Analyzing PACS Usage Patterns by Means of Process Mining: Steps Toward a More Detailed Workflow Analysis in Radiology

Daniel Forsberg^{1,2} · Beverly Rosipko² · Jeffrey L. Sunshine²

Published online: 9 September 2015
© Society for Imaging Informatics in Medicine 2015

Abstract In this paper, statistical analysis and techniques from process mining are employed to analyze interaction patterns originating from radiologists reading medical images in a picture archiving and communication system (PACS). Event logs from 1 week of data, corresponding to 567 cases of single-view chest radiographs read by 14 radiologists, were analyzed. Statistical analysis showed that the numbers of commands and command types used by the radiologists per case only have a slightly positive correlation with the time to read a case (0.31 and 0.55, respectively). Further, one way ANOVA showed that the factors time of day, radiologist and specialty were significant for the number of commands per case, whereas radiologist was also significant for the number of command types, but with no significance of any of the factors on time to read. Applying process mining to the event logs of all users showed that a seemingly “simple” examination (single-view chest radiographs) can be associated with a highly complex interaction process. However, repeating the process discovery on each individual radiologist revealed that the initially discovered complex interaction process consists of one group of radiologists with individually well-structured interaction processes and a second smaller group of users with progressively more complex usage patterns. Future research will focus on metrics to describe derived interaction processes in order to investigate if one set of interaction patterns can be considered as more efficient than another set when reading radiological images in a PACS.

Keywords PACS · Workflow · Chest radiographs · Data mining · Efficiency · Productivity

Introduction

The introduction of new technology (e.g., new modalities, picture archiving and communication system (PACS), RIS, and other IT tools) has introduced large changes in the radiological workflow, and much research has been devoted to study how these technology changes affect the workflow and efficiencies [1–4]. However, most of this research has focused on a very high-level view of the radiological workflow, i.e., on a macroscopic level where workflow steps are related, e.g., to patient arrival, start and end of imaging procedure, start of image interpretation, and report sign-off [5, 6]. Very few have provided a more detailed level of investigation to the level of individual interaction steps of a radiologist reading an examination in a PACS, where an interaction is defined as a command issued by a user via an input device such as a mouse or a keyboard. In the few examples of more detailed analysis in the literature, focus has either been limited to analyzing frequency of commands used in a PACS [7–9] or on creating abstract representations of the process associated with reading an examination in a PACS [10]. However, analysis of actual specific recorded PACS interactions to discover underlying processes and workflow patterns remains absent.

An important difference between broader high-level workflow related to the pathway of a patient or an examination in radiology and the more granular low-level PACS usage workflow is that the high-level workflow, to a certain degree, is defined by external constraints. For example, an examination has to be performed and completed before a radiologist can start his/her interpretation. On the other hand, the workflow of a radiologist, when reading an examination in a

✉ Daniel Forsberg
daniel.forsberg@sectra.com

¹ Sectra, Teknikringen 20, SE-583 30 Linköping, Sweden

² Department of Radiology, Case Western Reserve University and University Hospitals Cleveland, 11100 Euclid Avenue, Cleveland 44106, OH, USA

PACS, seldom has any explicit constraints on how the radiologist can interact with the PACS. For example, there are no constraints forcing the radiologist to change window-level before he/she can zoom or pan an image.

We have the opportunity to study the detail on the workflow of radiologists to investigate the potential of using this specific knowledge to improve the representation and, perhaps from that, the efficiency of radiologists' usage of PACS. Potential benefits are based on the ability to better understand and quantify differences in PACS usage, something which would allow the characterization of certain workflows or interaction patterns as more efficient than others in relation to different contexts (e.g., different examination types). Hence, it may become relevant to discuss best practices in terms of PACS usage when reading radiological examinations, something which already has been indicated in [11] but which no one, to the best of our knowledge, thus far has explored in greater detail.

One approach for analyzing workflows and processes is found in process mining [12]. In general, there exists three types of process mining: process discovery (deriving workflow models from event logs), conformance checking (analyzing the difference between a modeled workflow and a set of event logs), and process enhancement (improving current workflows). Although process mining has been applied in a number of different contexts including health care [13–15], it has primarily been applied to discover and understand business processes. Recent research suggests the applicability of process mining for analyzing user behavior (i.e., click streams) in a web context [16, 17] and for understanding user interaction in a message-based software client–server framework [18].

