Analyzing Machine Performance Using Data Mining

Milan Pospíšil, Vladimír Bartík, Tomáš Hruška
Department of Information Systems,
Brno University of Technology
Brno, Czech Republic
{ipospisil, bartik, hruska}@fit.vutbr.cz

***Abstract*— This paper focuses on analysis of machine performance in a manufacturing company. Machine behavior can be complex, because it usually consists of many tasks. Performance of these tasks depends on product attributes, worker's speed, and therefore, analysis is not simple. Performance analysis results can be used for different purposes. Prediction and description are typical products of data mining. Prediction should be used for online monitoring of the manufactory process and as an input for a scheduler. Description can serve as information for managers to know which attributes of products cause problems more frequently. However manufacturing processes are complex, every process is quite unique. Our long term goal is to generalize the most common patterns to build general analyzer. This task is not simple because the lack of real word data and information. Therefore this work may contribute to the other researchers in their understanding of real world manufacturing problems.**

Keywords — Process mining, data mining, manufacturing, performance analysis, simulation, prediction, monitoring, scheduling.

I. INTRODUCTION

Scheduling is an important part of business processes, mainly in manufacturing. The important input for scheduling is a correct knowledge of setup times, work times, error probabilities and its time lengths. These inputs are not always easily obtainable because times can be dependent on product attributes and worker's speed. Moreover, there are some problems resulting from practice, such as changes in time or incomplete data. Measurement of these values can also be a complex problem. This work deals with analysis of a non-trivial machine performance from a time perspective. This machine is composed of three different tasks, which are typical in this area. The first task is performed by the machine itself and the next two tasks are performed by workers. Our goal is to predict time needed to process a product on this machine.

Result of analysis could be valuable for managers, because errors and high variance tasks are not suitable for a good schedule. Managers could use the results to understand the process in detail and discover some hidden dependencies. Finally, they can use it for various improvements of the process. Also, scheduler could obtain more accurate times about machine execution in case that some interesting dependencies are found.

This work continues our research [1][2][3]. Previous work was focused on general models for predictions, whereas this paper is focused on one particular machine. We believe these case studies are important for others to understand what may be happening in the real manufacturing and how hard is to solve that problems. The long term goal is to generalize the manufacturing processes (mainly line productions) and to describe common behavioral patterns and how to solve them using datamining.

II. RELATED WORKS

Data mining of processes is not a new topic. Group around W. M. P. Van der Aalst et al. has done a lot of work around Process Discovery. Original methods focused on mining process model from a process log. Later, they added other perspectives – social, decision rules and time perspective [4][5][6]. Later, Rozinat also dealt with simulations, which are used for predictions [7][8]. Our first work [9] continued in their research, but it was much more focused on performance measurement and time perspective of processes. Grigori [10][11] also did some performance measurement, but their work was focused on a process as a black box, whereas our approach deals with process model and queues.

However, these papers are not focused on prediction of time needed by a machine to make an operation on some product according to various properties of that product. This paper is mainly focused on analysis of performance measurement of a non-trivial machine workplace and it shows typical practical problems and their possible solutions. A new metric to count similarity of product clusters is also presented in this paper. Results of this analysis could be further used for scheduling and simulations.

We have analyzed some another workplaces in [1] and [2]. This paper is focused only on one specific workplace, so there is no need for a reader to know about our previous work. Moreover, different approach has been used for analysis in this paper.

III. PROBLEM DESCRIPTION

We cannot give reader a precise description because of the information protection of our manufacturing company but we will give enough information to understand our results and their contribution. The workplace is composed of three different tasks. It is a machine that needs human participation. First, several products are clustered together to allow the

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

machine operate products more effectively (without adjustments between individual products). Second task consists of moving products from the cluster (palette) into the machine. Third task is a task performed by the machine, which makes some operations over product. These tasks will be called as: Preparation, Insertion and Machine task.

It is important for manager to know the performance of this workplace because it is one of the bottleneck machines. It is not easy to schedule it because its work variance is quite large. It also cannot be easily measured by a normalizer because we assume that times and errors are dependent on attributes of products. If we are able to discover these dependencies, managers will be able to better schedule their manufactory process. Also the outlier detection can be valuable for them. For example, if we discover that some attributes of products lead to an outlier execution time length or error probability, they can focus on a concrete improvement in the process.

