PROCESS MINING OF PRODUCTION MANAGEMENT DATA FOR IMPROVEMENT OF PRODUCTION PLANNING AND MANUFACTURING EXECUTION

Gašper Mušič[(a)], Primož Rojec[(b)]

[(a)]University of Ljubljana, Faculty of Electrical Engineering, Ljubljana, Slovenia
[(b)]INEA d.o.o, Ljubljana, Slovenia

[(a)]gasper.music@fe.uni-lj.si,[(b)]primoz.rojec@inea.si

ABSTRACT

The paper deals with analysis of data captured within production information systems. Generally, a large amount of data is acquired and stored within these systems, but only a few data are used to support decisions on the production control level. This can be improved by using advanced data analysis techniques, that are capable of building models of other meaningful data representations. Process mining is a technique that results in a discrete state-transition model that can be interpreted as a Petri net. Such a model can be used to improve the understanding of manufacturing processes, improve the processes and asses their conformance to desired operation. The paper presents results of a case study, where a number of product specific routing data were recorded and analyzed in order to detect similarities that would allow for optimization of work order processing.

Keywords: Process mining, Petri nets, simulation, optimization

1. INTRODUCTION

Information technology has a considerable impact on the efficiency and quality of manufacturing especially in terms of enabling better control and optimization. However, the use of computers was in the past due to several reasons limited to the support of business functions on the one hand, and to low level machine and process control on the other. Only a couple of decades ago computers and information technology started to penetrate also into production control level where scheduling, dispatching, plant wide optimizations and coordinations are typically performed. A standard software application in this area is a Manufacturing Execution System (MES), which can be provided as a commercial software package by one of the specialized software companies or can be tailor made for the specific production environment. A common characteristic of these tools is that they are able to collect a vast amount of data and present it in various forms, but are relatively week in offering more tangible support for decision, optimization and control. And if there are functions which allow advanced data processing and analysis of the production process, only experts are able to extract valuable information. One can speak about a gap where we have plenty of data about the production on the one side of the gap and very few if any advices (suggestions) how to (re)act in order to achieve better production results on the other side of the gap.

Especially in the plant-wide control in manufacturing the existing challenges require a form of technical intelligence that goes beyond simple data, through information to knowledge (Morel, Valckenaers, Faure, Pereira and Diedrich 2007). Available models and standards are merging traditionally disparate functions and systems across the enterprise. The corresponding information technology solutions allow for access of the right information, in the right place, at the right time and in the right format. But the extraction of the knowledge from the large amounts of collected data and its integration in the production control scheme remains a challenge. The integration of data and process mining methods (Choudhary, Harding and Tiwari 2009, van der Aalst, Pesic and Song 2010, van der Aalst 2011) into the decision making on the production control level is required. The resulting knowledge in the form of static and dynamic models will facilitate new opportunities for collaboration throughout the plant, and across the supply chain. This will enable to meet the increasing demands on flexibility and reactivity within the Intelligence in Manufacturing (IIM) paradigm. The main disadvantage of these approaches is huge complexity, the need to cope with an enormous number of details and low robustness of solutions to changes in the production.

Among modelling formalisms suitable for description of systems with highly parallel and cooperating activities, Petri nets are perhaps the most widely used one. With Petri nets, production systems' specific properties, such as conflicts, deadlocks, limited buffer sizes, and finite resource constraints can be easily represented in the model (Tuncel and Bayhan 2007). The simplicity of model building, the possibility of realistic problem formulation as well as the ability of capturing functional, temporal and resource constraints within a single formalism motivated the investigation of Petri net based knowledge extraction methods. The main results have been achieved in the field of Workflow management where the related Process Mining methodology has been developed (van der Aalst 2011).

The paper investigates the applicability of process mining methodology in the context of production planning and optimization. This is one of the fields where information technology has an immediate and considerable impact on the efficiency and quality of production control and related manufacturing processes. A case study is presented, where a large set of product routing data was analyzed by basic process mining algorithm and various uses of extracted information were proposed to improve the production planning and manufacturing execution.

2. PROCESS MINING

Information systems are becoming closely interlinked with supported operative processes. This results in logging a high number of events within these systems. Despite the large quantity of available data, organizations have difficulties in acquiring useful information from that data. The aim of process mining is to use the available data to obtain information about underlying processes, e.g., to automatically discover a process model by observing the archived data only.