We hypothesize that process mining is applicable for analyzing underlying workflows related to interaction patterns of PACS usage.

The purposes of this paper then are to demonstrate an approach for analyzing interaction patterns obtained from radiologists reading examinations in a PACS and to consider this as a possible route for further radiological workflow efficiency improvements. In this work, we employ standard statistical tests to analyze interaction patterns from a group of radiologists to determine the significance of various factors in regard to PACS usage. Further, by applying process discovery followed by complexity analysis and conformance checking on recorded interaction patterns, differences in discovered process models amongst a group of radiologists will be categorized.

Material and Methods

This study was considered as exempt from review by our institutional review board as part of a larger quality project in our department. The following sub-sections provide

information about the included material (event logs tracking interaction patterns of PACS users) and the methods employed to analyze this material (i.e., process mining and related software).

Event Logs

Event logs form the starting point for any work within process mining. An event log consists of a number of traces, each describing a specific case with an arbitrary number of ordered events or activities. Additional information can also be recorded for each event, e.g., timestamps, resources involved in the event, and time to complete the event. Prior to process mining, event logs need to be derived or extracted from relevant IT systems. Table 1 provides an example of what an event log could look like in a ticket refund request setting.

The PACS (Sectra PACS, version 15.2) installed at University Hospitals in Cleveland provides the possibility to log all interactions performed, i.e., all commands issued by each PACS user. Control occurs at the level of logging of which users, applications, and/or workstations to log commands for. The log files themselves are stored as standard text files, although with a specific formatting, and contain information such as which command, who issued the command, when, and on which workstation. In the current study, command logs from December 6 to December 12 2014 were extracted for further processing, i.e., a week of data running from Saturday to Friday.

Before analysis, the logs were filtered to remove irrelevant information and converted to a format recognized by process mining software, the XES format [19]. The pre-processing consisted of concatenating the daily logs to a single file, splitting user sessions into separate traces (based upon the command “SetCurrentExam”), and removing commands irrelevant to the workflow of reading examinations in the PACS. Further, only traces related to radiologists, single chest radiographs, and including the command “Dictate Report” were retained for further processing. All traces had an artificial end activity added (“ArtificialEnd”), which basically obtained its timestamp from the next SetCurrentExam command in the session. As such, each trace will contain at least three commands SetCurrentExam, Dictate Report, and ArtificialEnd. The event logs were finally saved as xml files according to the XES format, with each event containing information about the command issued, when, and by whom.

Table 2 lists some basic statistics of the unprocessed event logs. Note that almost all users have read cases where only the minimum three commands SetCurrentExam, Dictate Report, and ArtificialEnd are used. Table 3 shows information about when cases were read. Regarding the users, it can be noted that all users were board certified for diagnostic radiology but that users 4–6 and user 9 were ER radiologists. Statistical analysis

Table 1 Example of an event log describing a process related to handling requests for refunds on tickets

Order number (case id)	Activity (activity name)	Time stamp	User (resource)
1101	Register request	1/1/2015@09.15	Rob Smith
1102	Register request	1/1/2015@09.25	Rob Smith
1103	Register request	1/1/2015@09.27	Rob Smith
1101	Check ticket	1/1/2015@10.01	Sue Jones
1101	Examine causally	1/1/2015@10.11	Fred Scott
1101	Decide	1/1/2015@10.15	Fred Scott
1103	Examine thoroughly	1/1/2015@10.20	Sue Jones
1101	Pay compensation	1/1/2015@10.40	Al Green
1102	Examine causally	1/1/2015@10.56	Fred Scott
1102	Check ticket	1/1/2015@10.56	Sue Jones
1102	Decide	1/1/2015@11.01	Ann Anderson
1101	Reject request	1/1/2015@11.46	Al Green
...	...	...	...

of the interaction patterns was performed using Python 3.4.1 and SciPy 0.15.1. A p value of <0.05 was set to define significance.

Process Mining

The following paragraphs will discuss some of the specifics of the process mining methods applied in this work. The interested reader is referred to [12] for a more complete overview of process mining or to [20] for an introduction to process mining within health care.

