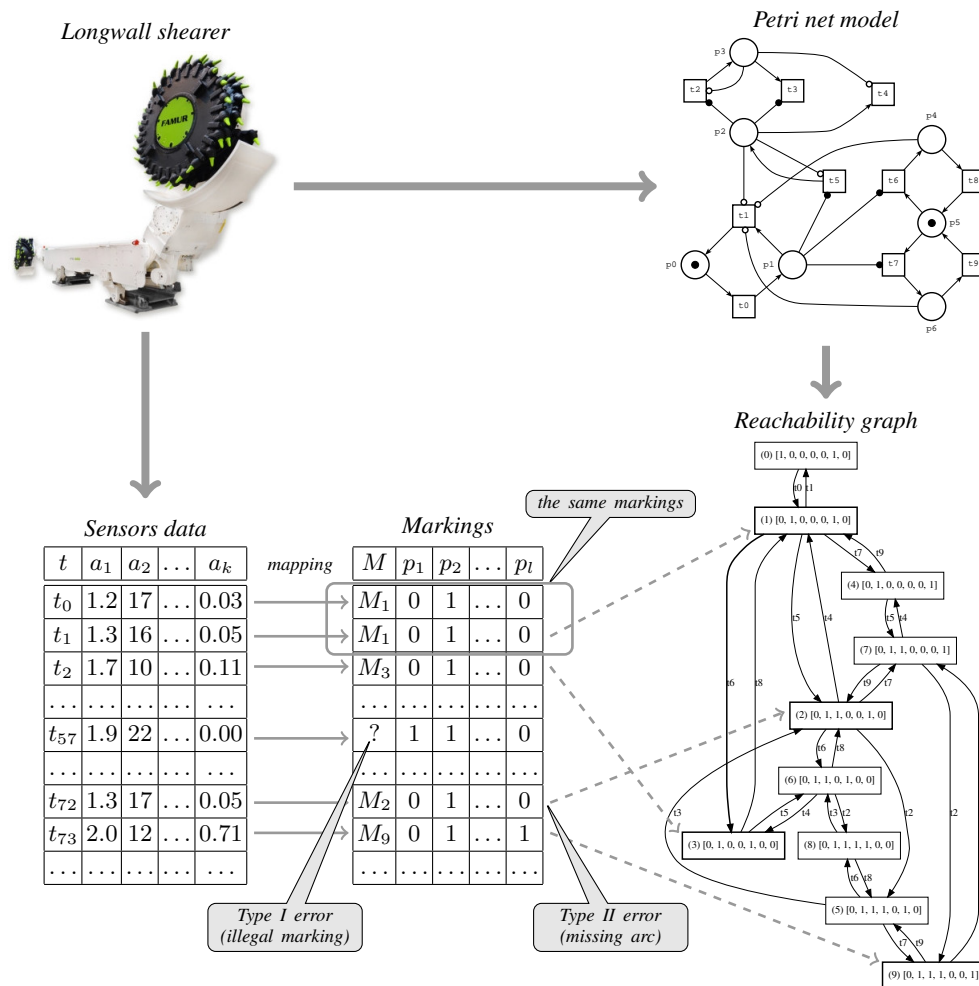
The main operation stages of the shearer are related to the coal cutting. The cutting stage can occur in any location in the longwall face. The return to the drive stage can occur only at the beginning or at the end of the longwall face. There are also other characteristic stages at the beginning and end of the longwall face, named the stoppage in on mode (marked in Figure 3 with bullets). They are strictly related to planned changes in the positions of the organs. The detailed description of the shearer operation stages is presented in Table 2.

Table 2. Main stages in the shearer operation.

Item	Stage Name	Part of the Cycle	Location	Location	Move Direction	Speed	Arm Up/Down
1	cutting_beginning_along	along	0 – 30	beginning	left	low	
2	stoppage_beginning_along / organ_change_I	along	30 – 40	beginning	–	0	+
3	return_to_drive_along	along	30 – 0	beginning	right	low	
4	stoppage_beginning_along / organ_change_II	along	0	beginning	–	0	+
5	cutting_beginning_along	along	0 – 30	beginning	left	low	
6	cutting_middle_along	along	30 – (max-30)	middle	left	medium/ high	
7	cutting_end_along	along	(max-30) – max	end	left	low	
8	stoppage_end_along / organ_change_III	along	max	end	–	0	+
9	cutting_end_return	return	(max-30) – max	end	right	low	
10	stoppage_end_return / organ_change_IV	return	(max-40) – (max-30)	end	–	0	+
11	return_to_drive_return	return	(max-30) – max	end	left	low	
12	stoppage_end_return / organ_change_V	return	max	end	–	0	+
13	cutting_end_return	return	(max-30) – max	end	right	low	
14	cutting_middle_return	return	30 – (max-30)	middle	right	medium/ high	
15	cutting_beginning_return	return	0 – 30	beginning	right	low	
16	stoppage_beginning_return / organ_change_VI	return	0	beginning	–	0	+

### 4. Our Approach in a Nutshell

The overview of the approach presented in this paper is given in Figure 4. The method uses place-transition Petri nets [35,36] for modelling of the operations of the longwall shearer. The model was built based on expert knowledge and represents the behaviour of industrial machinery at such a level of abstraction that we can distinguish important stages of the machinery’s operation.



**Figure 4.** Conformance checking of longwall shearer operation using Petri nets.

On the other hand, the trace of the actual behaviour of the industrial machinery takes the form of data records that store the sensors readings. Any such record can be mapped to a marking of the considered Petri net model automatically. Thus, because of the one-to-one correspondence between the sensors’ readings and formal model states, a real-time mapping of the machinery’s behaviour to a path in the reachability graph is possible.

