Process Mining Techniques in Conformance Testing of Inventory Processes: An Industrial Application

Zbigniew Paszkiewicz

Department of Information Technology, Poznań University of Economics,
al. Niepodległości 10, 61-875 Poznań, Poland
zpasz@kti.ue.poznan.pl

Abstract. Contemporary business information systems record business events in event logs. Process mining techniques permit to explore these logs to understand and analyze processes. Although, there are a number of case studies demonstrating usefulness of process mining techniques in real-life case studies, process conformance check techniques miss such practical evaluation. Also the commercial tools have not been tested for complementarity with the widely used academic tool ProM Framework in context of conformance check. In this paper an application of process conformance check techniques in a production company is described. Using a variety of techniques, an inventory processes supported by Warehouse Management System (WMS) have been analyzed. In particular, special attention has been put on *Product Management* process. The goal of this paper is to demonstrate the applicability of conformance check techniques to quality management in a production company. Both academic and commercial tools are used and evaluated in terms of efficiency and complementarity with each other.

Keywords: process mining, business process intelligence, inventory management, quality management, warehouse management system.

1 Introduction

Modern organization is interested in information about conformance of its processes to rules that should be observed. Those rules follow from law, industry standards or internal regulations. Frequently, conformance of processes to rules is tested by analysis of organizational processes. Any identified deviations from assumed rules are a potential source of legal, financial or operational risk.

Conformance testing is performed periodically in organizations. Typically used methods are based on financial document analysis, workshops and employee interviews. Such analysis needs long time and requires a lot of resources. As a consequence, in big organizations only partial information concerning organization processes is analyzed. Conclusions derived from partial information are then generalized using statistical methods. If the analysis of partial information uncovers no deviations, it is assumed that all the operations performed in the whole organization conform to rules. Great examples of such analysis are audits [1].

W. Abramowicz (Ed.): BIS 2013 Workshop, LNBIP 160, pp. 302–313, 2013.

In many cases this approach is insufficient. In global economy, turbulent organization environment strongly influences organization's operation requiring fast adaptation to new operation rules. Modern approaches to management require new methods of conformance analysis that are faster, cheaper, free of human judgment, and based on possibly complete set of information.

Proliferation of information technology is followed by increased number of processes that are performed by electronic means. The concept of *Process Aware Information Systems (PAISs)* has been proposed in [2]. A PAIS is *"a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models"* [2]. Examples of PAISs are: workflow management systems, Customer Relationship Management (CRM) systems, Warehouse Management Systems (WMS). Modern PAISs log enormous numbers of events providing detailed information about the activities that have been executed. Events are stored in repositories called *event logs*. Event logs are a basis for process mining. The term process mining is used to describe *"techniques, tools, and methods to discover, monitor and improve real processes by extracting knowledge from event logs commonly available in today's information systems"* [3].

Exploration of event log aims at discovering process models describing actually executed process instances and facts associated with those models [7, 8]. Among others, process mining techniques include process conformance testing. *Conformance check* techniques aim at automatic comparison of envisioned rules or models with behavior recorded in an event log to identify discrepancies [2, 9].

Two types of process models and two types of data are distinguished and used in process mining methods. A *de jure* process model is normative, i.e., it specifies how things should be done or handled. For example, a process model used to configure a BPM system is normative and forces people to work in a particular way [3]. A *de facto* model is descriptive. Its goal is not to steer or control reality. Instead, *de facto* models aim to capture reality.

Application of basic process mining techniques to real-life data and business problems is well documented [1, 4, 5]. The study concentrating on conformance check techniques still miss wide practical evaluation. For example, they raise questions concerning their effectiveness and usability. Moreover, no study concerning tool support for conformance check is available. Therefore, it is important to confront existing techniques and tools supporting conformance checking with real-life questions and event logs taken from real-life systems.