We have three main types of process mining: process discovery, conformance checking, and process enhancement. Process discovery refers to the type of process mining where a discovery technique is applied to an event log and produces as output an abstract model describing the underlying process. Essentially, the discovered process model describes what activities are involved, the order of these activities, and whether

there exists splits and joins in the process. The splits and joins can be of the type AND, OR, or XOR. An AND split after an activity signifies a workflow where all directly following activities are executed, an OR split where any combinations of the directly following activities are executed, and an XOR split where only one directly following activity is executed. It is the AND and OR splits that can cause concurrent execution paths in a workflow. The process model is typically visualized using a graph model. For example, applying process discovery to the event log of a workflow for requesting refunds on tickets exemplified in Table 1 could result in a process model as shown in Fig. 1.

In this study, the heuristics miner has been selected as process discovery algorithm [21]. The heuristics miner was chosen because of its ability to handle noise and to discover the main behavior of an analyzed event log. The output from the heuristics miner includes a process model showing activities and their internal structure as

Table 2 Basic statistics of the event logs included for analysis, for all, and for each individual users with more than 20 read cases

Users	Number of cases	Commands per case mean [min, max]	Command types per case mean [min, max]	Median time to read an examination (s)
All	567	10 [3, 69]	5 [3, 17]	71.5
User 1	41	3 [3, 13]	3 [3, 8]	75.0
User 2	98	12 [3, 65]	7 [3, 16]	85.5
User 3	22	5 [3, 10]	4 [3, 6]	61.0
User 4	27	15 [3, 38]	5 [3, 9]	68.0
User 5	118	10 [4, 28]	4 [4, 8]	73.5
User 6	96	16 [3, 69]	6 [3, 17]	93.0
User 7	44	4 [3, 14]	3 [3, 10]	150.0
User 8	44	7 [3, 19]	5 [3, 8]	90.5
User 9	32	4 [3, 20]	3 [3, 10]	70.5

Table 3 Number of read cases during weekday or weekend daytime/nighttime

Users	Weekday daytime	Weekday nighttime	Weekend daytime	Weekend nighttime
All user	381	147	16	23
User 1	41	0	0	0
User 2	98	0	0	0
User 3	22	0	0	0
User 4	1	10	9	7
User 5	44	72	0	2
User 6	38	56	0	2
User 7	44	0	0	0
User 8	44	0	0	0
User 9	5	9	6	12

Daytime is defined as between 6 a.m. and 10 p.m., i.e., nighttime as 10 p.m. to 6 a.m., and with the weekend running from Fri 10 p.m. to Mon 6 a.m.

described by a graph, but it also includes information such as how often an activity has been performed, how often one activity has been performed before or after another, and the strength of the connection between two activities (given as a number between 0 and 1). As such, it describes the most likely set of possible behaviors.

Discovered process models can be described using a number of different graphical representations, e.g., Petri nets, causal nets, or event-driven process chains. A Petri net is the most common representation used in later steps of process mining (conformance and enhancement). Therefore, derived process models using other representations are typically converted to a Petri net prior to further analysis.

One way to analyze a discovered process model as represented by a Petri net is to compute the complexity of the model, where a higher complexity refers to a

workflow with more splits and joins and thus supports a greater variety of possible behaviors or interaction patterns as in our case. A number of different measures exist for this purpose, and in this work, we have employed three measures (chosen based upon their availability within the employed process mining software). Two of the measures are given by simply computing the sum of all splits and joins in a net and by computing the ratio of the number of arcs (connecting transitions and places in a Petri net) over the number of places (signifying a state) where, for both metrics, a larger number indicate a more complex workflow. For example, in Fig. 1, there is a total of nine splits and joins and 19 arcs/seven places=2.71 arcs/places. The third metric [22] computes the complexity of a workflow as the number states that can be reached by all splits. For example, assuming a split with n outgoing arcs, then an AND split adds a single state, a XOR

Fig. 1 Example of a process model for the workflow of requesting refunds to tickets represented as a Petri net process model. The *squares* in the graph denote transitions representing activities, and the *circles* denote places representing states in the workflow

split adds n states, and an OR split adds $2^n - 1$ states. All three complexity measures were included, since existing literature thus far only has suggested them but offered no further explanation regarding their differences in terms of assessing complexity. Note that the measured complexity will only refer to the complexity of the interaction patterns as described by the discovered process model; it does not measure the complexity of the complete workflow related to reading an examination as pertaining to the cognitive process of reading an examination and interacting with other IT systems apart from the PACS.