Such an analysis can reveal two types of errors. Type I error means a situation where we cannot assign a legal model marking to the given log record. Type II error denotes a situation where we found two different consecutive markings that are not connected by an arc in the reachability graph. Regardless of the type of error, its occurrence indicates the unusual behaviour of the shearer and is a signal for a detailed analysis of such a situation.

### 5. The Formal Model of the Shearer Operation

The formal model is a crucial part of the presented approach. Based on expert knowledge, the model represents the shearer behaviour in a way convenient for non-experts. We decided to use place-transition Petri nets with inhibitor arcs [37] for modelling purposes. The subclass of Petri nets is



supported by TINA software (See the project webpage at <http://projects.laas.fr/tina/>, [36]). The tool can be used for model development, interactive model simulation and formal model analysis. The net can be formally defined as follows:

**Definition 1.** A place-transition net with inhibitor arcs is a tuple  $\mathcal{N} = (P, T, A, I, M_0)$ , where

- $P$  and  $T$  are finite sets of places and transitions such that  $P \cap T = \emptyset$ ;
- $A \subseteq P \times T \cup T \times P$  is the flow relation;
- $I \subseteq P \times T$  is the inhibiting relation;
- $M_0: P \rightarrow \mathbb{N}$  is the initial marking, where  $\mathbb{N} = \{0, 1, 2, \dots\}$ .

The preset of a transition  $t$  is the set  $\bullet t = \{p \in P: (p, t) \in A\}$ ; the postset of  $t$  is the set  $t^\bullet = \{p \in P: (t, p) \in A\}$ . The inhibitor set of a transition  $t$  is the set  ${}^\circ t = \{p \in P: (p, t) \in I\}$ . A marking  $M$  is a mapping  $M: P \rightarrow \mathbb{N}$ ; i.e., it is a distribution of tokens in places of the net.

**Definition 2.** A transition  $t$  is enabled in a marking  $M$  iff

$$\forall p \in \bullet t: M(p) \geq 1 \quad (1)$$

$$\forall p \in {}^\circ t: M(p) = 0 \quad (2)$$

If a transition  $t$  is enabled in a marking  $M$ , it may fire, changing the marking  $M$  to a marking  $M'$ , such that for any  $p \in P$

$$M'(p) = \begin{cases} M(p) + 1 & \text{for } p \in t^\bullet \setminus \bullet t \\ M(p) - 1 & \text{for } p \in \bullet t \setminus t^\bullet \\ M(p) & \text{otherwise} \end{cases} \quad (3)$$

We write  $M \xrightarrow{t} M'$  to denote the change of markings. A firing sequence starting at marking  $M$  is a sequence of transitions  $t_1, \dots, t_n$  such that there exists a sequence of markings  $M_1, \dots, M_n$  such that  $M \xrightarrow{t_1} M_1 \cdots \xrightarrow{t_n} M_n$ . In such a case we say that  $t_1, \dots, t_n$  leads from  $M$  to  $M_n$  and that  $M_n$  is reachable from  $M$ . An empty firing sequence leads from  $M$  to  $M$ . The set of all markings that are reachable from  $M$  is denoted by  $Reach(M)$ .

A Petri net model of a shearer is shown in Figure 5. Let us focus on transition `move_arms` and places `organs_working` and `organs_moving`. The TINA software uses an empty-circle arrowhead to denote an inhibitor arc. Such an arc leads from place `organs_moving` to transition `move_arms`, which means that the place must be empty if the transition is enabled. We say that there is a self-loop on a given place  $p$  and transition  $t$  if both  $(p, t) \in A$  and  $(t, p) \in A$ . This means that if  $t$  fires, it removes a token from  $p$  and adds back a token to  $p$ . In other words,  $t$  only tests the presence of a token in place  $p$ . To make a model more readable, TINA provides read arcs to represent self loops. It uses filled-circle arrowheads to denote read arcs. Such an arc leads from place `organs_working` to transition `move_arms`, which means that there must be a token in the place if the transition is enabled.

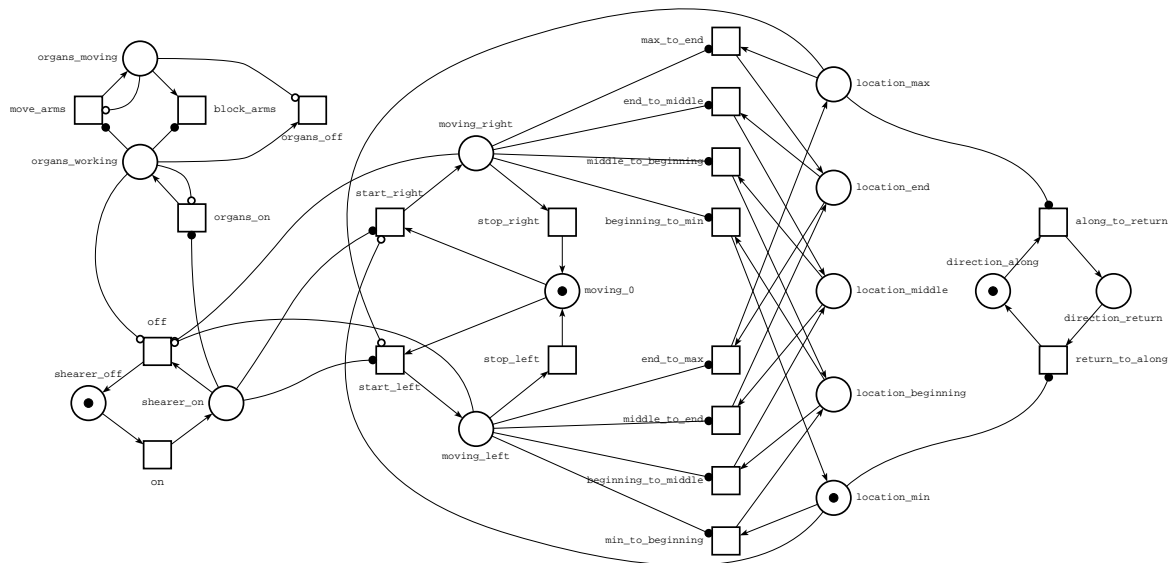


Figure 5. Petri net model of a shearer.