Process mining represents a technique of obtaining useful process related information from event logs and extends the approaches generally found within Business Process Management (BPM). BPM is a discipline that combines information technology and management science and combines the resulting knowledge for management of business processes.

The relevance of Process mining can be illustrated through BPM life cycle, shown at Figure 1. The design phase creates the process, which becomes operational in the phase of configuration/implementation. Then starts the monitoring phase where the processes are observed to identify required changes. Some of these changes are implemented in the adjustment phase where the process and the related software are not re-designed but only adapted and additionally configured if necessary. In the diagnosis/requerements phase the process is assessed, which can trigger a new cycle in the BPM life cycle.

As shown in Figure 1 models play an important role in design phase and in configuration/implementation phase, while data are important in the monitoring phase and in diagnostic phase. The practice shows that diagnostic phase is not systematically and continuously supported and only some process changes are able to trigger a new life cycle iteration. Process mining enables to establish a true BPM life cycle loop. The archived data contain information that can be used to gain a better insight into the actual process, which means the eventual deviations can be analyzed and the models can be improved.

Process mining can be positioned in between machine learning and data mining on the one hand and between process modelling and analysis on the other (van der Aalst 2011). It aims to identify, monitor and improve real processes by acquiring knowledge from archived data in the contemporary information systems.

Information systems in support of production processes that collect a vast amount of data are ERP systems (SAP Business Suite, Oracle, in Microsoft Dynamics NAV), PDM systems, MES systems. Detailed information about occurring events is stored, which is shown in

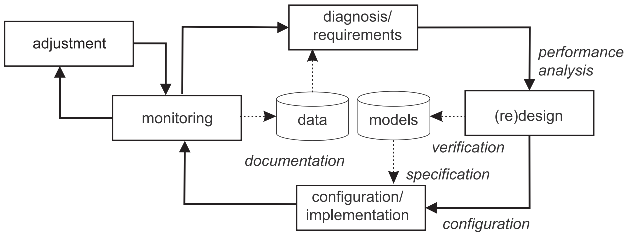

Figure 1: BPM life cycle and the use of process models (van der Aalst 2011)

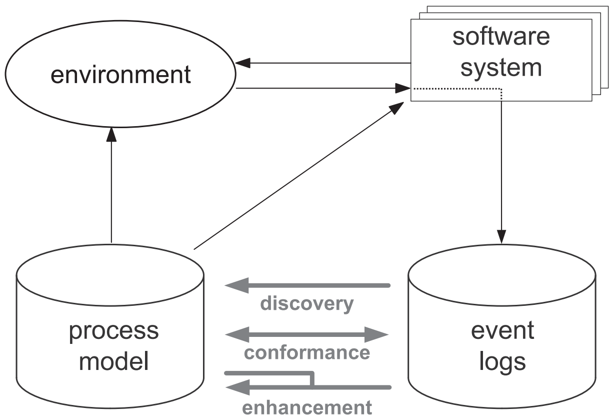

Figure 2: Types and use of process mining (van der Aalst 2011)

Figure 2 under the term event logs. Most information systems collect the data in unstructured form and some preprocessing is required to extract event data. In the following we assume that there is possible to sequentially record events such that each event refers to an activity (i.e., a well-defined step in the process) and is related to a particular case (i.e., a process instance) (van der Aalst 2011). Such a log can then be analyzed by a corresponding process mining algorithm.

Event logs can be used for three types of process mining as indicated in Figure 2. The first is model discovery or process identification. A process model is automatically generated from an event log without any a-priori information. In the sequel such an algorithm is described that generates a Petri net model based on event-log information. The second type of process mining deals with conformance testing. An existing model is compared to the event log in order to detect any deviations. The deviations can be located and also quantified to estimate the severity of changes in the process behavior and related risks. The third type of process mining is related to model improvement or enhancement. An existing model is improved or extended based on the event log information. This can be connected to improving the reflection of reality by the model or to adding new model perspectives. E.g., the initial model may only concern the sequential order of process activities. By adding time information from some event log, such model can be used also to analyze performance, detect bottlenecks, etc.

In general, different mining perspectives can be identified:

- the control-flow perspective is focused on the ordering of activities;
- the organizational perspective deals with resources in the background of recorded activities;
- the case perspective focuses on properties of recorded case instances;
- the time perspective deals with event occurrence times and event frequencies; these can be used to assess performance, locate bottlenecks, measure the utilization, etc.