Following process discovery in process mining, conformance may be evaluated. In conformance checking, a set of traces are replayed on a process model in order to assess if deviations exist between the traces and the model. In this work, a conformance metric based upon optimal alignments of event logs and models has been chosen [23] to compute the trace fitness, ranging between $[0, 1]$ and where 1 equals a perfect fit.

Process enhancement refers to the last step of process mining and describes a set of practices that can be employed to add actual value to current processes. The most obvious option is to include timestamp data and to conduct a performance analysis to identify bottlenecks in a process and thereby finding potential sub-processes that can be subject to process improvements. Another option includes operational support where ongoing cases in a process are analyzed in order to detect anomalies, predict outcomes of a case, or to recommend future courses of action. Note that process enhancement is not employed in this study but is intended to be included in subsequent studies.

Process Mining Using ProM

The event logs were analyzed using ProM (<http://www.promtools.org>). ProM is an open source software for process mining and provides an extensible platform implemented in Java with support for many of the process mining techniques presented in recent literature. In this work, two different versions of ProM were employed, versions 5.2 and 6.4.1, which was necessary because they differ in terms of available functionality.

The event logs were analyzed according to the following scheme, first on the event logs with all users and then on the

event logs from each individual radiologist with more than 20 read examinations:

1. Import event logs into ProM 5.2.
2. Perform process discovery using the “Heuristics miner” with default settings.
3. Convert the discovered heuristics net to a Petri net.
4. Analyze the complexity of the Petri net using the “Petri Net Complexity Analysis” plug-in.
5. Export the Petri net as a pnml file.
6. Import the Petri net and event logs into ProM 6.4.1.
7. Perform conformance checking (trace fitness) using the plug-in “Replay a Log on Petri Net.” Trace fitness was assessed using default parameters apart from the following:

- No final marking.
- Silent transitions were set to be invisible.
- Penalize improper completion = true

Results

Statistical Analysis of Interaction Patterns

Statistical analysis was applied to the unprocessed individual traces/interaction patterns previously summarized in Tables 2 and 3. The Pearson correlation coefficient was computed to assess the linear relationship between the number of commands per case, the number of command types per case, and time to read each case. The computed correlations were 0.78 ($p < 0.001$) between number of commands and number of command types, 0.31 ($p < 0.001$) between number of commands and time to read, and 0.55 ($p < 0.001$) between number of command types and time to read. During this analysis, only interaction patterns with a time to read of less than 10 min were included. This limitation was employed to avoid including traces related to users taking a break or being interrupted during their interpretation of a traditional chest radiograph examination.

One-way ANOVA was employed to determine which factors of time of day, radiologist, and specialty that were related with significant differences for the number

Table 4 Effect of time of day, radiologist, or specialty on number of commands, number of command types, and time to read

Factor/Response variable	Number of commands	Number of command types	Time to read
Time of day	<0.001	0.39	0.55
Radiologist	<0.001	<0.001	0.07
Specialty	<0.001	0.72	0.98

P values from one-way ANOVA <0.05 show statistically likely effects

of commands per case, the number of command types per case, and the time to read a case. The results are given in Table 4 and show that all factors were

significant for the number of commands, whereas only radiologist was significant for the number of command types per case and none of the factors for the time to

Fig. 2 Detected heuristics net when mining the event logs of all users. The discovered process model has a spaghetti-like structure with a lot of connections between many different activities; i.e., a single well-ordered workflow is difficult to discern. Note that the visualized heuristics net

contains integer values describing how often a certain command/activity has been performed along with how often a path between two activities has been travelled and decimal values describing how strong a connection between two activities is (value between 0 and 1)

read a case. Interaction patterns were included if time to read was less than 10 min, and the reading radiologist had read at least 20 cases.

Discovered Heuristics Nets

This and the following sub-sections will highlight some of the results that were obtained during the process discovery and subsequent analysis.