The Petri net presented in Figure 5 mimics the behaviour of a coal mine shearer and was built to identify the current shearer stage, as shown in Figure 3 or its illegal (unsafe) behaviour. The meanings of different parts of the model are shown in Figure 6. The main switch part represents the process of preparing the shearer to work. A token in place *shearer\_on* denotes that all systems (parts of shearer) have been checked and are ready to work. The organs’ state submodel represents the behaviour of shearer arms and organs. Both organs are working or are disabled at the same time. The tractor’s state part models tractor’s behaviour. Three modes are possible: moving right, moving left or stoppage. The shearer position part represents the position of the shearer (see Figure 3). Transitions from the shearer movement part are used to change the position. Finally, the general direction subnet is used to set the general shearer direction: along or return.

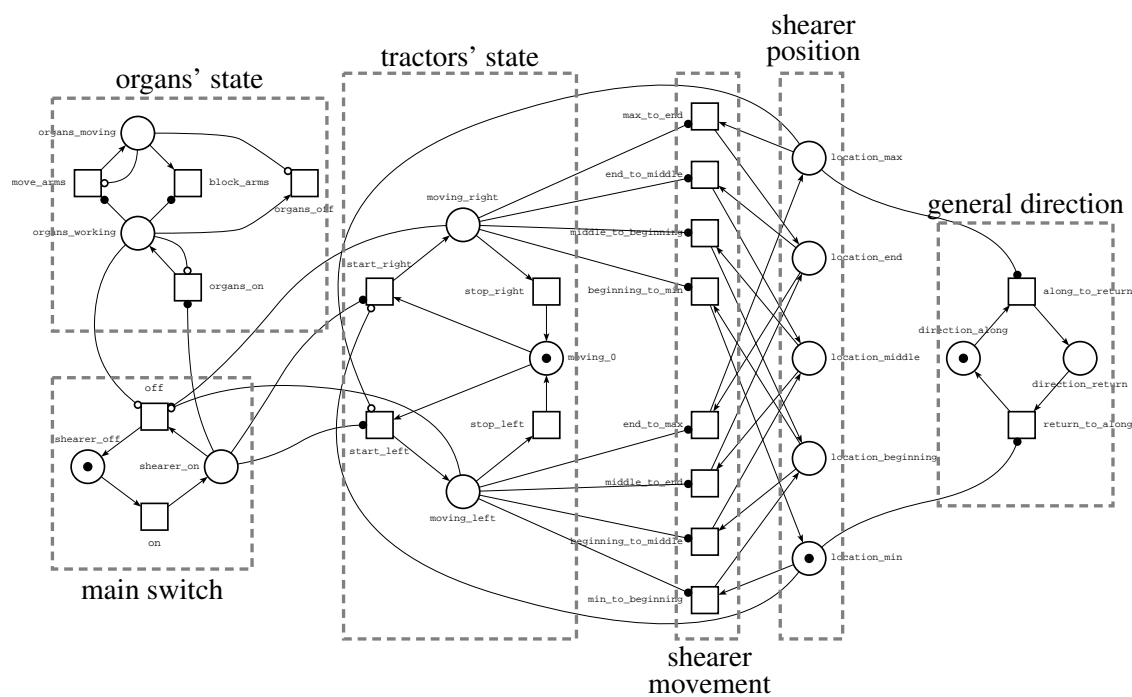


Figure 6. Petri net model of a shearer—meanings of model elements.

The state-space of a Petri net model is represented using a reachability graph [35]. The nodes of such a graph represent reachable model’s states. Each arc is labelled with the transition name that leads from the arc source to destination marking. The TINA tool can be used to generate a reachability graph automatically. The reachability graph generated for the model presented in Figure 5 contains 80 markings and 220 arcs.

Let us assume that the set of places of the Petri net is ordered as follows:  $P = \{ \text{shearer\_off}, \text{shearer\_on}, \text{moving\_right}, \text{moving\_0}, \text{moving\_left}, \text{organs\_working}, \text{organs\_moving}, \text{location\_min}, \text{location\_beginning}, \text{location\_middle}, \text{location\_end}, \text{location\_max}, \text{direction\_along}, \text{direction\_return} \}$ . A small part of the reachability graph is shown in Figure 7. To make the states more readable, markings are broken into four lines and some extra labels are included. Moving along a path in the graph, we can track the consecutive stages of the modelled machinery behaviour.