IV. Supposed Dependencies

Data mining is usually about finding dependencies but current state of the art often needs expert information as input. Every data mining method is built around a model that assumes several dependencies. This is caused by multiple dependencies and it is beyond our options to try to discover them all. For example, there can be sequence dependencies – if product A was produced and the product B after that, a sequence dependence means that time of product B also depends on times of product A and several previous products, or vice versa (in practice, there is usually reverse dependence caused by the setup time). Another problem is that company has multiple data sources and if we are not aware of potential dependences, we may not have prepared data for it. And last, data have various formats. Some data are in a relational form (constant number of attributes in a table), another are rather transactional, i.e. some attributes that are either present (in product) or not and they can be present multiple times.

A. Preparation Task

Preparation task is the most complex task. Workers prepare the cluster of products and required material. Execution time depends mainly on worker's speed, type of construction (single attribute) and needed materials. Because this is one of the first workplaces, the products are in a raw form and therefore there are no dependencies on complex attributes. They rather depend on materials. Data about materials are not in a classical relational form. Each product needs different amounts of materials and there are several material types. This is not suitable for most of classification methods. Execution time does not depend on number of products in a cluster, because the cluster is already on a palette but the cluster and material must be prepared by workers.

B. Insertion Task

When the whole cluster is prepared, the insertion task can be started. All products from the cluster are moved into the machine and processed until the cluster is empty. Insertion task is dependent only on worker's speed and type of construction, therefore this should not be complicated to analyze.

C. Machine Task

Machine task is the least important for analysis because machine works almost constantly if no error occurs. Management believes that there is no dependency between product attributes and error probability. We will check if this assumption is true.

V. Methodology

The aim of this paper is to present how data mining techniques can be used to solve some real analytical problems in a manufactory. Designing data mining methods took only 10-20% of our effort. The rest was the communications with managers, data cleaning and preparation and discussion about the quality of data, results and typical problems. This could be avoided with experience. We have chosen several data mining methods mentioned below to perform analysis.

A. K-Nearest Neighbor

K-Nearest-Neighbor is a simple classification method. It is based on finding some number (k) of similar items in the data to an unlabeled item. There exist many variations of the algorithm, we describe our variant more in detail in the result section. This method was chosen according to our experience with similar tasks where its accuracy was usually the best among various classification methods.

B. Regression Trees

Regression trees are similar to decision trees except the target attribute, which is numerical, not categorical. The nodes do not contain the most probable target class, but mean value and deviation of the target attribute. This method is much faster than kNN classification but its accuracy is quite worse. We described Regression tree more in detail in our work [3].

C. Association Rule Mining

Association rules mining is a descriptive kind of knowledge obtained by data mining methods. Its task is to find values in data, which occur frequently together. The result is a set of association rules of a form $A \Rightarrow B$, where A and B are sets of items (values in the data). As the first step, we have to discover a set of frequent itemsets, i.e. the sets of items that occur frequently in data. Afterwards, association rules are generated from them. The two best known approaches for mining the frequent itemsets are the Apriori based algorithms [12] and FP tree based algorithms [13], which is more efficient. In this work, we use association rules to discover combinations of attributes that lead to a longer time needed for preparation of a product cluster. We mainly concentrate on construction types of products, materials needed to produce them and the variability of products contained in the product cluster and its influence on time needed to prepare the cluster. To obtain association rules we use the Apriori algorithm.

VI. Data Preparation and Cleaning

Data quality is the weakest point in our analysis as usual. There are only two measurement spots (at the machine entrance and at the machine output). Time of the machine task is completely measured, but two other tasks are not. We can only obtain the Preparation and Insertion time as difference between two start times (at the entrance to the machine). Of course, this brings new problems into the data mining process – we have to distinguish between preparation times, insertion, breaks and pauses. This makes our data more complicated to be analyzed.

Fortunately, we have additional information – we know about clusters, because identifier of a cluster is available for each product. However, first insertion of a product is measured together with preparation time, because the start is measured at the beginning of the machine. There is the start information about the last product of cluster, then cluster preparation, insertion and another start. The difference represents two worker's tasks.

