Process mining in industrial control systems

Midhun Xavier *, Victor Dubinin † Sandeep Patil*, Valeriy Vyatkin* ‡
* Department of Computer Science, Computer and Space Engineering, Lulea Tekniska Universitet, Sweden
† Independent researcher
‡Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland

Email: midhun.xavier@ltu.se, dubinin.victor@gmail.com, sandeep.patil@ltu.se, vyatkin@ieee.org

***Abstract*—In this paper, we discuss how process mining techniques can be applied in industrial control systems for modeling, verification, and enhancement of the cyber-physical system based on recorded data logs. Process mining is used for extracting the process models in different notations from the recorded behavioral traces of the system. The output model of the system's behavior is mainly derived using an open-source tool called ProM. The model can be used for such applications as anomaly detection, detection of cyber-attacks and alarm analysis in industrial control systems with the help of various control flow discovery algorithms. The extracted process model can be used to verify how the event log deviates from it by replaying the log on Petri net for conformance analysis.**

***Index Terms*—Cyber-physical automation systems, IEC 61499, Process mining**

I. Introduction

Process mining [1] extracts the behavior of the system by analyzing the events in order and it consists of process discovery, conformance checking and process enhancement. Process models derived from event logs can be classified in different ways like how formal the model is, how the model is constructed, etc., and popular process modelling paradigms are Transition systems, Petri nets, Workflow nets, Business Process Modelling Notations (BPMN), Causal nets (C-nets), etc. Data mining and process mining have some differences even if they used to predict patterns from data logs. Data mining is used to discover or predict the patterns by analyzing the data sets but process mining combines data analysis with modelling for extracting deep insights about the processes from recorded event logs [2]. While data mining ignores the processes, process mining is really interested in processes using the data.

There are three components in process mining technique [3] i.e., process discovery, conformance checking and enhancement. The process discovery generates a model from a recorded event log with the help of several control flow discovery algorithms. The generated model and other event logs from the same system can be compared to identify the deviations this is called conformance checking. The generated model can be updated by analysing a new set of event logs called enhancement.

Many process scenarios can be constructed by simulating the Petri net [4] and this method is called 'Play Out'. Instead of simulating the Petri net, we can use the simulation model or digital twins or even real systems to create the event log and inferring the model from many scenarios or traces is called process discovery or 'Play In'. It is possible to identify the deviation of the model by replaying a scenario on a built model. These features in process mining are helpful to identify bottlenecks in process and where machines deviate from expected process [5].

Cyber-physical systems (CPS) [6] is a popular designation for complex industrial automation systems with decentralized control logic distributed across many communicating devices, often embedded into various mechatronic components. The IEC 61499 architecture [7] is considered as a suitable method for modelling cyber-physical automation systems. In the modern automation industry world, mixed structure of distributed controllers in different mechatronic components introduces the verification and validation challenges, so formal modelling of CPS is necessary for their formal verification. The latter helps making the system less prone to errors by checking their behaviour comprehensively on compliance with specifications expressed in such formal languages, as temporal logic, e.g., LTL or CTL [8]. Closed-loop modelling is considered beneficial for the verification but it requires the model of the plant. The implementation of the plant model is complex and resource consuming, and it is normally done by manually. The process mining approach opens a wide range of opportunities for modelling industrial control systems. In this paper, we discuss how process mining can be effectively applied in the field of industrial automation systems and it also describes the implementation of process model from event log by process discovery algorithm. Then we demonstrate how to check if the new event log deviates from the expected behaviour.

The paper is structured as follows: Section II discusses the related work and process mining in industrial control systems. Section III explains the event log structure, process mining tools and its advantages, selection of process discovery algorithms and conformance checking of process model in detail. Finally, Section IV concludes the paper and outlines future goals.

II. Process mining in industrial control systems : Overview

A. *Process mining in factory automation*

Process mining in industrial control systems can be applied in various directions. Process mining techniques applied in factory automation are used for model enhancement and conformance checking [9]. The industrial control systems are used to record the events and this trace of events is called event log. The event log can be in various formats like CSV, XES, MXML etc. The control work-flow model is discovered from the event log with the help process mining algorithms. There are several types of control flow discovery algorithms, but we need to select one of them according to the event log and depending on the goal of the process model. The model built by the process discovery algorithms is evaluated with the help of basic performance analysis which considers fitness, precision, over-fitting, and simplicity parameters. These parameters can be measured by replaying event log on derived model.