In this paper a case study based on a log of inventory processes is presented. Inventory processed were performed in a Polish company producing mattresses. The company employs in all the departments about five hundred people. The event log was generated by WMS supporting company operation which is not a workflow system. The event log used in the study described in this paper includes 781 134 events recorded from January 1st, 2013 to May 25th, 2013. The log contains information about 126 832 process instances. The analysis of the event log aimed at answering the questions raised by warehouse managers concerning organization operation. In this paper, only questions concerning conformance testing of the *Product Management* process are discussed. Open source ProM Framework [6] and commercial Fluxicon Disco (htttp://www.fluxicon.com/disco) process mining tools have been used to conduct the analysis. ProM Framework was selected because it provides the functionality allowing the wide spectrum of analysis. Commercial tools in genera offer very similar functionality and Disco is their great representative well-known from its effective implementation of algorithms.

The remainder of this paper is organized as follows. The basic facts concerning the case study are presented in Section 2. In particular business goals of the analysis and basic facts concerning the event log are outlined. In Section 3, the case study is described in detail. In Section 4, process mining tools are discussed. Finally, Section 5 concludes the paper.

2 Understanding Inventory Processes

2.1 Process Scope

There is number of processes performed in a warehouse including: *Material Receiving* process, *Product Shipping* process, *Product Management* process, and *Material Management* process. Due to a limited scope of this paper and for the sake of simplicity, only *Product Management* process is analyzed. Only a part of information from the event log is associated with this process.

The *Product Management* process encompasses activities required to take the product, i.e., mattress, from the production line and ship it to company's client. Products waiting for shipment are stored in the warehouse. Products are organized in pallets which are the smallest shipment and storage units. The products are categorized into *families*, which are understood as mattress types. There are twenty various product families. The transport of pallets among production lines, storage areas and shipment areas is done by storekeepers. Storekeepers work 24 hours per day on three shifts except weekends.

The process involves the following types of activities that are performed:

- *Production* – refers to a storekeeper activity of taking a pallet from production line;
- *Rest* – refers to a storekeeper activity of putting a pallet in the storage area;
- *Shipment approved* – refers to a storekeeper activity of preparing a pallet to shipping by putting it in a shipment area;
- *On fork* – refers to a storekeeper activity of transporting a pallet within the warehouse;
- *Shipped* – refers to an activity of actually shipping a pallet form a warehouse; obviously the shipping of an article is a process itself but WMS records it as a single activity and it is analyzed this way;
- *Deleted* – refers to an activity of removing a pallet from evidence.

Each activity performed by a storekeeper is recorded in the WMS. Before performing any activity, a storekeeper is obliged to scan a bar code available on every pallet. The WMS keeps track of a pallet life cycle and associates each scanning with appropriate activity. For instance, once a pallet is scanned after production (*Production* activity), the next recorded activity must be *On fork* activity and *Rest* activity. By choosing an option *"Start shipment"* in the WMS user panel, a storekeeper has a possibility to perform *Shipment approved* and *Shipped* activities. *Deleted* activity is performed only in exceptional situations.

There is one *de jure* model describing the life cycle of each pallet. In Fig. 1 *de jure* model is presented as a Petri net. The model assumes sequential execution of activities. When a pallet is transported within a warehouse multiple times the subsequence *On fork* > *Rest* can be performed multiple times.

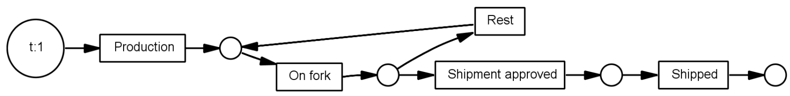

Fig. 1. Petri Net describing desired pallet life-cycle

2.2 Conformance Rules

The following aspects of conformance of the *Product Management* process have been identified by interviewing the *Product Management* process owner:

- *Conformance to model* – process instances must follow *de jure* model presented in Fig. 1;
- *First In – First Out policy* – products that were produced first must be shipped first; the FIFO rule must be satisfied for particular mattress family and size; to conform to FIFO rule, storekeepers must follow WMS recommendations concerning which pallets must be handled as the first; WMS does not recommend a particular pallet but a stand where the oldest pallet is located; the decision of which pallet to take from the stand is up to a storekeeper;
- *Quality assurance* – all the pallets shipped to a client must be checked by a quality department;
- *Process performance* – a particular pallet cannot be stored in a warehouse for more that fourteen days;
- *Pallet damage handling* – when a pallet is in disrepair, it must be transported to a special storage area; all the storekeeper are responsible for handling damaged pallets in this way;
- *Work distribution* – all the three shifts should perform equal amount of work; storekeepers are divided into two groups: (1) taking pallets from production lines, (2) shipping pallets from a warehouse; storekeepers from one group should not be involved in activities of the other.