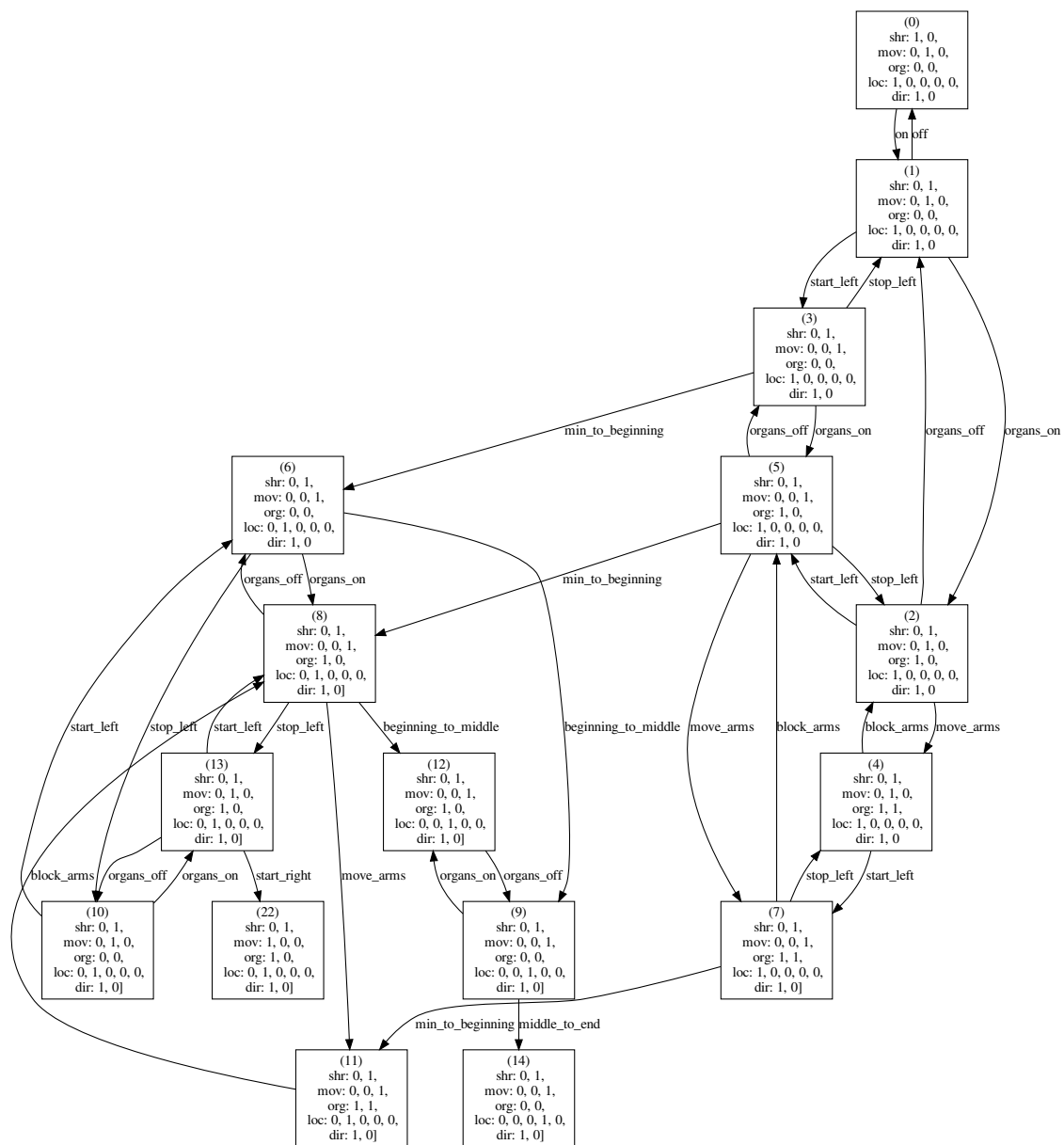


Figure 7. Part of the reachability graph for the model shown in Figure 5.

Selected results of application of the model for a shearer operation analysis are presented in the next section.

## 6. Experimental Results

Information gathered from the longwall machinery contains values for more than 150 attributes. Only a small subset, as shown in Table 1, is necessary to identify the current state of the shearer. To show the usability of the presented approach, we used samples of data collected from a coal mine in Poland where Famur equipment is used. Each sample contains between 50,000 and 200,000 records—data collected once a second. In this case, an offline analysis of historical data was carried out, but the same approach can be applied online while a shearer is working. A few records taken from one of the samples are shown in Table 3.

To supervise a shearer's work using its formal model, firstly, we have to map the current sensors' readings into a marking of the Petri net. The mapping rules are as follows:

- The marking of places from the general direction subnet (see Figure 6) is defined based on the direction attribute value; i.e., if the value is equal to along, place `direction_along` is marked. Otherwise, if the value is equal to return, place `direction_return` is marked.
- The marking of places from the shearer position subnet is defined based on the `SM_Location` attribute value. Even when analysing samples gathered for the same longwall, the minimum and maximum position may be slightly different. The position of the shearer is estimated based on the distance from its end point, but an end point is defined by pressing a reset button on the remote control by the shearer operator. Thus, we cannot be sure that the reset button is always used at the same real locations. If the shearer position is equal to its minimal location, then place `location_min` is marked; if the value belongs to the interval  $(\min, \min + 30]$ , then place `location_beginning` is marked, etc. (see Figure 2 for the descriptions of the middle and end sections).
- The marking of places from the tractor's state subnet is defined based on the values of attributes `speed`, `Move_Left` and `Move_Right`. If `Speed = 0`, place `moving_0` is marked. Otherwise, the values of the other two attributes are used to determine which of the places `moving_left` and `moving_right` should be marked.
- Finally, the place `organs_moving` is marked if at least one value of the attributes `Right_Arm_Up`, `Right_Arm_Down`, `Left_Arm_Up` and `Left_Arm_Down` is equal to true. The place `organs_working` is marked if arms are moving or at least one value of the attributes `LDC`, `RDC` is greater than 0.

Petri net's markings that correspond to the records in Table 3 are shown in Table 4.

The second stage is to track the path in the reachability graph that corresponds to current shearer state changes. If two consecutive shearer's stages map to the same marking, nothing happens from the formal model's point of view (records 42519, 42520 and 42521 all map to the marking 74). If they map to two different markings  $M$  and  $M'$ , then it is necessary to check whether there exists an arc leading from  $M$  to  $M'$  (see records 42521 and 42522 that map to different markings).

Sometimes changes between two consecutive shearer states are not expressed by firing a single transition. If such a change can be expressed by firing two transitions, we can choose whether it is to be reported in the analysis log or not.