Fig. 3 Detected heuristics net from user 3. Note the much simpler interaction process, i.e., “SetCurrentExam” → “Dictate Report” → “Next Hanging” → “Previous Hanging” → “ArtificialEnd.” User 3 was the user with the least complex interaction process

In the first step of process discovery, the event logs corresponding to all users were analyzed and the

Fig. 4 Detected heuristics net from user 1. In this case, the detected process model is slightly more complex than for user 3; however, the detected process model shows a greater fitness with its corresponding event logs (see Table 5)

connections. What can be noted though is the relatively high frequency of some of the commands, “WindowLevel,” “Pan,” “Invert,” “Next Hanging,” “Previous Hanging,” “DragDisplayUnit,” and “HangDisplayUnits,” the two latter related to changing a hanging protocol.

The diagram illustrates a complex state transition system, likely representing a medical image analysis workflow. Each state is represented by a box containing its name, a status (e.g., "complete"), and a numerical value. Transitions between states are labeled with a probability and a count.

Key States and Transitions:

- SetCurrentExam (complete) 96** transitions to **Dictate Report (complete) 100** with probability 0.989 and count 96.
- Dictate Report (complete) 100** transitions to **WindowLevel (complete) 889** with probability 0.925 and count 95.
- WindowLevel (complete) 889** is a central hub with multiple outgoing transitions:
 - To **Matrix Window (complete) 19** (0.812, 15)
 - To **DragDisplayUnit (complete) 46** (0.944, 18)
 - To **HangDisplayUnits (complete) 44** (0.923, 11)
 - To **Zoom In (complete) 8** (0.95, 22)
 - To **Pan (complete) 45** (0.5, 4)
 - To **Distance (complete) 11** (0.545, 6)
 - To **ROI Circle (complete) 1** (0.9, 7)
 - To **DeleteOverlay (complete) 6** (0.25, 2)
 - To **Zoom Out (complete) 7** (0.5, 7)
 - To **Next Hanging (complete) 7** (0.667, 3)
 - To **Previous Hanging (complete) 6** (0.667, 3)
 - To **Grayscale Presets (complete) 11** (0.667, 7)
 - To **ArtificialEnd (complete) 96** (0.01, 79)
- Matrix Window (complete) 19** transitions to **DragDisplayUnit (complete) 46** with probability 0.944 and count 18.
- DragDisplayUnit (complete) 46** has a self-loop (0.957, 2) and transitions to **HangDisplayUnits (complete) 44** (0.589, 43).
- HangDisplayUnits (complete) 44** has a self-loop (0.957, 1) and transitions to **1 x 2 (complete) 1** (0.5, 1), **Information Window (complete) 19** (0.667, 6), **Retrieve from Archive (complete) 4** (0.667, 2), and **Pan (complete) 45** (0.9, 7).
- Pan (complete) 45** has a self-loop (0.941, 20) and transitions to **Distance (complete) 11** (0.667, 4).
- Distance (complete) 11** transitions to **ROI Circle (complete) 1** (0.8, 4) and **DeleteOverlay (complete) 6** (0.5, 1).
- ROI Circle (complete) 1** transitions to **DeleteOverlay (complete) 6** (0.5, 1) and **Arrow (complete) 14** (0.5, 1).
- DeleteOverlay (complete) 6** transitions to **Arrow (complete) 14** (0.5, 1) and **Zoom Out (complete) 7** (0.5, 2).
- Arrow (complete) 14** transitions to **Zoom Out (complete) 7** (0.5, 2).
- Zoom In (complete) 8** transitions to **Zoom Out (complete) 7** (0.25, 2).
- Next Status (complete) 92** transitions to **Chat Window (complete) 2** (0.667, 2) and **ArtificialEnd (complete) 96** (0.988, 46).
- Chat Window (complete) 2** transitions to **ArtificialEnd (complete) 96** (0.667, 2).
- Invert (complete) 98** has a self-loop (0.974, 19) and transitions to **ArtificialEnd (complete) 96** (0.5, 44).
- Next Hanging (complete) 7** has a self-loop (0.923, 4).
- Previous Hanging (complete) 6** has a self-loop (0.923, 4).
- Grayscale Presets (complete) 11** has a self-loop (0.667, 6).