While the above description implies that the process mining is performed in the frame of an off-line archived data analysis, it is also used in several operational support approaches. E.g., the deviations from the nominal process behaviour can be detected on-line, while the process is running. Process mining is therefore not only relevant in design and diagnosis phases but also in monitoring and adjustment phases (van der Aalst, Pesic and Song 2010).

3. PETRI NETS

Petri nets were introduced by C. A. Petri in his PhD thesis in 1962 and initially had the form of Condition/Event Systems with only binary markings and simple arcs. Later a number of modifications of the basic system model was introduced, including integer markings and weighted arcs. The resulting Place/Transition Petri nets became a central model, which is well explored in terms of analysis and synthesis techniques (Desel and Reisig 1998).

3.1. Place/Transition Petri nets

Informally, a Place/Transition (P/T) Petri net is defined as a bipartite graph, consisting of two types of nodes: places, typically drawn as circles, and transitions, typically represented by bars or rectangles. Nodes are interconnected by directed arcs, which either originate from a transition and end in a place or reversed, i.e., an arc always connects two nodes of different types. An arc can be weighted, which is an abbreviated representation of a set of parallel arcs. An example of a P/T Petri net is shown in Figure 3.

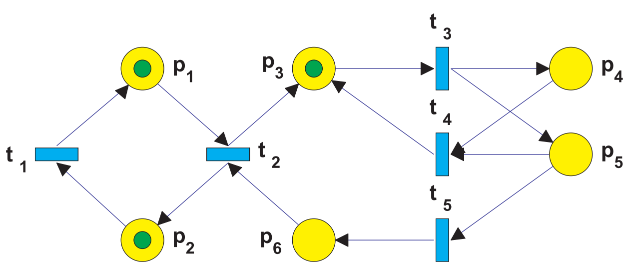

Figure 3: A Place/transition Petri net

Formally, a $PN = (\mathcal{N}, M_0)$ is a P/T Petri net system, where: $\mathcal{N} = (P, T, Pre, Post)$ is a P/T Petri net structure:

- $P = \{p_1, p_2, \ldots, p_k\}, k > 0$ is a finite set of places.
- $T = \{t_1, t_2, \ldots, t_l\}, l > 0$ is a finite set of transitions (with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$).
- $Pre : (P \times T) \rightarrow \mathbb{N}$ is the pre-incidence function and defines weighted arcs between places and transitions. It can be represented by a matrix whose element $Pre(p, t)$ is equal to the weight of the arc from p to t. When there is no arc between the given pair of nodes, the element is 0.
- $Post : (P \times T) \rightarrow \mathbb{N}$ is the post-incidence function, which defines weights of arcs from transitions to places. It can be represented by a matrix whose element $Post(p, t)$ is equal to the weight of the arc from t to p or 0 when there is no arc between the given pair of nodes.

$M : P \rightarrow \mathbb{N}$ is the marking of place $p \in P$ and defines the number of tokens in the place p. Net marking M can be represented as a $k \times 1$ vector of integers. M_0 is the initial marking of a P/T Petri net.

Functions Pre and $Post$ define the weights of directed arcs, which are represented by numbers placed along the arcs. In the case when the weight is 1, this annotation is omitted, and in the case when the weight is 0, the arc is omitted. Let ${}^\bullet t_j \subseteq P$ denote the set of places which are inputs to transition $t_j \in T$, i.e., there exists an arc from every $p_i \in {}^\bullet t_j$ to t_j. Transition t_j is enabled by a given marking if, and only if, $M(p_i) \geqslant Pre(p_i, t_j), \forall p_i \in {}^\bullet t_j$. An enabled transition can fire, and as a result removes tokens from input places and creates tokens in output places. If transition t_j fires, then a new marking is determined by $M'(p_i) = M(p_i) + Post(p_i, t_j) - Pre(p_i, t_j), \forall p_i \in P$.

The switching rule of a Petri net is given as follows:

i) a transition is enabled if each of the input places of this transition is marked with at least as many tokens as the weight of the corresponding arc,

ii) an enabled transition may or may not fire, which may depend on an additional interpretation,

iii) a firing of a transition is immediate (includes no delay) and removes a number of tokens equal to the arc weight from each of the input places and adds as many tokens to each of the output places as the weight of the corresponding arc.

The basic Place/Transition Petri net model can be extended in different ways, leading to other Petri net classes, e.g. timed models (Bowden 2000).