The analysis procedure is implemented using the Python programming language. Two main arguments for the script are the reachability graph generated by the TINA tool (the verbose version) and a CSV file that contains sensors data. The reachability graph is represented by two data structures (two Python dictionaries). The former takes a marking (tuple) as a key and returns its number in the reachability graph. The latter takes a marking number and returns the set of its successors in the reachability graph. The sensors readings are stored using the Pandas DataFrame. The data frame is extended to store the results of mapping records to the corresponding markings and its content is analysed using the two dictionaries. A simplified representation of the verification algorithm is presented in Figure 8.

**Table 3.** Data collected from sensors.

No	Date_time	LDC	RDC	LHC	RHC	Speed	SM_Location	Right_Arm_Up	Right_Arm_Down	Left_Arm_Up	Left_Arm_Down	Move_Left	Move_Right	Direction
42519	24.01.2018 11:48:37	31	32	41	40	0	133	F	F	F	F	F	F	return
42520	24.01.2018 11:48:38	32	33	40	42	0	133	F	F	F	F	F	F	return
42521	24.01.2018 11:48:39	32	34	47	47	0	133	F	F	F	F	F	F	return
42522	24.01.2018 11:48:40	34	34	44	42	7	134	F	F	F	F	F	T	return
42523	24.01.2018 11:48:41	34	34	44	42	7	134	F	F	F	F	F	T	return
42524	24.01.2018 11:48:42	34	34	44	42	7	134	F	F	F	F	F	T	return

**Table 4.** Results of mapping of records from Table 3 into Petri net's markings.

Marking No	Shearer_Off	Shearer_On	Moving_Right	Moving_0	Moving_Left	Organs_Working	Organs_Moving	Location_Min	Location_Beginning	Location_Middle	Location_End	Location_Max	Direction_Alone	Direction_Return
74	0	1	0	1	0	1	0	0	0	1	0	0	0	1
74	0	1	0	1	0	1	0	0	0	1	0	0	0	1
74	0	1	0	1	0	1	0	0	0	1	0	0	0	1
61	0	1	1	0	0	1	0	0	0	0	1	0	0	1
61	0	1	1	0	0	1	0	0	0	0	1	0	0	1
61	0	1	1	0	0	1	0	0	0	0	1	0	0	1

```

# reach_dict -- markings dictionary
# arcs       -- arcs dictionary
# df         -- DataFrame (sensors data)

l_min = minimal_shearer_position()
l_max = maximal_shearer_position()
flag_init = True
prev_n = None
for i, r in df.iterrows():
    map_record_to_marking(r)
    m = marking(r)
    if m not in reach_dict.keys():
        log_illegal_marking(m)
        flag_init = True
    else:
        n = reach_dict[m]
        if flag_init == True:
            flag_init = False
        else:
            successors = arcs[prev_n]
            if (n != prev_n) & (n not in successors):
                successors2 = two_steps_successors()
                if (n in successors2):
                    log_multi_step(m)
                else:
                    log_illegal_arc(m)
            prev_n = n

```

Figure 8. Verification algorithm.

The `flag_init` variable stores the information whether there exists a legal previous marking or not, so it is possible to check the existence of an arc in the reachability graph. The `prev_n` variable stores the number of the previous legal marking. The meanings of other variables and functions seem straightforward.

The output of the procedure is a log file that contains information about all changes that do not match the model. A small excerpt from an analysis log is shown in Figure 9. Let us focus on the first presented problem. It refers to records 42521 and 42522 presented in Table 3 and the corresponding markings presented in Table 4. There does not exist an arc leading from marking 74 to marking 61 in the reachability graph, and there is not a two-length path leading from marking 74 to marking 61. Thus, the change from marking 74 to 61 was logged as an illegal arc. After analysing the records, we can see that the shearer begins to move to the right, but its position changes as for moving to the left. It is obvious that this means an error in the data obtained from the shearer sensors.

```

ILLEGAL ARC:
(42521) [32 34 47 47 0 133 F F F F F F 'return'] [0 1 0 1 0 1 0 0 0 1 0 0 0 1] (74)
(42522) [34 34 44 42 7 134 F F F F F T 'return'] [0 1 1 0 0 1 0 0 0 0 1 0 0 1] (61)
ILLEGAL ARC:
(50278) [36 38 43 48 8 134 F F F F T F 'return'] [0 1 0 0 1 1 0 0 0 0 1 0 0 1] (75)
(50279) [38 38 41 47 8 133 F F F F T F 'return'] [0 1 0 0 1 1 0 0 0 1 0 0 0 1] (85)
ILLEGAL ARC:
(50545) [43 44 51 50 5 128 F F F F T F 'return'] [0 1 0 0 1 1 0 0 0 1 0 0 0 1] (85)
(51001) [58 52 45 44 4 104 F F T F F T 'return'] [0 1 1 0 0 1 1 0 0 1 0 0 0 1] (71)
ILLEGAL ARC:
(51016) [39 38 43 42 4 105 T F F T F T 'return'] [0 1 1 0 0 1 1 0 0 1 0 0 0 1] (71)
(51017) [39 38 43 42 1 105 F F F F T F 'return'] [0 1 0 0 1 1 0 0 0 1 0 0 0 1] (85)
MULTI-STEP (2 TRANSITIONS):
(51106) [45 37 61 61 2 102 F F F F T F 'return'] [0 1 0 0 1 1 0 0 0 1 0 0 0 1] (85)
(52767) [37 36 45 44 6 89 F F F F F T 'return'] [0 1 1 0 0 1 0 0 0 1 0 0 0 1] (65)

```

Figure 9. Excerpt from analysis log.