2.3 Basic Facts

The WMS stores information concerning activities in a database. The following information is associated with each activity in the WMS event log: activity name, activity timestamp, name of a storekeeper executing the activity, identifier of a pallet being a subject of the activity, mattress family, warehouse name, optional stakeholder comment, optional pallet description. Information available only for the subset of activities include: storage area code (*Rest* activity), shipment area code (*Shipment approved*), recommended storage area code (*Rest*), information whether the recommendation was followed by a stakeholder (*Rest*). To validate all the rules defined in Section 3.2, the following attributes were derived from other data available in the WMS database: stakeholder shift, information if a pallet was damaged, information if a pallet was approved in terms of quality. The maximum number of attributes associated with a particular activity is sixteen. The pallet identifier is used to group activity instances into process instances.

Product Management process analysis is performed on basis of captured 554 745 events associated with 87 660 instances. This is a subset of information stored in the event log (*cf.* Section 1). 81% of process instances required less that 7 activities to be executed. The number of process instances performed for the most popular mattress family is 17 749. Execution of all the recorded *Product Management* process instances involved 55 persons.

3 Mining Process Aspects

In this section, conformance of process instances to rules defined in Section 3.2 is tested using process mining techniques.

3.1 Conformance to Model

In Fig 2, *de facto* model discovered from the event log generated by Fluxicon Disco is presented. Numbers assigned to activities and transitions indicate the number of process instances they appeared in the log.

De facto model presented in Fig 2 indicates that the execution of the process is far more complex than assumed in the *de jure* model (*cf.* Fig. 1). In particular, many additional transitions among various activities appear which are not included in the *de jure* model, e.g. transition from *Shipped* activity to *Rest* activity, self-loops for activities *On fork*, *Production*, *Shipment approved*, *Shipped* and *Rest*. *De facto* model indicates that not only *Shipped* activity, but also *Rest*, On *fork* and *Deleted* activities are closing the process. Finishing *Rest* and *On fork* activities correspond to process instances that are still running.

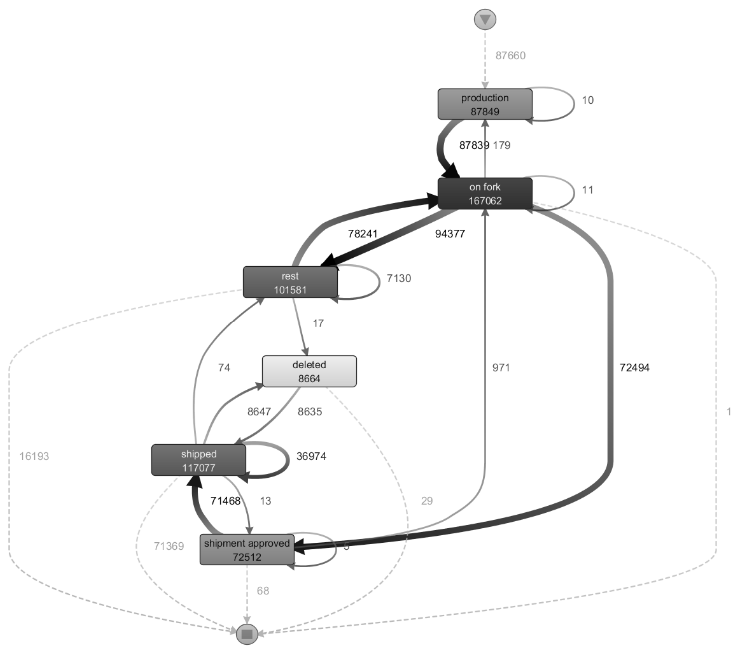

Fig. 2. Discovered model of the Product Management process

The most frequently executed activities and transitions marked with thick arrows and dark rectangles fit to the *de facto* model almost perfectly. Thus, in general, the process conforms to the model but some deviations must be investigated more closely. The result of automatic comparison of the event log behavior with the *de jure* model is presented in Fig 3. The validation was performed using *Conformance Checker* plugin [10] of the ProM Framework. The validation was performed for 1068 process instances executed in May 2013.