We have to mention also the typical problems of data preparations generally, according to our experience from many analytical tasks of various workplaces in manufactories. These problems include:

- **Incomplete measurement.** Some workplaces are measured only partially. Only start or end information is available. One of possible solutions is described in [1]. The worst scenario is when no measurement is available – only context tasks around can serve as an inaccurate substitute – if previous task ends, this could be approximate start of our task. But this is true only if no buffer and low latency is available.
- **Cluster measurement.** Sometimes setup time and errors are measured together with work time. There is a problem how to compute it when task is stochastic and based on various attributes.
- **Unmeasured sub-process.** Even if time of task is measured, task could contain subprocess with unknown execution length. In this situation we know only execution length of whole the process but sometimes we need to know the times of its component tasks to better predict real execution time with lowest variance possible.
- **Changes in time.** Real processes are not static, they change over time. We discussed changes in [1]. There are two basic solutions – adjust method to changes or ignore changes and work only with new relevant data. Selection of solution depends on the situation. If changes are slow (human workplace – learning problem), methods could be easily adjusted. On the other hand, if changes are larger, new data may be a better option. Moreover, sometimes larger changes can change only a small part of a process, and we want to keep old knowledge of unchanged data. Another problem itself is the detection of changes but this depends on data quality. We can also monitor changes to measure effect of them (how useful, fast and error-prone is a new machine). This problem will be possibly solved in our future work.

VII. Results

A. Machine Task

Machine is supposed to work constantly. To analyze this, we have ordered machine execution times ascending and grouped results by product construction attribute (which was the most determinative in regression tree – it is distinguished by colors). The result is in Figure 1. The number of executions was different for each group (construction attribute), so we normalized them. For example if one group has 200 items, another 400 items, we get only every even item in second group. Vice versa, if the group has 100 items, every item will be twice in graph. This allows us to compare time behavior of the groups.

From Figure 1 we can see that machine is usually working constantly. The graph is limited to some maximum value to ensure that visualization is clear, real graph reaches much higher times in extremes. We have also to mention the problem with breaks of employees – we cannot distinguish what is break for food and what is a problem in the process. We can suppose that breaks are nearly equally distributed over the data.

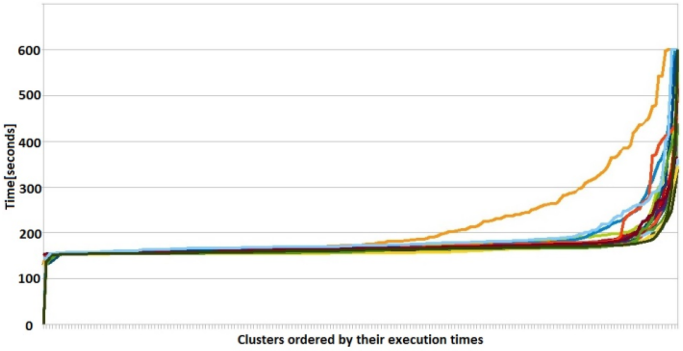

Fig. 1. Execution times of machine. Times are ordered and grouped by construction type.

We can see that in certain point (depends on attributes) there is a growth of execution time, which can be either slow or fast. We can also see how another attributes affect error probability. Because machine works usually with constant speed and when some problem occurs, the time grows up, it is possible to use regression tree for analysis. Regression tree will give us mean values (and deviations) of the tree nodes and because time is usually constant, the higher mean or deviation is caused by higher error probability.

If we want to know exact error probability of attributes combination we can also use regression tree, but with another input. Instead of time, we will use numbers 0 or 1 to represent normal execution (0) and error execution (1). A record will be considered as error if its time is higher than the standard machine execution time. The values in nodes will represent the error probability of a node (a value in a range between 0 and 1). Now we can see that nodes with higher mean and deviation also have the higher error probability.