4. PETRI NETS AND PROCESS MINING

Process Discovery is a central task within process mining. It aims in constructing a process model based on event log. It deals with control-flow perspective and can be defined as follows: Let L denote an event log. A process discovery algorithm is a function that maps L onto a process model that is able to reproduce the behaviour captured in the event log.

This definition does not specify the kind of modelling formalism used to represent the process model. E.g.,

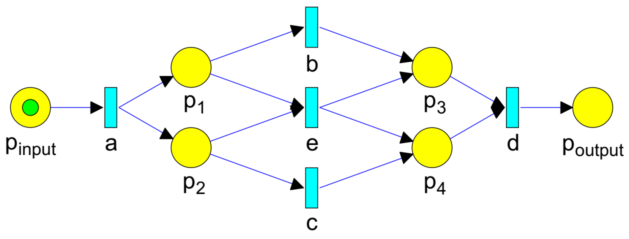

Figure 4: WF-net N_1 discovered from event log L_1

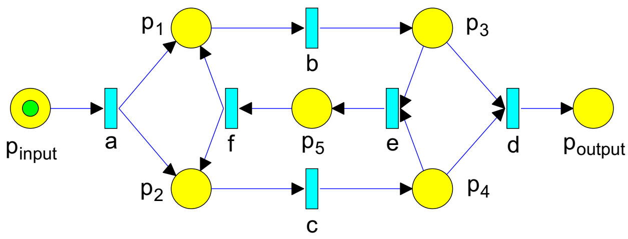

Figure 5: WF-net N_2 discovered from event log L_2

BPMN, EPC, YAWL, or Petri nets can be used. In the following a Petri net representation of the model will be used. Additionally, a simple event log will be used, which is a multi-set of traces over a set of activities $\mathcal{A}$, i.e., $L \in \mathbb{B}(A^*)$. Example of such a log is:

$$L_1 = [< a, b, c, d >^3, < a, c, b, d >^2, < a, e, d >] \quad (1)$$

L_1 is a simple event log that describes six cases. A process discovery algorithms should generate a PN, that is able to replay all the sequences in L_1. The PN should belong to the class of sound Workflow net (WF-net), which is safe and contains one input place ($\bullet i = \emptyset$), and one output place ($o\bullet = \emptyset$) (van der Aalst, Weijters and Maruster 2004).

The problem of process discovery can then be reformulated as: A process discovery algorithm is a function γ, that maps a log $L \in \mathbb{B}(A^*)$ into a marked PN $\gamma(L) = (\mathcal{N}, M)$. N is a sound WF-net and all traces in L correspond to possible firing sequences of $(\mathcal{N}, M)$.

Based on L_1, such an algorithm discovers the net shown in Figure 4. There the so called α-algorithm was used (van der Aalst, Weijters and Maruster 2004), which is a basic process discovery algorithm. It is able to discover PN models from a large class of event logs, although it also has some limitations (van der Aalst 2011).

It can be observed that all the possible firing sequences of PN in Figure 4 match the sequences in L_1. In general this is not the case and the PN generated by the α-algorithm is able to generate a larger set of event sequences. E.g., given the log:

$$\begin{aligned} L_2 = [&<a,b,c,d>^3, <a,c,b,d>^4,\\ &<a,b,c,e,f,b,c,d>^2, <a,c,b,e,f,b,c,d>^2,\\ &<a,b,c,e,f,c,b,d>, <a,b,c,e,f,b,c,e,f,c,b,d>] \end{aligned} \quad (2)$$

the WF-net N_2 shown in Figure 5 is generated. The shown net can generate all the traces from L_2 but also others. E.g., the trace $< a, b, c, e, f, c, b, d >$ is feasible but is not contained in L_2. This complies to the above definition as it is required only that the net is able to generate traces in L while no restrictions related to generation of other traces were given.

Even when the above process discovery algorithm definition is related to WF-nets, other models could also be used. E.g., the WF-net in Figure 4 can be translated into an equivalent BPMN model or equivalent EPC, UML activity diagrams, YAWL models, etc. (van der Aalst 2011).

5. PROCESS MINING EXAMPLE

To asses the usability of α-algorithm in conjunction to real production data a testing set of data was collected in cooperation with INEA company. The set comprises technological routing data of over 30.000 door side panel items that were processed in the frame of furniture production.