B. Anomaly detection using log data

Anomaly detection using log data found to be another application using process mining technique. Paper [10] introduces a method for identifying anomalous behaviour of the industrial control system using device logs with the help of process discovery and conformance checking. Process mining techniques are mainly used in business related areas to improve the process by analysing the event log, but it is possible to detect cyber-attacks and anomalous behaviour of the industrial control system by analysing the event log using conformance checking. In the paper [10], process discovery algorithm generates process models which are used for conformance checking. The latter is done with the help of a token game which compares the new event log with the generated model. Myers found in [11] that models created with the help of inductive miners give good fit compared to other miners, especially in the field of industry control system.

C. Alarm analysis from the event-log database of an industrial plant

In industries alarm analysis can be done with the help of process mining techniques. Abonyi and Dorgo explain in [12] how process mining techniques can be used effectively for the alarm analysis from the event-log database of an industrial plant. Here, the process model is derived from fuzzy miner instead of the alpha miner [13] because alpha miner does not consider number of times the traces are repeated in event log while the fuzzy miner keeps the highly important behaviour of the system.

D. online parameter estimation for CPPS with process mining

In the cyber physical production systems (CPPS), the need to adjust in production entails the need to change the automation software which requires a lot of manual engineering effort. In order to fix this, paper [14] provides online parameter estimation for a CPPS using process mining. Alpha algorithm is one of the popular process discovery algorithms which creates dependency graphs from event logs. The dependencies between each event are created according to its order of events in the event log. If any noises present in the system, then the alpha algorithm produces a high difference from the expected behaviour [14].

E. PLC programming logic modelling and other applications in ICS

The alarm analysis, parameter estimation and detection of cyber-attacks in cyber physical systems based on outlier analysis in event logs were major application of the process mining. However, Theis at al in [15] propose a method to model PLC programming logic by analysing event log with the help of process mining. The paper uses split miner as a process discovery algorithm and uses DREAM-NAP (decay replay mining - next activity prediction) for predicting next activities in the running process. In the modern industry world robots help in the manufacturing system to increase the overall productivity of the processes. The data captured from the robots can be used to create a general model of the manufacturing system to understand and extract hidden behaviour of the system. The generated model and other event logs from the same system can be compared to identify the deviations. In this case, the process discovery algorithm inductive miner is used because it gives a more generalized model compared to other control flow discovery algorithms. Another paper [16] which collects data from the factory floor and converts it to event stream in order to generate a model for conformance checking. The conformance checking detects deviation from the manufacturing floor.

F. Plant model generation from event log

In recent years, cyber physical systems are used in almost all wireless communication areas. These systems produce several flaws due to its decentralized and heterogeneous structure. Formal verification of these systems become more relevant in order to verify and detect possible errors in the system. Modelling of the controller is straightforward because logic is already known but the construction of plant model is difficult. Previously, the researchers manually constructed the model of the plant and used it for verification. The paper [17] describes how to construct plant models automatically from event logs for formal verification. Formal model of the plant in SMV format is developed and verification is done with the help of a symbolic model checker tool called NuSMV.

III. PROCESS MODEL EXTRACTION

Consider a running example of 'Gripper and conveyor' system and see how the process flow of the system can be derived from the event log with the help of process discovery algorithms. The structure of the system is shown in the Figure 1 consisting of a gripper and conveyor. Gripper can either move in 'upward' or 'downward' direction and the clamp attached to the gripper component can 'open' and 'close' to grab an object. The conveyor component moves in one direction when the actuator signal is triggered. There are five sensors and three actuators exists in the system and their description is given below:

1) Sensor signals:

- S1_cup_detected : Whenever a cup or an object appears in the sensor (S1) then its value will change to 'TRUE' otherwise its value remains the same as 'FALSE'.