The lack of a one-length or two-length path from marking 74 to marking 61 is also presented using two screenshots from the model simulation presented in Figures 10 and 11. Solid squares represent active transitions. Firing transition `start_right` (other enabling transitions do not lead to a marking similar to marking 61) leads from marking 74 (Figure 10) to marking 65 (Figure 11). It is not possible to fire a single enable transition in marking 65 to reach marking 61.

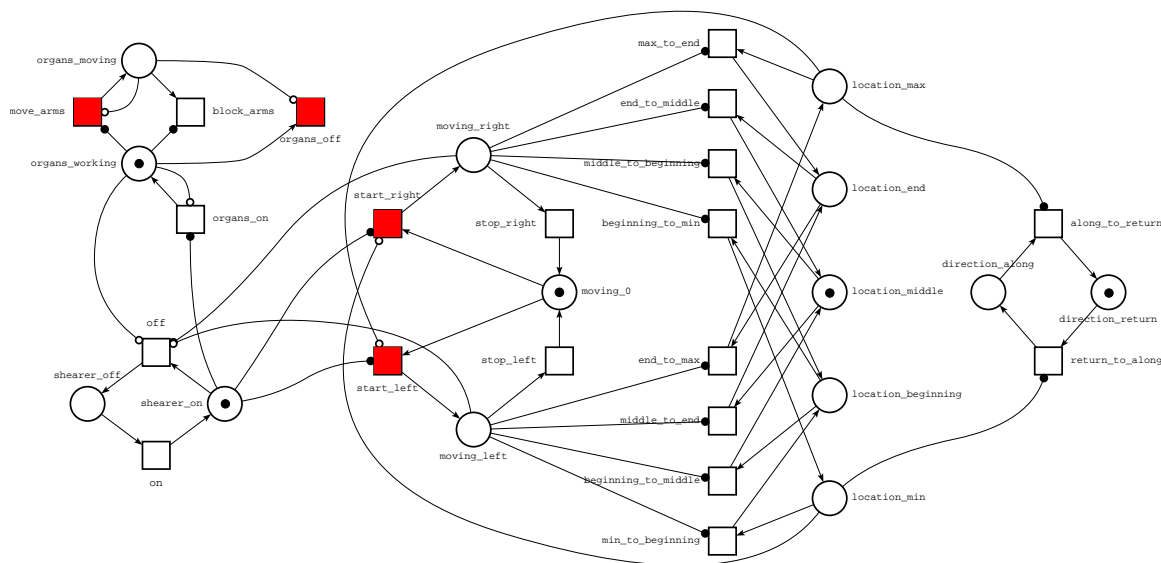


Figure 10. Model simulation using TINA—marking number 74.

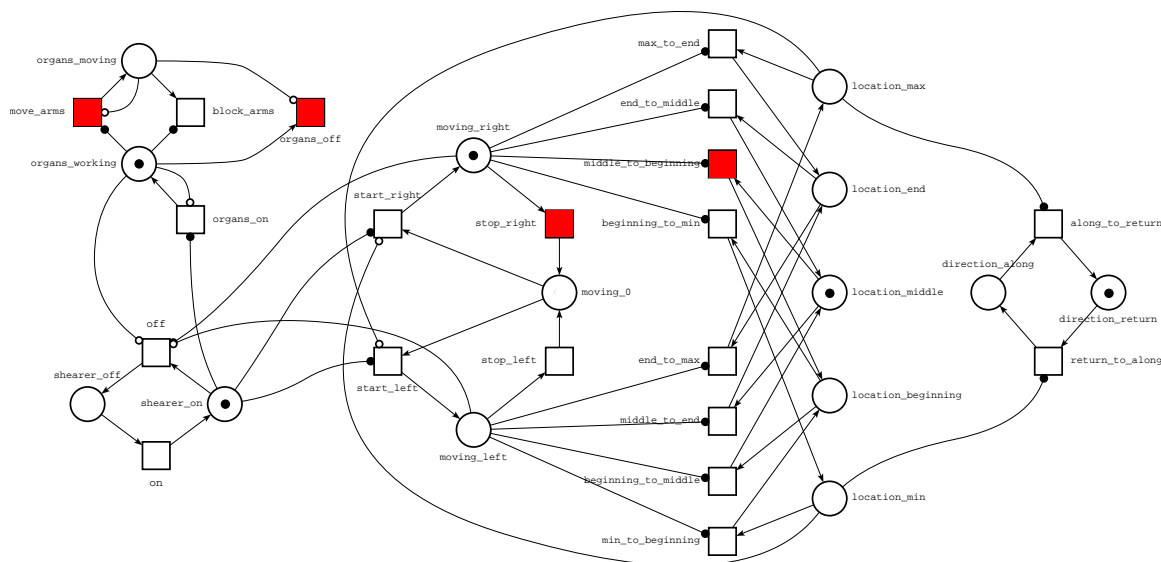


Figure 11. Model simulation using TINA—marking number 65.

Other errors found in the data include, but are not limited to:

- The shearer shifted by one meter even though the sensor indicated speed 0.
- The direction of the shearer's movement changed right to left without stopping.
- The shearer moved to the left after reaching the maximum position.
- The shearer moved to the right after reaching the minimum position.
- Gaps in data, i.e., breaks in the recording of sensor readings.

The average number of errors reported when the multi-steps are not logged is approximately 0.03% of the number of records. If we log multi-steps it is about 0.12% of the number of records in a sample. It is worth noting that most multi-steps represent legal changes, but there are situations when a multi-step represents a significant error in the data, e.g., a sudden change of direction without stopping, which is impossible for a machine with such weight as a shearer.