The data record for an item consists of a set of attribute values followed by a set of time values that correspond to duration of the manufacturing operations. A fixed record length is maintained and in case the attribute is not relevant or certain operation is not performed on an item, the value is NULL.

An item is described by 10 attributes, shown in Table 1. In the routing part there are altogether 67 possible manufacturing operations, such as *Edge processing, Lock and hinge holes drilling, Chipboard panel assembly, Honeycomb panel assembly*, etc. The operation duration table is relatively sparse as only a small subset of operations is typically performed during an item processing.

To derive a suitable event log the data was first preprocessed in a way which separated individual items data and build corresponding cases. Production of each item now represents a production case with operation sequence and timing. In the following only ordering data were used, although the analysis of timing data was later added, too. A simple analysis shows that nearly 98 % of recorded operation sequences only contains two or three operations. These are repeated very often, up to more than 7000 repetitions as shown in Figure 6, while most of the other sequences repeat less than 100 times. A better view is obtained if only different sequences are considered. This is illustrated in Figure 7, which shows the distribution of the lengths (number of operations) of the different processing procedures that correspond to the items in the database.

Table 1: Attributes of manufactured items

Attribute	Meaning
Item ID	Unique item designator
Standard	Internal standard
Core	Type of the panel core
Surface type	Panel surface material
Wood	Panel surface appearance
Surface finish	Type of surface finish
Edge	Edge designation
Groove	Type of additional edge processing
Thickness	Panel dimension
Width	Panel dimension

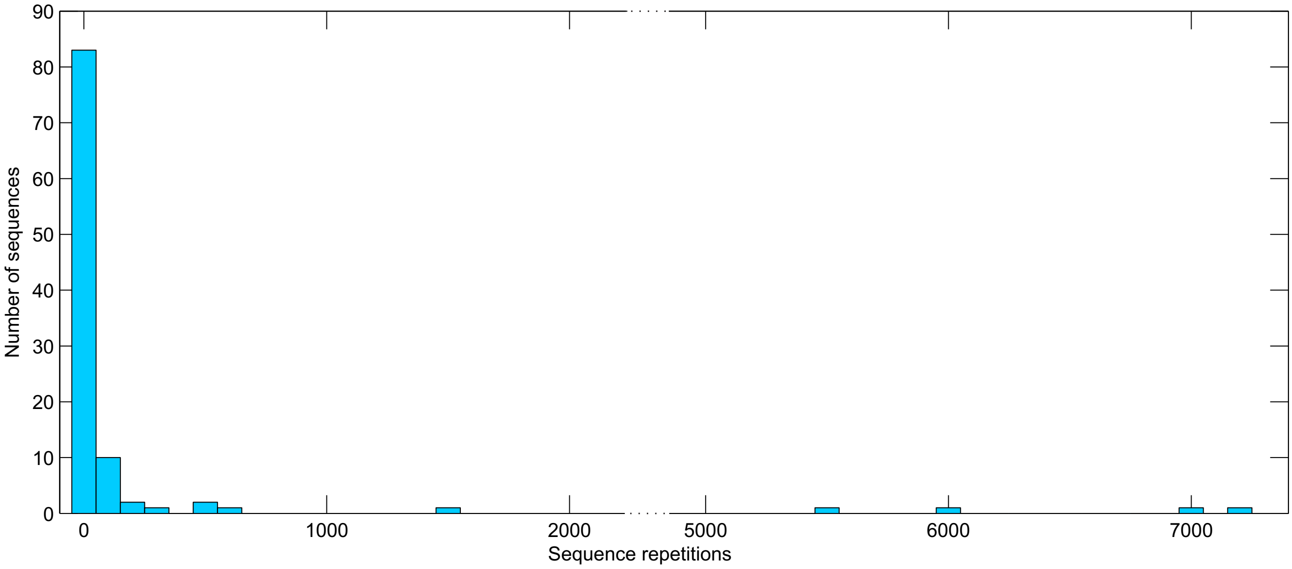

Figure 6: Distribution of the sequence repetitions

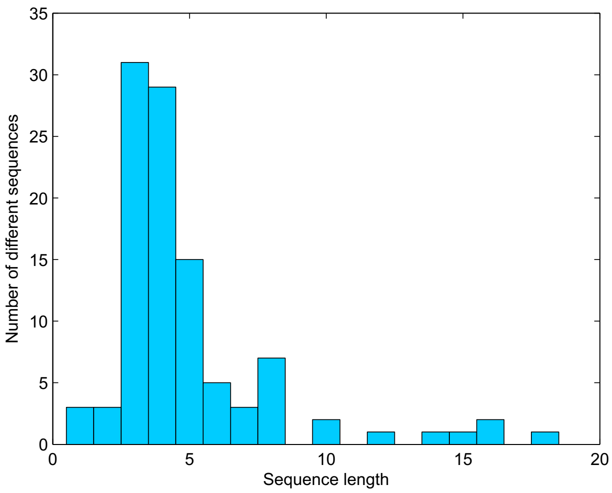

Figure 7: Distribution of the recorded operation sequence lengths in the database