Example of a regression tree:

All (Mean: 120s, Deviation: 70s)
    Construction: Standard (Mean: 105s, Deviation: 70s)
        Height: 130 (Mean: 100s, Deviation: 80s)
        Height: 180 (Mean: 170s, Deviation: 110s)
    Construction: Exclusive (Mean: 150s, Deviation: 95s)

We tested the predictability of the result. Data was divided into two parts – training and testing with the ratio 90:10 (the ratio was high because system will be filled with new data every day in a real environment) and we used cross-validation. We tested Regression Tree and K-Nearest-Neighbor and compared it to primitive classifier that always supposes the constant value.

The error was chosen as the criterion. We computed error as absolute value of difference between the actual and predicted value. Then we computed the average error as:

$$avgerr = \frac{|v_{actual} - v_{predicted}|}{n} \quad (1)$$

We measured average error for each method. The results are following:

TABLE I. COMPARISON OF PREDICTION METHODS ACCURACY

Method of Prediction	Average error
Regression Tree	82 seconds
K—Nearest Neighbor	74 seconds
Primitive Classifier	94 seconds

We can see that our approach is slightly better than the primitive classifier. Unfortunately, we assume that precision could not be better because it is natural for this type of problem. Machines usually work with constant time and errors are rather random. But except of accuracy improvements, we are able conclude what attribute combination leads to bigger error probability.

B. Insertion Task

We analyzed the insertion task using a regression tree and k-nearest-neighbor classifier and the results showed that there are also some dependencies. You can see it on the graph (Figure 2). Values of times are ordered ascending and grouped by construction of product (distinguished by colors - that attribute had the biggest determination power – it formed the root node of regression tree, another attributes were not determinative enough). We can see that there is no constant time typical for machines because it is not a machine workplace.

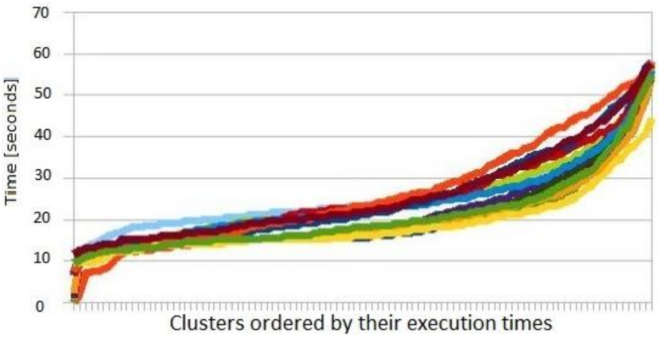

Fig. 2. Ordered times of the insertion task grouped by the construction of the product

C. Preparation Task

Preparation task is the most complicated. There are several products in a cluster that must be prepared. Preparation is dependent on construction of product and materials that must be prepared together with product. The cluster size contains between several to max. 50 products. The example of data representing a cluster is following (numbers in bracket are amounts of those materials):

TABLE II. PART OF A CLUSTER WITH PRODUCTS AND THEIR ATTRIBUTES (EXAMPLE)

Product 1	{ Construction: AAA, Material1(3), Material2(3), Material7(1) }
Product 2	{Construction: AAA, Material2 (2), Material6 (1)}
Product 3	{Construction: ABB, Material3 (1), Material5 (2)}

Usually there is a need to cluster products with the same construction together. This is present in most of the data. Typically, every cluster contains 1-3 construction types.

Data are not in a classical relational form, where the number of attributes in a table is constant. Here, every cluster has different number of products (which is not supposed to be determinative), different construction types (potentially every product could have its own construction type) and different materials (and their amounts).

We have used the k-nearest-neighbor classifier with a suitable distance metric, which is called cluster similarity.

This metric is obtained as sum of two similarity values – **type similarity** and **material similarity**. Both are in a range between 0 and 1. We always compare lead cluster (the one that needs to be predicted) to other clusters in the dataset. Then, the kNN classifier takes k clusters with highest values of similarity.

At first, we have to define the **ratio of construction type** r_c, which is obtained as:

$$r_c(clust) = \frac{n_C}{N} \quad (2)$$

where n is the number of products with construction type *C* in the cluster and *N* is the number of all products in the cluster *clust*.