Fig. 5 Detected heuristics net from user 6. This is the most complex process model of all mined models from the individual users

Fig. 6 Each detected heuristics net was converted to a Petri net. In this case, compare the derived Petri net with the detected heuristics net in Fig. 3 from user 3 and note the one-to-one correspondence between the

two models. The *black squares* refer to so-called silent transitions that are needed to ensure that a place in the Petri net is not directly linked to another place

observed. For Figs. 3 and 4, the most notable difference from that of all users in Fig. 2 is the small number of commands used with corresponding much simpler process model. In contrast, the discovered interaction process in Fig. 5 is similar to the interaction process of all users, showing a very complex interaction pattern.

Heuristics Nets Converted to Petri Nets

Figures 6 and 7 provide two examples of heuristics nets converted to a Petri net. Note that the conversion from a heuristics net to a Petri net mainly is a matter of representation; i.e., the underlying process model is not affected.

Petri Net Complexity Analysis and Conformance Checking

Results from the complexity analysis and the conformance checking of each Petri net and associated event logs are given in Table 5. It can be noted that as visually observed for the discovered process models, the process model for all users and the process models corresponding to some of the users, users 2, 4, and 6, show a higher degree of complexity than the others, as given by all three complexity measures. A strong correlation exist between the measures, 0.99 between sum of joins/splits and arcs per places, 0.88 between control-flow and sum of joins/splits and 0.85 between control-flow and arcs per places. This suggests that either sum of joins/splits or arcs per

Fig. 7 The discovered heuristics net of user 6 converted to a Petri net

places to be redundant. Further, the trace fitness shows no positive correlation with the different complexity measures but, instead, a somewhat negative correlation, indicating that a higher degree of complexity is not due to a better fit between the event log and the discovered process model.

Discussion

The results from the statistical analysis of the interaction patterns revealed a number of interesting findings. The obvious correlation (0.78) between the number of commands and the number of command types per case is expected. More interesting was the low correlation between these two variables and the time to read an examination. This demonstrates that an interaction process with lot of commands or command types does not have to equal a longer time to read an examination or that a process with less commands will result in a faster read. A specific example of this is given by user 7, who stands out for the long median time to read an examination (150 s, Table 3), whereas the mean number of commands (four as in Table 2) and command types (three as in Table 2) in no way stand out compared with the other users. However, important to note here is that, in this analysis, only PACS usage has been recorded, and hence, a case requiring a long time to read without extensive PACS interaction does not imply inactivity from the user who might be occupied simply viewing the image or interacting with other IT systems.

The p value from the one-way ANOVA showed that all three factors (time of day, radiologist, and specialty) are significant for the number of commands per case, whereas only radiologist was shown to be significant for the number of command types per case. No significance of the three factors on time to read was found. However, given the observational design of this study, the potential causalities implied by these

findings are of limited validity. Additionally, it is important to note the unequal sample distribution between the factors and corresponding factor levels; e.g., only four radiologists read during nighttime and two of them handled the majority of these cases. Hence, the discovered significance of the factor time of day might as well be related to the significance of the factors radiologist or specialty.

Following the statistical analysis on the unprocessed interaction patterns, techniques from process mining were applied to those patterns for process discovery. At first, all interaction patterns of all radiologists were analyzed. This analysis revealed little in terms of a consistent process of PACS usage employed when reading single-view chest radiographs, since the derived process model had a typical “spaghetti-like” pattern. The possibility of this outcome was high, as initially indicated, since there exist no (or very few) external constraints on PACS use limiting the order of interactions for a radiologist when reading. More interesting were the results from analyzing the interaction processes of each radiologist. As exemplified in Figs. 3 and 4, some of the radiologists read their examinations using a much more simple process, whereas others, as given in Fig. 5, use a more complex process. Hence, the initially discovered complex interaction process associated with all users can be understood as consisting of one group of users with more simple and well-structured interaction processes and another group of users with a progressively more complex behavior.