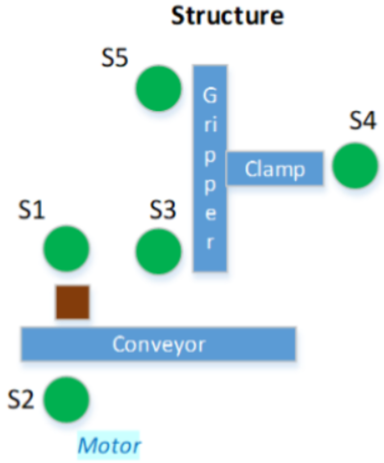


Fig. 1. Gripper and conveyor system structure

- S2_conveyor_running : If the conveyor is running then the sensor (S2) value becomes 'TRUE' otherwise 'FALSE'.
- S3_gripper_at_bottom : If the gripper's clamp reaches bottom position then the sensor (S3) value becomes 'TRUE' otherwise 'FALSE'.
- S4_gripper_closed : Whenever clamp is closed then sensor (S4) value will change to 'TRUE' otherwise its value remains same as 'FALSE'.
- S5_gripper_at_top : If the gripper's clamp reaches top position then the sensor (S5) value becomes 'TRUE' otherwise 'FALSE'.

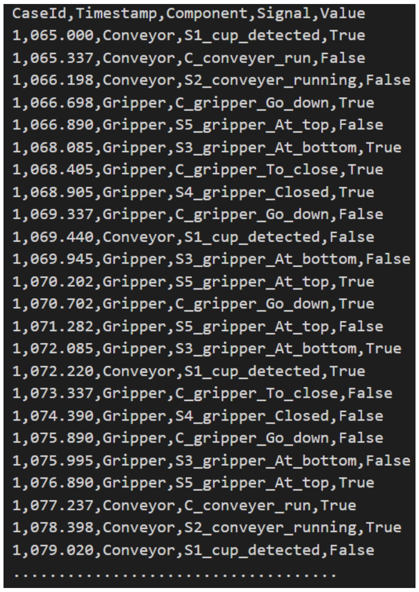
2) Control signals:

- C_conveyor_run : If the control signal becomes 'TRUE' then the conveyor starts running otherwise it stops.
- C_gripper_go_down : If this signal becomes 'TRUE' then the gripper starts moving downwards otherwise it moves upward.
- C_conveyor_to_close : If this signal becomes 'TRUE' then the clamp closes otherwise it opens.

A simple process sequence is taken into account and it works cyclically for a particular period of time. Initially, the conveyor is moving, the gripper rests at top position and the clamp is already at open condition. Whenever 'CUP' (workpiece) is detected by sensor s1 then the conveyor stops running. Gripper comes down and grabs the 'CUP' with the help of its clamp. After that the gripper returns the 'CUP' on the conveyor and the conveyor moves again. This process runs cyclically and if there is no object then the conveyor keeps on running and the gripper does not do anything.

A. Event log structure, attribute selection and pre-processing

An event description in the log consists of the following fields [10]: case identifier, event id/name and attributes. The case identifier is a unique id for each execution of processes, Normally, industrial control system processes are cyclic so each cycle can be considered as a different process instance. The event name refers to the triggered activities that occur while running the system. The attributes' part consist of resource, timestamp, etc. These attributes give additional information, i.e., ordering information, which component produced the event, etc. The attributes are not mandatory fields but if we get more information, then that would be useful to extract hidden features about the system.



```
CaseId,Timestamp,Component,Signal,Value
1,065.000,Conveyor,S1_cup_detected,True
1,065.337,Conveyor,C_conveyer_run,False
1,066.198,Conveyor,S2_conveyer_running,False
1,066.698,Gripper,C_gripper_Go_down,True
1,066.890,Gripper,S5_gripper_At_top,False
1,068.085,Gripper,S3_gripper_At_bottom,True
1,068.405,Gripper,C_gripper_To_close,True
1,068.905,Gripper,S4_gripper_Closed,True
1,069.337,Gripper,C_gripper_Go_down,False
1,069.440,Conveyor,S1_cup_detected,False
1,069.945,Gripper,S3_gripper_At_bottom,False
1,070.202,Gripper,S5_gripper_At_top,True
1,070.702,Gripper,C_gripper_Go_down,True
1,071.282,Gripper,S5_gripper_At_top,False
1,072.085,Gripper,S3_gripper_At_bottom,True
1,072.220,Conveyor,S1_cup_detected,True
1,073.337,Gripper,C_gripper_To_close,False
1,074.390,Gripper,S4_gripper_Closed,False
1,075.890,Gripper,C_gripper_Go_down,False
1,075.995,Gripper,S3_gripper_At_bottom,False
1,076.890,Gripper,S5_gripper_At_top,True
1,077.237,Conveyor,C_conveyer_run,True
1,078.398,Conveyor,S2_conveyer_running,True
1,079.020,Conveyor,S1_cup_detected,False
....................................
```