The process supervisor receiving the warning about detected difference (error) between a formal process model and the process execution can decide, based on his own assessment and experience,

whether to stop the process (stop the shearer) or continue the operations, because the deviation results from the known causes (e.g., software errors) or environmental conditions (in the excavation). Having up-to-date information about events in the excavation, the supervisor has more complete information that can be distributed among other participants and owners of the process and further used in decision-making process.

## 7. Summary

Most approaches to conformance checking available in the literature perform offline analyses. This means that the analysis of the process is performed after the completion of the current execution. The approach presented in this paper can be used both for offline and online analysis. It means it is capable of analysing the deviating behaviour on the fly during the process execution.

In this paper we focus on a specific use case related to the underground longwall mining and its main facility, which is a longwall shearer. As presented in Section 3, it is possible to describe the longwall process as a sequence of precisely defined stages. In fact, the shearer's behaviour can be significantly different from its ideal execution. Moreover, an analysis of the execution as a whole seems to be of little use in such a case. Therefore, our approach focuses on a small snippet of the log file to assess whether changes between the last saved shearer states can be considered legal.

The proposed method is implemented using the Python language and was used for verification of log files collected from a few coal mines located in Poland. Due to the lack of access to the real-life operating system, the work presented in this paper was restricted only to the offline analysis, as the online experiments were not possible. Therefore, the online experiments are one of our important directions for future work.

It is worth emphasising that the usability of the approach was assessed directly by Famur employees who deal with the analysis of data collected during the shearer work. Offline analysis of historical logs allowed us to indicate possibilities of improving the system of recording data collected from machinery sensors.

The presented results summarise one of the main stages of the PACMEL project. The described method can be further developed. First of all, the same approach can be applied not only to the shearer but also to the transport system. Such a task requires an extension of the formal model and the acquisition of data from other elements of the mining system. On the other hand, it is possible to apply a high-level Petri net for modelling purposes [38]. Such an approach would allow one not only to build a model at a lower abstraction level, but also to obtain a digital twin for the shearer.

**Author Contributions:** Conceptualisation, M.S. and G.J.N.; methodology, M.S., E.B. and J.K.; software, M.S.; validation, M.S. and E.B.; investigation, M.S., E.B., G.J.N. and A.N.; data curation, M.S., E.B. and J.K.; writing—original draft preparation, M.S., E.B., G.J.N. and A.N.; writing—review and editing, G.J.N.; supervision, G.J.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper was funded by the National Science Centre, Poland under CHIST-ERA programme, the CHIST-ERA 2017 BDSI PACMEL Project, NCN 2018/27/Z/ST6/03392.

**Acknowledgments:** We would like to thank Famur S.A. for cooperation and consultation.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Bezerra, F.; Wainer, J. Anomaly detection algorithms in business process logs. In Proceedings of the Tenth International Conference on Enterprise Information Systems—Volume 2: ICEIS, INSTICC, SciTePress, Barcelona, Spain, 12–16 June 2008; pp. 11–18. [[CrossRef](#)]
2. Chen, D.; Panfilenko, D.V.; Khabbazi, M.R.; Sonntag, D. A model-based approach to qualified process automation for anomaly detection and treatment. In Proceedings of the 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, Germany, 6–9 September 2016; pp. 1–8.



3. Böhmer, K.; Rinderle-Ma, S. Automatic Signature Generation for Anomaly Detection in Business Process Instance Data. In *Enterprise, Business-Process and Information Systems Modeling*; Schmidt, R., Guédria, W., Bider, I., Guerreiro, S., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 196–211.
4. Nolle, T. Seeliger, A.; Mühlhäuser, M. *Unsupervised Anomaly Detection in Noisy Business Process Event Logs Using Denoising Autoencoders*; Discovery Science; Calders, T., Ceci, M., Malerba, D., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 442–456.
5. Ferreira, D.R.; Szimanski, F.; Ralha, C.G. Improving Process Models by Mining Mappings of Low-Level Events to High-Level Activities. *J. Intell. Inf. Syst.* **2014**, *43*, 379–407, [[CrossRef](#)]
6. Rogge-Solti, A.; Kasneci, G. Temporal Anomaly Detection in Business Processes. In *Business Process Management*; Sadiq, S., Soffer, P., Völzer, H., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 234–249.
7. Choi, S.; Youm, S.; Kang, Y.S. Development of Scalable On-Line Anomaly Detection System for Autonomous and Adaptive Manufacturing Processes. *Appl. Sci.* **2019**, *9*, 4502. [[CrossRef](#)]
8. Cabanillas, C.; Di Ciccio, C.; Mendling, J.; Baumgrass, A. Predictive Task Monitoring for Business Processes. In *Business Process Management*; Sadiq, S., Soffer, P., Völzer, H., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 424–432.
9. Liu, Q.; Leng, J.; Yan, D.; Zhang, D.; Wei, L.; Yu, A.; Zhao, R.; Zhang, Z.; Chen, X. Digital twin-based designing of the configuration, motion, control, and optimization model of a flow-type smart manufacturing system. *J. Manuf. Syst.* **2020**, [[CrossRef](#)]
10. Shao, G.; Helu, M. Framework for a digital twin in manufacturing: Scope and requirements. *Manuf. Lett.* **2020**, *24*, 105–107. [[CrossRef](#)] [[PubMed](#)]
11. Cai, Y.; Wang, Y.; Burnett, M. Using augmented reality to build digital twin for reconfigurable additive manufacturing system. *J. Manuf. Syst.* **2020**. [[CrossRef](#)]
12. Weske, M. *Business Process Management—Concepts, Languages, Architectures*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2019. [[CrossRef](#)]
13. Janiesch, C.; Koschmider, A.; Mecella, M.; Weber, B.; Burattin, A.; Di Ciccio, C.; Fortino, G.; Gal, A.; Kannengiesser, U.; Leotta, F.; et al. The Internet of Things Meets Business Process Management: A Manifesto. *IEEE Syst. Man Cybern. Mag.* **2020**, *6*, 34–44. [[CrossRef](#)]
14. Seiger, R.; Zerbato, F.; Burattin, A.; García-Bañuelos, L.; Weber, B. Towards IoT-driven Process Event Log Generation for Conformance Checking in Smart Factories. In Proceedings of the 2020 IEEE 24th International Enterprise Distributed Object Computing Workshop (EDOCW), Eindhoven, The Netherlands, 5 October 2020.
15. Reißner, D.; Conforti, R.; Dumas, M.; La Rosa, M.; Armas-Cervantes, A. Scalable Conformance Checking of Business Processes. In Proceedings of the On the Move to Meaningful Internet Systems, OTM 2017 Conferences, Rhodes, Greece, 23–28 October 2017; pp. 607–627.
16. Baier, C.; Katoen, J.P. *Principles of Model Checking*; The MIT Press: London, UK, 2008.
17. van der Aalst, W. *Process Mining: Data Science in Action*; Springer: Berlin/Heidelberg, Germany, 2016. [[CrossRef](#)]
18. Diba, K.; Batoulis, K.; Weidlich, M.; Weske, M. Extraction, correlation, and abstraction of event data for process mining. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2019**. [[CrossRef](#)]
19. van Zelst, S.J.; van Dongen, B.F.; van der Aalst, W.M.P. Event stream-based process discovery using abstract representations. *Knowl. Inf. Syst.* **2018**, *54*, 407–435. [[CrossRef](#)]
20. van Zelst, S.J.; Mannhardt, F.; de Leoni, M.; Koschmider, A. Event abstraction in process mining: literature review and taxonomy. *Granul. Comput.* **2020**. [[CrossRef](#)]
21. Koschmider, A.; Janssen, D.; Mannhardt, F. Framework for Process Discovery from Sensor Data. In Proceedings of the 10th International Workshop on Enterprise Modeling and Information Systems Architectures, Kiel, Germany, 14–15 May 2020; pp. 32–38.
22. Rozinat, A.; van der Aalst, W. Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* **2008**, *33*, 64–95. [[CrossRef](#)]
23. Adriansyah, A. Aligning Observed and Modeled Behavior. Ph.D. Thesis, Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2014. [[CrossRef](#)]
24. Weidlich, M. Behavioural Profiles: A Relational Approach to Behaviour Consistency. Ph.D. Thesis, Universität Potsdam, Potsdam, Germany, 2011.