An initial process mining analysis was performed to check the variability of production sequences. Figure 8 shows the resulting model based on 10.000 cases. Clearly we have a large repeatability in the production sequences, i.e., close similarities in item routings. This information could be used to improve the efficiency of work order processing, e.g., by grouping similar items together in larger batches.

Despite the relatively condensed representation of the large number of cases in Figure 8, the analysis of derived net shows that the model is not useful. In contrast to the initial assumption on sound WF-net the derived net does not belong to that class. This can be observed from the graphical representation or shown by analysis of the reachable markings. The analysis shows some non-safe places as well as places that do not receive tokens at all, which indicates dead transitions.

These observations are not a consequence of the anomalous working of the applied α-algorithm but the consequence of the inherent *representational bias* of the algorithm. Such a representational bias is required by any process discovery technique and helps limiting the search space of possible candidate models (van der Aalst 2011). In case of the α-algorithm the technique is able to adequately capture concurrency, but can produce models with livelocks or deadlocks. Limiting the search space to only sound models would limit the expressiveness of the modelling language.

In the addressed case the concurrency is a desired property, as can be exploited to increase equipment utilization. Therefore we tried to circumvent the deficiency of the algorithm by an appropriate data segmentation. First, the number of sequence appearances is considered. Figure 7 shows there are only 8 different sequences with length of 10 or more operations. More detailed analysis shows that these sequences appear at 19 cases all together. These cases are considered exceptions and are filtered out.

The Petri net model obtained by process mining on the reduced set of data is still not acceptable, therefore additional segmentation based on the item attribute values was performed. Here the main problem was a rather poor quality of the data with many missing or questionable attribute values. After a set of experiments with different selection methods a suitable subsets of attributes were determined, which can be used to classify cases into separate groups that can be represented by a set of simple sound WF-type Petri nets. E.g., a combination of *Core, Surface type, Wood, and Surface finish* attributes enables to designate corresponding item routing sequences.

Next, a set of data was used to simulate an on-line conformance testing. At each item processing, the matching between previously developed models and the manufacturing operation sequence was observed. A deviation raised a warning. In practical implementation, such a warning would be used to notify the production operating personnel. Different metrics have been tested, which enable to adjust the warning action according to the degree of deviation.