Fig. 2. Event log

An event log of the 'Gripper Conveyor' system is shown in the Figure 2 and it consists of five columns: CaseId, Timestamp, Components, Signal, Value. The case identifier in this log is denoted as 'caseId' which is a unique id for each process execution and here the event is composed of three columns: component, signal, and value. Timestamp represents the time at which the event occurred and it is considered as an attribute. This event log is taken as the input for further processing so it's necessary to record the correct information. The event log is sorted using 'Time Stamp' because the order of occurrence of events is a key factor and process discovery algorithms work mainly with the relation of these events.

Event log should be cleaned to get a good quality event log. In many situations event log quality should not be compromised, in order to produce an accurate model of the system. For anomaly detection, the log pre-processing step is ignored because these outliers or irrelevant events help to detect the cyber-attack while modelling the system with the help of process mining technique [10].

B. Process mining tools and its advantages

Process mining consists of process discovery algorithms and conformance checking can be done via programs, but process mining tools give wider option to use these all algorithms and it provides the result in different notation to give better visualization. ProM [18] [19] and Disco [20] [21] are the most popular tools used for process mining. ProM is an open-source tool which is is widely used because of the following features.

ProM version 6 consists of 250+ plugins which are used for event log pre-processing, process discovery and visualizations. The representation of the process model can be expressed in different notations but most commonly represented as Petri nets. There are many process discovery algorithms [22] like Alpha, Alpha +, PL based, T alpha, Petrify miner, etc. to produce output models as Petri net and ProM tool which supports almost all of them. ProM also supports plugins for conformance checking, and LTL specification checking. The fitness of the model derived from event log can be analysed using conformance checking. The ProM framework has different plugins for basic performance analysis and these plugins help to identify how much the generated model deviated from event log. The event log analysis, log pre-processing, and conversion from one event log format to another like CSV to eXtensible Event Stream (XES) can be done easily with this tool. On the other hand, Disco, developed by Gunther in 2007, is based on the fuzzy miner algorithm. The fuzzy miner gives a better interactive representation to understand the system behaviour of complex logs and it also works in the ProM tool.

Most of the process discovery algorithms take input in XES format and produce process models. In order to convert the event log from CSV to XES format, the Standard XES attributes need to be mapped by selecting the ’Case’ columns, ’Event’ columns, ’Start Time’ column and ’Completion Time’ column from the CSV data log. In the ’Gripper Conveyor’ system the event log in CSV is converted by mapping the standard XES attributes as follows:

- Selected Case Columns : CaseId
- Selected Event Columns : Component, Signal and Value
- Start Time & Completion Time is not selected because timing information is not considered for this experiment.

C. Selection of Process discovery algorithms

There are several types of process discovery algorithms: Abstraction based, Heuristic based, Search based, Region-based algorithms etc and each algorithm is used to extract different process models using event log. Abstract -based algorithms generate models by ordering relations of events in an event log, and on the other hand heuristic miner generates models where events are ordered based on frequency of events. Events happening in fewer thresholds are ignored so heuristic miners perform better with event logs containing noisy data. Search based algorithms (Genetic algorithm Miner (GA)) which try to mimic the process of evolution. Process discovery technique key factors is the balance between fitness, precision, generalisation and simplicity [23]. While selecting the process discovery algorithm, one should consider the following questions: how fast analysis technique produce result, how much memory it is used, what is representation of process discovery algorithm and whether it solves the related problems.

The most commonly used process discovery algorithm is the alpha algorithm, which is an abstraction-based algorithm. The process model extracted from the event log 2 of ’Gripper-Conveyor System’ using the alpha algorithm is shown in the Figure 3 and it explains the process flow in the Petri net notation. The Petri net consists of 16 transitions and each transition denotes the activity from the event log. The alpha algorithm creates a dependency graph based on the order of events in the event log. It does not consider the frequency of trace, so noises present in the log makes a high difference from expected behaviour. In order to avoid this, we can use fuzzy miner which is a heuristic approach and it generates a model according to the frequency of the traces. According to [12], ”the algorithm calculates the importance of the activities and how closely the events follow each other”. The process model extracted from the same ’Gripper Conveyor’ system using fuzzy miner in Disco is shown in the Figure 4. It is exactly similar to the Petri net obtained from the alpha