25. Carmona, J.; van Dongen, B.; Solti, A.; Weidlich, M. *Conformance Checking—Relating Processes and Models*; Springer: Berlin/Heidelberg, Germany, 2018. [[CrossRef](#)]
26. Burattin, A.; van Zelst, S.; Armas-Cervantes, A.; van Dongen, B.; Carmona, J. Online Conformance Checking Using Behavioural Patterns. In *Business Process Management*; Weske, M., Montali, M., Weber, I., vom Brocke, J., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 250–267.
27. Burattin, A.; Carmona, J. A Framework for Online Conformance Checking. In *Business Process Management Workshops*; Teniente, E., Weidlich, M., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 165–177.
28. Rodriguez-Fernandez, V.; Trzcionkowska, A.; Gonzalez-Pardo, A.; Brzychczy, E.; Nalepa, G.J.; Camacho, D. Conformance Checking for Time Series-aware Processes. *IEEE Trans. Ind. Inform.* **2020**, *17*, 871–881, [[CrossRef](#)]
29. Leemans, S.J.; Fahland, D.; Aalst, W. Scalable Process Discovery and Conformance Checking. *Softw. Syst. Model.* **2018**, *17*, 599–631. [[CrossRef](#)] [[PubMed](#)]
30. Brzychczy, E.; Trzcionkowska, A. New possibilities for process analysis in an underground mine. *Zesz. Nauk. Organ. i Zarządzanie Politech. Śląska* **2017**, *111*, 13–25. [[CrossRef](#)]
31. Heyduk, A. Petri Networks for Mechanized Longwall System Simulation. In *Advanced, Contemporary Control*; Bartoszewicz, A., Kabziński, J., Kacprzyk, J., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 307–318.
32. Singh, R. *Principles and Practices of Modern Coal Mining*; New Age International: Kerala, India, 2014.
33. Snopkowski, R.; Napieraj, A. Method of the production cycle duration time modeling within hard coal longwall faces/Metoda probabilistycznego modelowania czasu trwania czynności cyklu produkcyjnego realizowanego w przodkach ścianowych kopalń węgla kamiennego. *Arch. Min. Sci.* **2012**, *57*, 121–138, [[CrossRef](#)]
34. Wojaczek, A. Telematics in underground mining. *Inform. Autom. Electr. Eng.* **2017**, *4*, 19–26. [[CrossRef](#)]
35. Murata, T. Petri Nets: Properties, Analysis and Applications. *Proc. IEEE* **1989**, *77*, 541–580. [[CrossRef](#)]
36. Berthomieu, B.; Ribet, P.O.; Vernadat, F. The tool TINA – Construction of Abstract State Spaces for Petri Nets and Time Petri Nets. *Int. J. Prod. Res.* **2004**, *42*, 2741–2756. [[CrossRef](#)]
37. Busi, N. Analysis issues in Petri nets with inhibitor arcs. *Theor. Comput. Sci.* **2002**, *275*, 127–177. [[CrossRef](#)]
38. Szpyrka, M.; Biernacki, J.; Biernacka, A. Tools and methods for RTCP-nets modeling and verification. *Arch. Control Sci.* **2016**, *26*, 339–365. [[CrossRef](#)]

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).