Then, we have to compute the **ratio of similarity for construction type** c_sim_c as:

$$c\_sim_c(clust1, clust2) = \frac{n_C(lclust) * r_C(clust1)}{r_C(clust2)} \quad (3)$$

where *clust1* and *clust2* are two clusters, similarity of which is computed. We choose a cluster with lower value of r_c as *clust1* because we have to ensure that the value of similarity is between 0 and 1. It is multiplied by the number of products with construction *C* in the lead cluster, which is *clust1* or *clust2*.

This is computed for all construction types and the resulting construction similarity is defined as:

$$c\_sim(clust1, clust2) = \sum_{i=1}^{M} c\_sim_{Ci}(clust1, clust2) \quad (4)$$

where M is the number of various construction types, which occur in clusters.

Material similarity is computed in the same way. At first, we have to compute a sum of material amounts for the same types (material is prepared independently of the products in cluster). For example, in Table 1, the value of n_M (count of material in the cluster) for Material2 is 5. Then we use equations (2), (3) and (4). In these equations, we replace the letter C with the letter M and we obtain the material similarity m_sim.

Sum of construction similarity and material similarity is the total cluster similarity. Both similarities could be weighted to gain more importance for construction or material similarity (it is described in experiments). We selected the 50 best clusters with the highest cluster similarity and computed mean value of execution time and we used the same metrics as in previous experiments and compared it to the primitive classifier. In k-NN experiment, weights of particular similarities (construction and materials) are always 1 or 2.

We did also another approach. We used the k-NN classification only when the deviance of the selected 50 best clusters is lower than the deviance of the whole data set. It improved experiments. In results, our experiment was made without this improvement – it is marked in the experiment as „without improvement“. We have used equation (1) again to count average error. In Table 3 you can see average errors of prediction for various settings of the similarity weighting:

TABLE III. COMPARISON OF PREDICTION METHODS ACCURACY

Classifier (settings)	Average error
Primitive classifier	98 seconds
K-NN (both weights 1; without improvements)	84 seconds
K-NN (both weights 1)	76 seconds
K-NN (weights: costruction: 1, material: 2)	84 seconds
K-NN (weights: costruction: 2, material: 1)	71 seconds
K-NN (weights: costruction: 3, material: 1)	75 seconds
K-NN (weights: costruction: 2,2, material: 1)	70 seconds

We can see that construction is more important than material, but not the only important – if we set ratio of construction to 3, error increases. The minimum error was found by random repeated experiment as 2.2, but error was not below 70 seconds. There is no need to search for exact minimum, because results will probably change over time based on new data.

VIII. USE OF ASSOCIATION RULES MINING

Because association rule mining is a descriptive data mining technique, we cannot use it to predict longer preparation time of a product cluster but we can use historic data about processes to describe the reasons of production time increase. To perform this task, we use the data similar to those described in the previous section.

As it was mentioned above, the data is not in the classical relational form but this not is a problem here because association rules have been primarily designed to use for transactional databases. Here, a database consists of a set of transactions and each transaction consists of an arbitrary number of items. The size of various transactions can be different. This property is very useful in our process mining task because there are various counts of products, construction types and materials in each product cluster. On the other hand, we are currently not able to take the amount of material into account. Finding a way to extend the association rule task to process amounts is a task for our future research.

Let us describe the data used as the input of our association rule mining method. If a new product cluster is identified in the data, the time of its preparation is counted and stored. If this value is higher than a common value of preparation time (time between 20 and 60 seconds) information about this cluster will be stored into the database. Otherwise, the cluster will be ignored. Moreover, also the clusters with extremely high time (more than 30 minutes) of their preparation are also ignored because it is a high probability that these clusters have been prepared after a longer pause or a lunch break. Therefore, these records in data do not refer to a problem or delay in the product preparation process.

If the record about a product cluster is stored, the record will contain information about all product construction types and materials used in the product cluster and then, we have added several attributes about the variety of products in that cluster – this includes number of different materials and number of different construction types.

In our process mining task, we do not use a classic association rule mining task (for example, as it is known from the task of market basket analysis) because we need to get association rules in the form:

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \Rightarrow preparation\_time = \text{'long'}.$$

Because the right-hand side of this association rule is ensured by the fact that all product clusters in our dataset satisfy it, the task can be reduced to the task of mining frequent itemsets from our dataset. To obtain this set of frequent itemsets, we use the Apriori algorithm. To get only relevant results it is necessary to set the value of support – the percentage of records (product clusters), which contain all values contained in the frequent itemset. According to experiments, we have set this value to 15%.