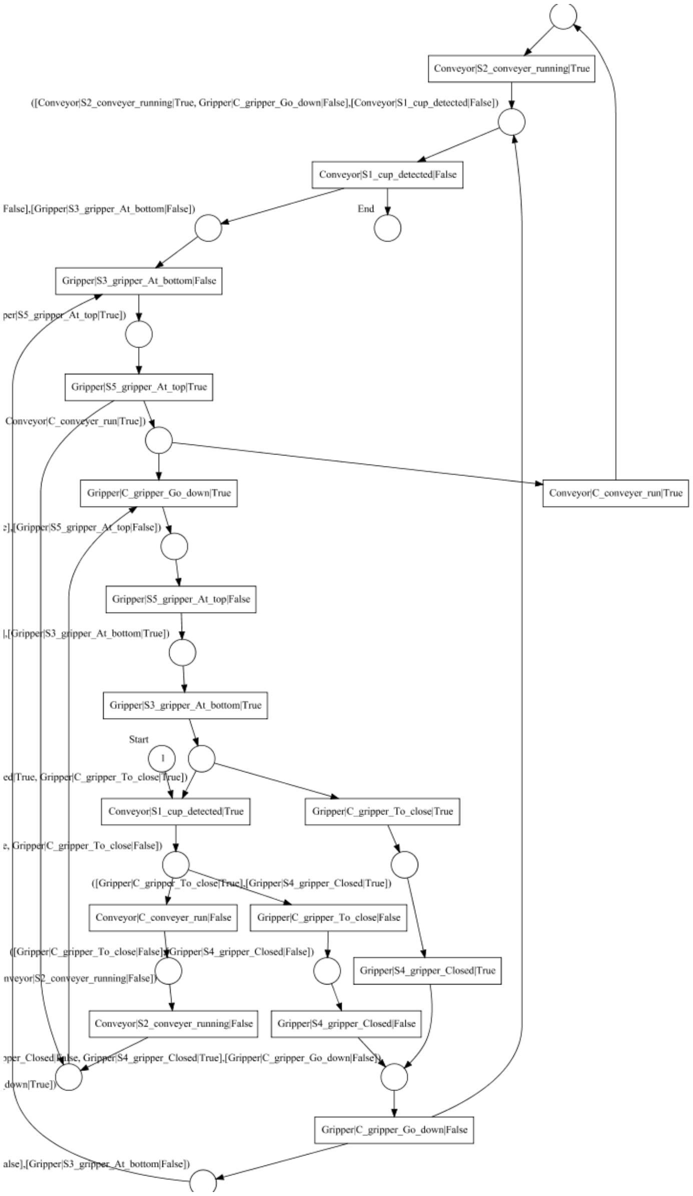

Conveyor|S2_conveyer_running|True
([Conveyor|S2_conveyer_running|True, Gripper|C_gripper_Go_down|False],[Conveyor|S1_cup_detected|False])
Conveyor|S1_cup_detected|False
False],[Gripper|S3_gripper_At_bottom|False])
End
Gripper|S3_gripper_At_bottom|False
per|S5_gripper_At_top|True])
Gripper|S5_gripper_At_top|True
Conveyor|C_conveyer_run|True])
Gripper|C_gripper_Go_down|True
Conveyor|C_conveyer_run|True
e],[Gripper|S5_gripper_At_top|False])
Gripper|S5_gripper_At_top|False
],[Gripper|S3_gripper_At_bottom|True])
Gripper|S3_gripper_At_bottom|True
Start
1
ed|True, Gripper|C_gripper_To_close|True])
Conveyor|S1_cup_detected|True
Gripper|C_gripper_To_close|True
e, Gripper|C_gripper_To_close|False])
([Gripper|C_gripper_To_close|True],[Gripper|S4_gripper_Closed|True])
Conveyor|C_conveyer_run|False
Gripper|C_gripper_To_close|False
([Gripper|C_gripper_To_close|False],[Gripper|S4_gripper_Closed|False])
nveyor|S2_conveyer_running|False])
Gripper|S4_gripper_Closed|True
Conveyor|S2_conveyer_running|False
Gripper|S4_gripper_Closed|False
pper_Closed|False, Gripper|S4_gripper_Closed|True],[Gripper|C_gripper_Go_down|False])
down|True])
Gripper|C_gripper_Go_down|False
alse],[Gripper|S3_gripper_At_bottom|False])