The quality of the *de facto* model is measured by fitness measure. *Fitness measure* estimates to which extent a model allows for the behaviour perceived in the event log. In this approach the *de jure* model is represented by the Petri net (*cf.* Fig. 1). In particular, fitness measure relates the number of missing tokens with the number of consumed ones and the number of remaining tokens with the number of produced ones [10]. If the model fully supports the behaviour seen in the event log, i.e., no tokens are missing nor remaining, the fitness is evaluated to 1. In the worst case, every produced and consumed token is remaining or missing, the metric evaluates to 0.

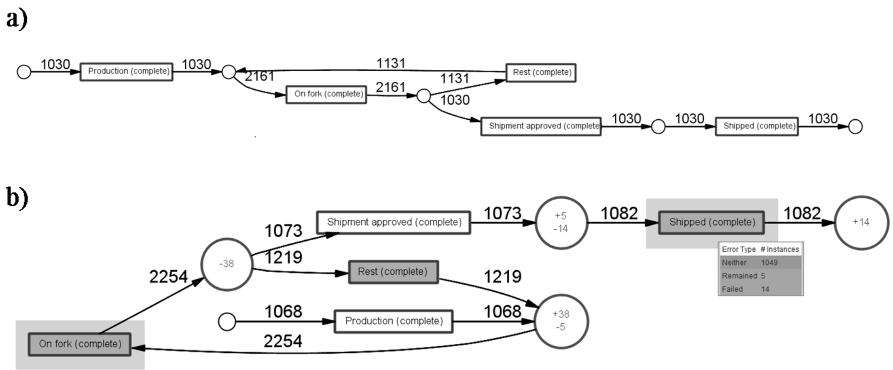

Fig. 3. Automatic conformance checker

The results of the analysis of the subset of 1030 process instances is presented in Fig 3a. Execution of these 1030 process instances is fully supported by the *de jure* model, so the fitness value is estimated to 1. Fig 3b includes the conformance testing of the full set of 1068 process instances performed in May. Fitness is evaluated to 0,9926. Visualization of the process model in Fig 3b provides some additional information:

- *Token Counter* (circles with numbers) – indicates parts in the model where the mismatch took place by visualization of the missing and remaining tokens;
- *Failed Tasks* (dark rectangles) – transitions that were not enabled and therefore could not be successfully executed;
- *Remaining Tasks* (gray bigger rectangles) – transitions that remained enabled, which indicates non-proper completion of the process; marked activities should have been executed;
- *Path Coverage* (thick borders) – transitions that were executed;
- *Passed Edges* (black numbers) – Indicates for each transition how often it was followed by various process instances.

The high value of the fitness measure indicates good conformance of a model with the event log behaviour. Problems appear mainly before the execution of *Shipped*, *Rest* and *On fork* activities. This observations are confirmed by complementary analysis performed in Fluxicon Disco. The analysis reveals ten variants of process execution. Each variant refers to aifferent sequence of activities performed in the process. Depending on the variant, the average duration of a process instances varies from 1 day 19 hours to 17 days 1 hour.

The most frequently executed variant fits the *de jure* model: *Production* > *On fork* > *Rest* > *On fork* > *Shipment approved* > *Shipped*. In May 2013, this variant was executed 946 times and accounts for 88,58% of all the process instances. Three other variants include multiple executions of *On fork* > *Rest* subsequence. Those variants also correspond to the *de jure* model. The variants not conforming to the *de jure* model include: execution of many *Rest* activities in a row, double execution of the *Shipped* activity at the end of the process, and execution of *On fork* activity after *Shipment approved* activity. Wrong variants were identifed in 38 process instances that account for 3,55% of all process instances perfomed in May. Those observations are in line with the results of analysis performed in *Conformance Check* plugin.