We have collected approximately 500 records (transactions) about product clusters, average size of which is 24 items (materials and construction types).

It is necessary to make a filtering phase after the Apriori algorithm finishes because of the great amount of association rules not understandable for a human. Because some materials and construction types are used very frequently in the products, they are very frequent in a dataset describing the clusters with long preparation times but they are also very frequent in clusters with low preparation times. We are interested in association rules describing only the "long preparation" part of database.

Our filtering phase works very simply. After some association rule is obtained, have to scan the whole database and count the support of this association rule within the whole database containing all product clusters. If the value of support in the whole database is similar (or higher), the frequent itemset has no significance for the analysis of long preparation times of product clusters. In our database, we have set the condition that the association rule's support must be 10 percent higher than its support within the whole database. Due to the significant reduction of association rules it is necessary to decrease the value of minimum support threshold before the process of mining association rules stars to obtain enough rules.

After all the steps described above were performed, we have obtained a set of 105 association rules. Association rules contain various items including both materials and construction types of products. There are some interesting association rules formed by a combination of construction types count and some concrete construction type, for example:

$$constructions\_count = 1 \wedge construction\_type = \text{'ABCD 01'} \Rightarrow preparation\_time = \text{'long'}.$$

This may lead to a conclusion that the preparation of a cluster with construction type 'ABCD 01' lasts longer because this is the only one rule of this form. Moreover, we can say that it takes longer time to prepare a cluster consisting of two different construction types because there are a lot of association rules, which contain the item constructions_count = 2. But this fact does not depend on concrete construction types because none of rules containing this item contained an item of a form construction_type = x.

There we have also association rules containing various kinds of materials obtained. Kinds of materials and their combinations are contained in most of them. Additionally we had to omit the rules with materials used in almost all products (their support value is almost 100 percent) – these rules are not interesting, our need is to find rules with some more specific values leading to a decision in our business process.

No association rule contained the value of materials amount. It does not have influence on the speed of cluster preparation. On the other hand, this attribute has a greater range and discretization of this attribute can be a solution of this.

Finally, we have to mention that we should use more data to make the results usable and representative. Our dataset is not large enough to make decisions based on the association rules we have discovered.

IX. Conclusions

As we can see, the typical errors were about 70% of 100% represented by the primitive classifier. This is a significant improvement but the result is still not ideal. There are several reasons of this error. First, workers did not work in constant speed. We did not encompass workers in the analysis because we did not have the worker information available but later analysis found that worker information improves prediction and it is possible that our future data will contain the worker attributes. Second, the time measurement at the start of the machine was measured by the worker. Workers sometimes do mistakes, sometimes even intentionally (as we experienced in another workplace in the same company).

Another problem was some unexpected events, which are not predictable in natural. The machine also needed maintenance such as changes of glue and etc. We did not have information about maintenance, we cannot distinguish it from the error, so there is no chance to predict when the glue should be changed – this is also dependent on product attributes, because every product is unique and needs the different amount of glue. Breaks are also not in database and therefore errors, maintenance and breaks are there together with no distinction.

Preparation task was the biggest issue. Situation in manufactory is different every day, some material could be available one day, another not. That is why the predictability is not an easy problem. But we still get some results - 30% error reduction (from 98s to 70s) is a significant improvement.

Another problem was the size of the data. We have good amount of them, but we have been able to use the whole data only for analysis of the machine task. We are waiting for the rest of them to try a better prediction of the preparation and insertion task. We assume that with more data, we will be able to create better methods to predict it or the same methods may return better results.

Maybe the biggest problem was to communicate it with managers. We did more experiments, for example cluster information was not available in the beginning - we tried to discover it from the data – similar products near together were put into the same cluster, but we discovered that this is not sufficient. It is essential for the analysis to get the right information and the right data. Before the real analysis, it is good to try to discover some obvious dependencies and if they are not here, some mistake occurred in the process. Expert information is also necessary. Workers know sometimes a lot and their experience must be in the data.