Fig. 3. Process model extracted using alpha algorithm in ProM

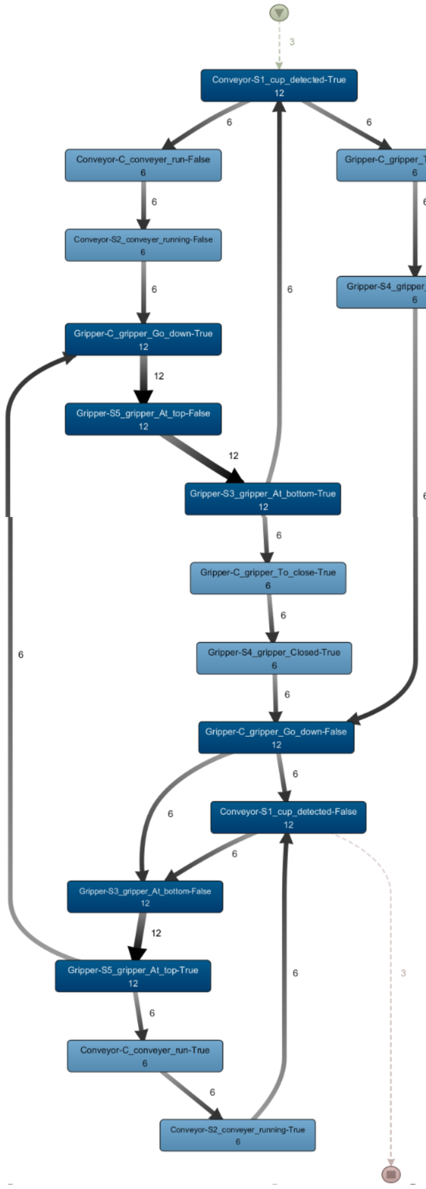
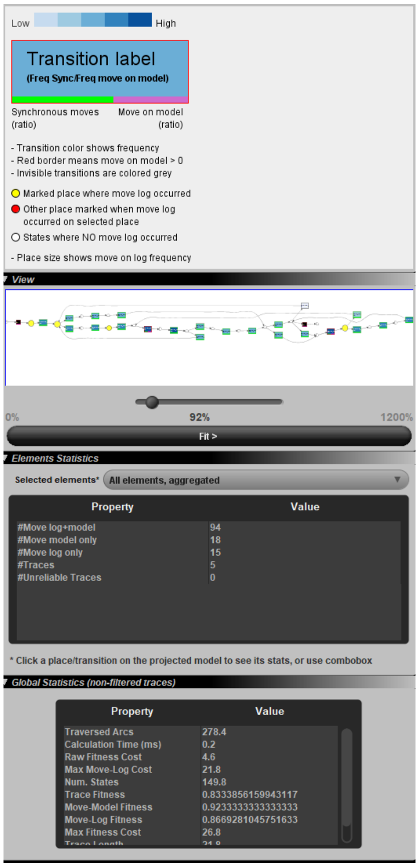

3
Conveyor-S1_cup_detected-True
12
Conveyor-C_conveyer_run-False
6
Gripper-C_gripper_To_close-False
6
Conveyor-S2_conveyer_running-False
6
Gripper-S4_gripper_Closed-False
6
Gripper-C_gripper_Go_down-True
12
Gripper-S5_gripper_At_top-False
12
Gripper-S3_gripper_At_bottom-True
12
Gripper-C_gripper_To_close-True
6
Gripper-S4_gripper_Closed-True
6
Gripper-C_gripper_Go_down-False
12
Conveyor-S1_cup_detected-False
12
Gripper-S3_gripper_At_bottom-False
12
Gripper-S5_gripper_At_top-True
12
Conveyor-C_conveyer_run-True
6
Conveyor-S2_conveyer_running-True
6

Fig. 4. Process model extracted using fuzzy miner in Disco

algorithm. The fuzzy model can be approximated by changing its 'Activity' and 'Path' detail from 0 to 100 percentage and the thickness of edges in the graph explains the number of times the particular event to another event is occurred. The fuzzy model is difficult to convert to other process modelling languages but its representation is easy to understand system behavior.