3.2 First In-First Out

The *Dotted Chart plugin* available in the ProM Framework is used for FIFO rule conformance testing. Generated charts are presented in Fig 4. Each row in the dotted chart corresponds to exactly one process instance and each dot corresponds to an activity instance. Process instances are sorted from the top according to start time. The color of the dot is associated with activity type: light grey color corresponds to *Production* activity while dark grey color corresponds to *Shipped* activity. Other activities were excluded from this analysis.

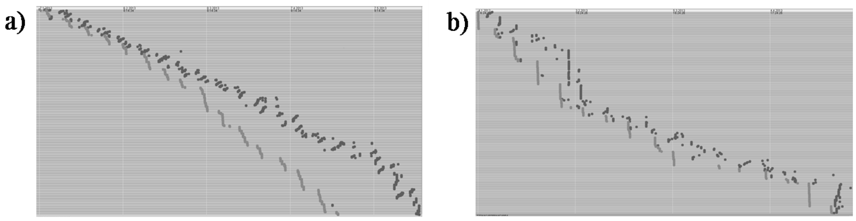

Fig. 4. FIFO analysis using Dotted Charts in ProM for two different mattresses

In Fig 4, the company efforts for FIFO assurance are clearly visible. Some deviations from the rule are visible, as some pallets are shipped significantly later than others. Conformance to the FIFO rule varies from product to product. This is visible when comparing diagram in Fig 4a and Fig 4b.

The non-conformance of storekeepers' behavior to recommendations generated by the WMS has been recorded for 5231 activities performed within 3665 process instances (4% of all the process instances). Only *Rest* and *Production* activities are affected by missing adherence to recommendations. In case of 644 process instances not following the recommendation is justified, as those instances involve damaged pallets. Such pallets must be transported to a separate storage area. The number of process instances affected by not recommended activities is not big but influences conformance to the FIFO rule.

It is worth to note that process mining helps in noticing some deviations or trends. Evaluation concerning positive or negative impact of identified problems on the organization must be done taking into account particular organizational context. For instance, the level of conformance to FIFO is strongly influenced by organization of stands in the warehouse. Current organization of stands forces early produced product pallets to be placed deeper on the stand. Access to such products requires removal of the later produced pallets which usually is not performed. Also the WMS provides recommendations concerning stands not pallets. This fact is visible especially in Fig 4a, where dark grey dots create small lines with direction opposite to light grey lines. Thus, it is worth to note that the perfect conformance of warehouse operation to FIFO will never be achieved by the company.

3.3 Quality Assurance

Using filtering based on activity attributes (*cf.* Section 3.3) it is possible to extract those process instances from the event log that were not accepted by the Quality Department. Exactly 12 such process instances were identified.

In Fig 5, the model describing the execution of such process instances is presented. All the 12 pallets were not shipped to a client. Ten of them were destroyed by performing *Deleted* activity. The two process instances were still running when the event log was created (*Rest* activity). This confirms high conformance to the quality assurance rule.

In Fig 5, performance characteristic of the process is also presented. The numbers correspond to the average time of transition execution. Typically, process instances are short and transitions are performed quite fast. Only transition from *Rest* to *Deleted* activity takes longer time. This is due to one process instance, where this transition took 2 days and 12 hours raising the average.

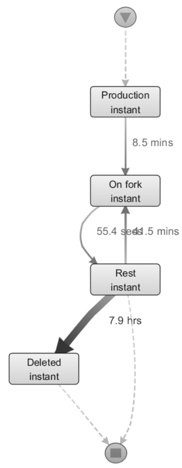

Fig. 5. Process instances missing quality acceptance

3.4 Process Performance

In evaluation of process performance, only process instances that finished with the *Shipped* activity are taken into consideration. The 14-day rule is satisfied for 15 800 of finished process instances. They account for 22% of all the process instances in the event log. 8 562 (11.99%) process instances were performed within 8 days, while 55 569 finished process instances exceeded the desired time.

Although, the rules are rarely satisfied, such behavior is justified in a particular context. During the analyzed time period, the company was creating inventory for an intensive selling period that always takes place in the second half of the year. Early produced pallets were waiting in the warehouse to be shipped in the second half of the year. This fact is confirmed by an increasing number of active process instances growing over time.