One of the hardest things was the trust over the data. When we were making experiments, we did not know if the data are adequately measured, if we got right information from the manager, if we did some mistake or if dependencies in data are different than it is supposed or if we are using the wrong methods. The comparison of obvious results to the workers made great help in controlling that we are on the good way.

ACKNOWLEDGMENT

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project „IT4Innovations excellence in science - LQ1602“.

REFERENCES

[1] Pospíšil, M., Mates, V., Hruška, T. 2013. Analysing Resource Performance and its Application in Company. In: The Fifth International Conference on Information, Process, and Knowledge Management, Nice, France, 24 February – 1 March: IARIA, 149-154.

[2] Pospíšil, M., Mates, V., Hruška, T., Bartík, V. 2013. Process Mining in a Manufacturing Company for Predictions and Planning, International Journal on Advances in Software, 2013(3): 283-297.

[3] Pospíšil, M., Mates, V., Hruška, T. 2013. Process Mining in Manufacturing Company. In: The Fifth International Conference on Information, Process, and Knowledge Management, Nice, France, 2013: IARIA, 143-148.

[4] van der Aalst, W.M.P., Reijers, H.A., Weijters, A.J.M.M. et al. 2007. Business process mining: An industrial application. Information Systems, 32(5): 713-732.

[5] Van der Aalst, W.M.P. 2010. Business Process Simulation Revisited. In Barjis, J., ed., Enterprise and Organizational Modeling and Simulation, Springer-Verlag, Berlin, 1-14.

[6] van der Aalst, W.M.P., Schonenberg, M.H., Song, M. 2011. Time prediction based on process mining, Information Systems, 36(2): 450-475.

[7] Rozinat, A., Mans, R.S. Song, M., et al. 2009. Discovering simulation models. Information Systems, 34(3): 305-327.

[8] Rozinat, A., Wynn, M.T., van der Aalst, W.M.P. et al. 2009. Workflow simulation for operational decision support, Data & Knowledge Engineering, 68(9): 834-850.

[9] Pospíšil, M., Hruška, T. 2012. Business Process Simulation for Predictions, In: BUSTECH 2012: The Second International Conference on Business Intelligence and Technology, Nice, France, 22-27 July: IARIA, 14-18.

[10] Grigori, D., Casati, F., Castellanos, M., et al. 2004. Business Process Intelligence, Computers. Industry – Process / Workflow Mining, 53(3): 321-343.

[11] Grigori, D., Casati, F., Dayal, U., Shan, M.C. 2001. Improving Business Process Quality through Exception Understanding, Prediction, and Prevention, In Proceedings of the 27th VLDB Conference, Roma, Italy, 11-14 September: Morgan Kaufmann, 159-168.

[12] Agrawal R., Srikant, R. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Chile, 12-15 September: Morgan Kaufmann, 487—499.

[13] Han, J., Pei, J., Yin, Y. 2000. Mining Frequent Patterns without Candidate, In Proceedings of the ACM-SIGMOD Conference on Management of Data, Dallas, USA, 16-18 May: ACM, 1-12.

[14] Nakatumba, J., Van der Aalst, W.M.P. 2009. Analyzing Resource Behavior Using Process Mining. In Business Process Management Workshops, Ulm, Germany, 7 September. Springer-Verlag, Berlin, 69-80.

[15] Song, M., van der Aalst, W.M.P., 2008. Towards comprehensive support for organizational mining. Decision Support Systems, 46(1): 300-317.

[16] van der Aalst, W.M.P., Weijters, A. J. M. M. 2004. Process mining: a research agenda, Computers in Industry, 53(3): 231-244.

[17] van der Aalst, W.M.P., van Dongen, B. F., Herbst, J., Maruster, L. et al. 2003. Workflow mining: A survey of issues and approaches, Data & Knowledge Engineering, 47(2): 237-267.

[18] van der Aalst, W.M.P. 2011. Process Mining, Berlin, Heidelberg: Springer Verlag.

[19] Wetzstein, B., Leitner, P., Rosenberg, F., et al. 2009. Monitoring and Analyzing Influential Factors of Business Process Performance. In Enterprise Distributed Object Computing Conference EDOC '09, Auckland, New Zealand, 1-4 September. IEEE, 141-150.