The main goal of the investigation is to explore the

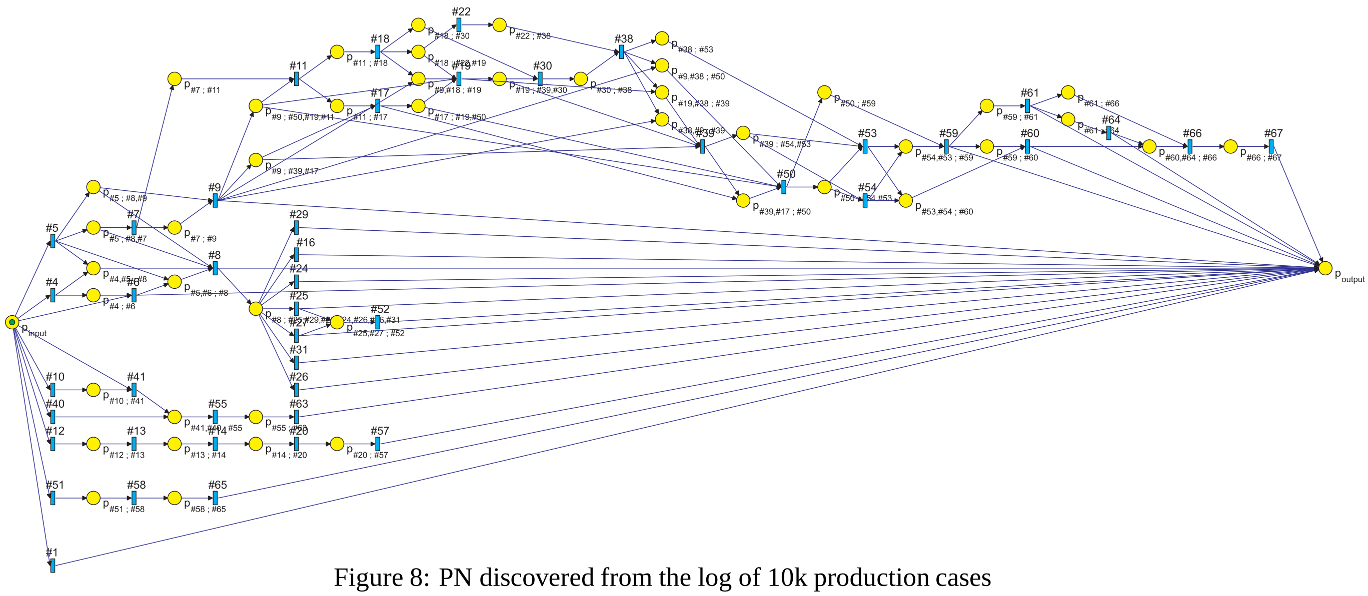

Figure 8: PN discovered from the log of 10k production cases

similarities among processed items in terms of temporal quantities. To achieve this also the time information was included in the models. This way the models can be refined to reflect also the time perspective, which enables to identify groups of items with similar processing times. This information can be used to improve production planning and scheduling.

6. CONCLUSIONS

The presented results indicate that the basic process mining algorithm can be used to analyze certain types of production management data. In particular, the product routing information and corresponding operation durations were used to analyze a large set of production data. Careful data preprocessing and segmentation were required to obtain useful models. The analysis showed similarities and relations among different groups of products that were hardly visible from the data directly, due to the very large number of different production items.

The obtained information will be used to improve the production planning and manufacturing execution.

ACKNOWLEDGMENTS

The presented work has been partially performed within the Competence Centre for Advanced Control Technologies, which is partly financed by the Republic of Slovenia, Ministry of Education, Science, Culture and Sport and European Union, European Regional Development Fund. The support is gratefully acknowledged.

REFERENCES

Bowden, F. D. J., 2000. A brief survey and synthesis of the roles of time in petri nets, *Mathematical & Computer Modelling*, 31, 55–68.

Choudhary, A. K., Harding, J. A. and Tiwari, M. K., 2009. Data mining in manufacturing: a review based on the kind of knowledge, *Journal of Intelligent Manufacturing*, 20, 501—521.

Desel, J. and Reisig, W., 1998. Place/transition petri nets, *in* W. Reisig and G. Rozenberg (eds), *Lectures on Petri Nets I: Basic Models*, Vol. 1491 of *Lecture Notes in Computer Science*, Springer, pp. 122—-173.

Morel, G., Valckenaers, P., Faure, J., Pereira, C. and Diedrich, C., 2007. Manufacturing plant control challenges and issues, *Control Engineering Practice*, 15, 1321–1331.

Tuncel, G. and Bayhan, G. M., 2007. Applications of Petri nets in production scheduling: a review, *International Journal of Advanced Manufacturing Technology*, 34, 762–773.

van der Aalst, W. M. P., 2011. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer.

van der Aalst, W. M. P., Pesic, M. and Song, M., 2010. Beyond process mining: From the past to present and future, *Lecture Notes in Computer Science*, 6051, 38–52.

van der Aalst, W. M. P., Weijters, T. and Maruster, L., 2004. Workflow mining: Discovering process models from event logs, *IEEE transactions on Knowledge and Data Engineering*, 16 (9), 1128–1142.

AUTHORS' BIOGRAPHIES

GAŠPER MUŠIČ received B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from the University of Ljubljana, Slovenia in 1992, 1995, and 1998, respectively. He is Associate Professor at the Faculty of Electrical Engineering, University of Ljubljana. His research interests are in discrete event and hybrid dynamical systems, supervisory control, planning, scheduling, and industrial informatics. His Web page can be found at `http://msc.fe.uni-lj.si/Staff.asp`.

PRIMOŽ ROJEC received B.Sc. and M.Sc. degrees in electrical engineering from the University of Ljubljana, Slovenia in 1990 and 2004, respectively. He works at Inea engineering company. His research interests are in planning, scheduling, and manufacturing execution systems.