3.5 Pallet Damage Handling and Work Distribution

The issue of pallet damage handling appeared in 676 process instances. This number is a number of process instances in which the performed *Rest* activity is associated with a storage area designated for damaged mattress storage. The number of affected process instances is equally distributed among various product families. The number of damages for the three most frequently damaged mattresses was: 127, 126 and 115.

Not all the storekeepers are equally involved in placing the damaged pallets in required storage areas. There are two persons that together deal with 24% of process instances involving damaged pallet. Then the group of 24 employees is responsible for the remaining 76% of process instances.

Observation about an unequal distribution of work among storekeepers can be generalized to all the *Product Management* process instances. It is possible to distinguish 4 persons performing more than 30% of activities within the *Product Management* process. It is worth to note that the *Product Management* process is one of many processes performed in the warehouse. Thus, the work distribution should be analyzed concerning all the processes performed in the warehouse. This is, however, out of the scope of this paper.

Analyzing the table presented in Fig 6, it can be also easily noticed that storekeepers are divided into two separate groups. In Fig 6, rows correspond to different storekeepers while columns correspond to activity types. Numbers contained in the table indicate the number of times a particular type of activity was executed by a particular person. Storekeepers that perform *Production* activities are usually not involved in *Shipped* activities.

On fork	Production	Rest	Shipment approved	Shipped
0	0	0	0	78
0	0	0	0	31
0	0	0	0	1
0	0	0	0	186
0	0	0	0	14
0	0	0	0	202
0	0	0	0	8
0	0	7	0	0
0	0	12	0	0
44	0	5	39	0
159	17	58	101	1
207	202	207	0	0
99	19	34	65	10
113	101	113	0	0
90	10	13	77	124
173	0	1	172	0
0	0	0	0	1
70	70	70	0	1
96	106	96	0	0
0	0	0	0	3
138	1	6	132	180
227	222	227	0	0
0	0	0	0	13
130	11	12	118	0
84	30	36	48	2
0	0	0	0	3
23	0	0	23	80
147	146	147	0	0
50	2	25	25	16
247	0	0	247	124
130	131	130	0	0
0	0	12	0	0
27	0	1	26	4
0	0	7	0	0

Fig. 6. Originator by task matric for Product management process instances executed in May

Finally, differences in the number of activities performed on different shifts are significant. While the first shift performs 37,54% of activities, the third shift handles only the 29,12% of activities.

4 Tools

The set of activities with corresponding attributes was extracted from a database using standard SQL statements. Both Java and Python languages were used to clean the data and extract additional information.

Two tools for process mining were used in the analysis: Fluxicon Disco (version 1.3.6) and ProM Framework (version ProM 5). Disco is a commercial software developed by the Fluxicon company. The academic license allows analysis of event logs up to 1 million events. The process map generator (*cf.* Fig. 3), build-in filtering mechanism, import of CSV files and export to MXML and XES formats compatible with ProM application has been used in the analysis. Disco's built-in filtering algorithm allows including or excluding process instances and activities based on the appearance of one or more attributes. Although Disco does not provide support for process conformance checking, the filtering algorithms can be used to partially fill this gap. This is especially useful when testing various hypotheses. In this way process discovery techniques can be used for conformance testing.

ProM Framework complements the functionality of Disco. ProM has been used to perform conformance tests, generate dotted charts and calculate originator to task matrix. ProM provides refined functionality of conformance testing but general usability of this software is low, e.g., finding basic facts concerning a particular process requires a lot of effort. The most important drawback of the ProM Framework is low efficiency of algorithms composing it which prevents its users from analysis of big volumes of data.

5 Conclusions

Current methods of conformance testing based on process mining are applicable for real-life data. The presented analysis demonstrated business value coming from such analysis. The analysis was performed on data coming from the WMS as it is, without any modifications of the system. The analysis required cooperation with the process owner and occasional support from company's IT department during data preparation. No other resources were involved.

Among others, the analysis performed using process mining techniques lead to the following action undertaken in a company: training were scheduled for storekeepers not following de jure model, the company received a precise information on level of conformance to a FIFO policy for various families of products and was able to schedule action aiming to eliminate discrepancies. Many conformance problems are not always a consequence of storekeepers' behavior or wrong work organizations. Frequently, the wrong configuration of WMS is an issue, e.g., activities saved twice in the database. In such case, process mining methods contribute to information system testing.