D. Conformance checking

Conformance checking is used to identify the deviation of the model by replaying a scenario on a built model. The obtained process model is compared with the event log of the same simulation model or real system. Conformance checking can be seen in two perspectives, i.e., how process model deviates from log or how event log deviates from the process model. The first one helps in fixing the process model and second one explains the error occurred in the simulation system or real system. There are several algorithms for conformance analysis like checking causal footprint, token-based replay, aligning observed and modelled behaviour, etc.



- Transition color shows frequency
- Red border means move on model > 0
- Invisible transitions are colored grey

○ Marked place where move log occurred
● Other place marked when move log occurred on selected place
○ States where NO move log occurred

- Place size shows move on log frequency

Elements Statistics

Selected elements* All elements, aggregated

Property	Value
#Move log+model	94
#Move model only	18
#Move log only	15
#Traces	5
#Unreliable Traces	0

* Click a place/transition on the projected model to see its stats, or use combobox

Global Statistics (non-filtered traces)

Property	Value
Traversed Arcs	278.4
Calculation Time (ms)	0.2
Raw Fitness Cost	4.6
Max Move-Log Cost	21.8
Num. States	149.8
Trace Fitness	0.8333856159943117
Move-Model Fitness	0.9233333333333333
Move-Log Fitness	0.8669281045751633
Max Fitness Cost	26.8
Trace Length	21.8

Fig. 5. Conformance checking using ProM

Checking causal footprint method which is easy to compare footprint matrices of the event log with the existing reference model and fitness of the model can be measured by checking the deviation in two dependent events. This method does not consider the frequency of trace and it compares the event log with Petri net process model notation. Fitness of the model varies from value 0 to 1. If the fitness value of the process model is 1 then everything seen in the event log is possible. In order to consider frequency of trace to account, a basic token replay approach is used and it identifies the deviation and fitness by analyzing the missing and remaining token after replaying each trace on the reference process model. If there is no missing and remaining token on a modeled Petri net then there is no deviation otherwise it does not conform to the derived process model. Token replay consists of following disadvantages : All transitions in Petri net should be uniquely labeled otherwise may be it choose wrong path and give wrong measure of fitness, Token flooding is another problem because whenever there is no transition it adds more token and atlast every transition will be triggered and when local decisions about the path misleads the fitness measure won't be reliable. Advanced method for analysing conformance is done using alignments and the problems which present in Basic token replay and checking causal footprints never occur in this method. Conformance analysis using alignments is independent of process model notation and it identifies optimal alignment using user defined cost function.

The existing event log is added with noise and the derived process model is used for conformance analysis. The Replay event log on Petri net for conformance checking and its analysis given by the ProM is shown in the Figure 5. The places and transitions where the deviation occurred is explained by this method. It also provides elements statistics and global statistics which helps to get a complete picture of the deviations and other metrics like fitness of the process model. There are different plugins available in ProM which can be used for measuring precision (avoid under-fitting), generalization (avoid over-fitting), fitness (explain observed behaviour) and simplicity. There is no such thing as the best process model because each one will have a different better metric over one another. One need to identify the relevant metric for the analysis and the model which performs better in all those metrics can be selected as the appropriate model.

IV. Conclusion and future work

There are several applications based on process mining techniques in the field of industrial control systems. The different process discovery algorithms help to implement the model in various notations and each discovery algorithm has its own advantages and disadvantages. The appropriate model is selected by considering relevant metrics like fitness, precision, generalization, simplicity etc. The conformance checking in process mining is considered as the most important feature which identifies the deviations by comparing event log and reference model.

The process model extraction using discovery algorithms and conformance checking opens a wide range of opportunities in industrial control systems. The extracted process model derived from event log can be used for the following purposes: generation of monitors in IEC 61499 standard can be used for embedding and monitoring closed-loop system in real-time [24], verification of the conformity of the control system to certified with the derived process model, real-time verification of event log to determine whether the system deviates from its actual process , re-implementing controller design and migration from legacy control systems to the IEC 61499 standard. It is possible to incorporate the process model extraction method with the IEC 61499 tool chain [25] for automatic verification and validation of closed-loop control systems using CTL or LTL specifications. The automatic generation of plant model from this derived reference process model could be considered as the next step in future.