The main conclusion is that conformance testing questions requiring the analysis of mutual impact of process instances and social relation structures could not be answered. For instance, does the presence of the two particular storekeepers on the same shift contribute to the increase of damaged pallets number? Currently, no automatic analysis concerning mutual impact of process execution and social relations on one another can be performed. Moreover, the existing conformance testing methods support only structured and repeatable processes like the ones analyzed in this paper. In future works, the methods contributing to conformance testing of unstructured process are planned to be developed. The developed methods must support conformance testing of both processes and social relations. The analysis of mutual impact of process execution on social relations and vice versa should demonstrate great practical value. A step towards such analysis was made in [11] where service protocols were proposed as an approach to modeling social relations as a part of process model was proposed. The next step would be to develop techniques automatically discovering and validating service protocols.

Pure conformance testing functionality is implemented only in the ProM Framework. ProM Framework requires very specialized knowledge concerning processes modeling and interpretation of generated results. ProM has low efficiency of analysis of big amounts of data. Events log with more then 700 000 events can hardly be imported into application. This prevents conformance testing based on data coming from wider time periods. For example, in Section 4.1 only one month data have been analyzed. To answer real-life conformance-related questions, techniques traditionally classified to *process conformance checking* group (*cf.* Section 2) are not sufficient. In practice, the use of those methods must be backed up by *process discovery* methods.

Acknowledgments. This work has been partially supported by the Polish National Science Center. Grant no. DEC-2011/01/N/ST6/04205.

References

1. Aalst, W.M.P., Hee, K.M., Werf, J.M., Verdonk, M.: Auditing 2.0: Using Process Mining to Support Tomorrow's Auditor. Journal Computer 43(3), 90–93 (2010)
2. Dumas, M., Aalst, W.M.P., Hofstede, A.H.: Process Aware Information Systems: Bridging People and Software Through Process Technology. John Wiley&Sons, Hoboken (2005)
3. IEEE Task Force on Process Mining: Process Mining Manifesto (2012), `http://www.win.tue.nl/ieeetfpm/doku.php?id=shared:process_mining_manifesto` (last access: July 18, 2013)
4. Mans, R.S., Schonenberg, M.H., Song, M., Aalst, W.M.P., Bakker, P.J.M.: Application of Process Mining in Healthcare – A Case Study in a Dutch Hospital. In: Fred, A., Filipe, J., Gamboa, H. (eds.) Biomedical Engineering Systems and Technologies, pp. 425–438 (2008)
5. Caron, F., Vanthienen, J., Weerdt, J., Baesens, B.: Beyond X-Raying a Case-Flow: Adopting Different Focuses on Care-Flow Mining (2011), `http://www.win.tue.nl/bpi2011/doku.php?id=challenge` (last access: July 18, 2013)
6. Aalst, W.M.P., Dongen, B.F., Günther, C., Rozinat, A., Verbeek, H.M.W., Weijters, A.J.M.M.: ProM: The Process Mining Toolkit. In: BPM 2009 Demonstration Track, Ulm, Germany (2009)
7. Weijters, A.J.M.M., Aalst, W.M.P.: Process Mining: Discovering Workflow Models from Event-Based Data. In: 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001), pp. 283–290. BNVKI, Maastricht (2001)
8. Aalst, W.M.P., Rubin, V., Dongen, B.F., Kindler, E., Günther, C.W.: Process Mining: A Two-Step Approach to Balance Between Underfitting and Overfitting. Software and Systems Modeling 9(1), 87–111 (2009)
9. Rozinat, A., Aalst, W.M.P.: Conformance Checking of Processes Based on Monitoring Real Behavior. Information Systems 33(1), 64–95 (2008)
10. Conformance Checker, `http://www.processmining.org/online/conformance_checker` (last access: July 18, 2013)
11. Picard, W.: A Formalization of Social Requirements for Human Interactions with Service Protocols. In: Information Sciences, pp. 1–21. Elsevier (2013)