V. Acknowledgements

This work was sponsored, in part, by the H2020 project 1-SWARM co-funded by the European Commission (grant agreement: 871743).

References

[1] W. Van Der Aalst, "Process mining: Overview and opportunities," *ACM Transactions on Management Information Systems (TMIS)*, vol. 3, no. 2, pp. 1–17, 2012.

[2] W. M. Van der Aalst, *Process mining: data science in action*. Springer, 2016.

[3] W. Van Der Aalst, "Process mining," *Communications of the ACM*, vol. 55, no. 8, pp. 76–83, 2012.

[4] C. A. Petri, "Kommunikation mit automaten," 1962.

[5] W. v. d. Aalst, A. Adriansyah, A. K. A. d. Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. v. d. Brand, R. Brandtjen, J. Buijs *et al.*, "Process mining manifesto," in *International conference on business process management*. Springer, 2011, pp. 169–194.

[6] E. A. Lee and S. A. Seshia, *Introduction to embedded systems: A cyber-physical systems approach*. MIT Press, 2017.

[7] "IEC 61499-1: Function Blocks Part 1: Architecture," 2012.

[8] L. H. Yoong, P. S. Roop, Z. E. Bhatti, and M. M. Kuo, "Verification of function blocks," in *Model-Driven Design Using IEC 61499*. Springer, 2015, pp. 123–136.

[9] N. Khajehzadeh, "Data and process mining applications on a multi-cell factory automation testbed," Master's thesis, 2012.

[10] D. Myers, S. Suriadi, K. Radke, and E. Foo, "Anomaly detection for industrial control systems using process mining," *Computers & Security*, vol. 78, pp. 103–125, 2018.

[11] D. Myers, "Detecting cyber attacks on industrial control systems using process mining," Ph.D. dissertation, Queensland University of Technology, 2019.

[12] J. Abonyi and G. Dorgo, "Process mining in production systems," in *2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES)*. IEEE, 2019, pp. 000 267–000 270.

[13] W. Van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE transactions on knowledge and data engineering*, vol. 16, no. 9, pp. 1128–1142, 2004.

[14] J. Otto, B. Vogel-Heuser, and O. Niggemann, "Online parameter estimation for cyber-physical production systems based on mixed integer nonlinear programming, process mining and black-box optimization techniques," *at-Automatisierungstechnik*, vol. 66, no. 4, pp. 331–343, 2018.

[15] J. Theis, I. Mokhtarian, and H. Darabi, "Process mining of programmable logic controllers: Input/output event logs," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 216–221.

[16] A. Farooqui, K. Bengtsson, P. Falkman, and M. Fabian, "Towards data-driven approaches in manufacturing: an architecture to collect sequences of operations," *International Journal of Production Research*, vol. 58, no. 16, pp. 4947–4963, 2020.

[17] M. Xavier, J. Håkansson, S. Patil, and V. Vyatkin, "Plant model generator from digital twin for purpose of formal verification," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021, pp. 1–4.

[18] "Prom," https://www.promtools.org/.

[19] B. F. Van Dongen, A. K. A. de Medeiros, H. Verbeek, A. Weijters, and W. M. van Der Aalst, "The prom framework: A new era in process mining tool support," in *International conference on application and theory of petri nets*. Springer, 2005, pp. 444–454.

[20] "Disco," https://fluxicon.com/disco/.

[21] C. W. Günther and A. Rozinat, "Disco: Discover your processes." *BPM (Demos)*, vol. 940, no. 1, pp. 40–44, 2012.

[22] B. F. Van Dongen, A. Alves de Medeiros, and L. Wen, "Process mining: Overview and outlook of petri net discovery algorithms," *transactions on petri nets and other models of concurrency II*, pp. 225–242, 2009.

[23] J. C. Buijs, B. F. v. Dongen, and W. M. van Der Aalst, "On the role of fitness, precision, generalization and simplicity in process discovery," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2012, pp. 305–322.

[24] M. Wenger, A. Zoitl, and J. O. Blech, "Behavioral type-based monitoring for iec 61499," in *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2015, pp. 1–8.

[25] M. Xavier, S. Patil, and V. Vyatkin, "Cyber-physical automation systems modelling with iec 61499 for their formal verification," in *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, 2021, pp. 1–6.