The visual interpretation of the derived heuristics nets is also confirmed by the complexity analysis performed on the corresponding Petri nets, as given in Table 5. Here, it can be noted that the users 2, 4, and 6 stand out as having more complex interaction processes than the other users, if considering all three complexity measures (sum of joins/splits, arcs per place, and control flow). Note that the same group of users (i.e., users 2, 4, and 6) already stand out in Table 2, if one

Table 5 Results from the complexity analysis and conformance checking of the Petri nets corresponding to the discovered interaction processes

Users	Sum of joins/splits	Arcs per place	Control-Flow	Trace fitness
All	68	4.60	848	0.78
User 1	6	3.09	44	0.99
User 2	42	3.78	207	0.74
User 3	4	2.86	34	0.91
User 4	27	3.48	142	0.90
User 5	13	3.31	41	0.89
User 6	45	4.04	944	0.90
User 7	11	3.20	145	0.95
User 8	19	3.31	59	0.81
User 9	13	3.25	65	0.97

Higher numbers represent a higher degree of complexity in the first three columns. In the last column, trace fitness, a better fit between an event log and a process model is given by a higher number and where 1.00 equals a perfect fit

considers the mean and max values for number of commands and number of command types. Hence, it could be postulated that the more commands and command types that a radiologist uses, the more complex the discovered interaction process. Further, the complexity of the derived interaction processes has no positive correlation with the trace fitness from the conformance checking that analyzes how well the discovered process models match the behavior described in the event logs. Since an increased trace fitness between an event log and a discovered process model typically is associated with an increased complexity in the process model, this implies that the increased complexity of an individual interaction process is not because that specific interaction process might have a better trace fitness but rather because its underlying interaction pattern is more complex.

The present analysis has a number of limitations. For example, the timestamps associated with the ArtificialEnd events do not necessarily correspond to the actual time when a radiologist completed the process of reading an examination, potentially causing the recorded time to read to be somewhat longer than in reality. The event logs contain no record of whether a radiologist has been interrupted or not during a reading session, which we have attempted to mitigate by excluding study readings longer than 10 min. Another limitation is our lack of confirmation that the included interaction patterns only refer to readings of examinations that have not previously been viewed. Some of the included interaction patterns might represent a radiologist that previously viewed the examination but did not start to dictate at that time and, instead, returned later to the same examination to perform the dictation through an apparently streamlined process. We should also note that the analyzed event logs contain very little information of the actual examination, so that it is possible that the interaction processes of the three users 2, 4, and 6 are more complex simply because the examinations read by them were more complex. This, on the other hand, would suggest that the other users only read cases that are easy to interpret, which is an unlikely scenario given our distribution of daily work. Finally, the ability to generalize the findings described in this work is limited given that the data included originates from one institution, spanning a single week and only for a single examination type.

Conclusions and Future Work

In this paper, we have shown how to perform an analysis of interaction patterns related to the radiologists' use of PACS while reading medical images. This method holds potential for achieving a better understanding of the workflow processes used to read medical images and for creating new areas of potential workflow efficiencies. The analysis consisted of two parts, one where the interaction patterns themselves were statistically analyzed and one where process models were

derived and analyzed using process mining. Statistical analysis of the interaction patterns surprisingly showed that the number of commands and command types per case only have a slightly positive correlation with the time to read a case. The factors time of day, radiologist, and specialty were shown to be significant for the number of commands per case and that radiologist is also significant for the number of command types. Further, process mining showed that most users have a more simple interaction pattern while reading single-view chest radiographs whereas a smaller group of users has progressively more complex interaction patterns.

Future work includes extending the current analysis to include more detailed studies of how factors such as examination type, time to read, and time of read affect the complexity of the derived interaction processes. With the aid of process mining, we may extend the performance analysis of the interaction patterns to not only include the transit time from start to finish of an interaction process but also to include performance analysis of the sub structures to search for potential bottlenecks. Including a more detailed performance analysis would provide the opportunity to investigate if it is possible to identify certain interaction processes as more efficient than others. Another compelling future research direction would be to include information extracted from the request and report texts and to use this information as way of categorizing the interaction patterns before process mining.

Funding DF is supported by a grant (2014–01422) from the Swedish Innovation Agency (VINNOVA).

References

1. Reiner B, Siegel E, Carrino JA: Workflow optimization: current trends and future directions. *J Digit Imaging* 15:141–152, 2002
2. Srinivasan M, Liederman E, Baluyot N, Jacoby R: Saving time, improving satisfaction: the impact of a digital radiology system on physician workflow and system efficiency. *J Healthc Inform Manage* 20:123–131, 2006
3. Nitrosi A, Borasi G, Nicoli F, Modigliani G, Botti A, Bertolini M, Notari P: A filmless radiology department in a full digital regional hospital: quantitative evaluation of the increased quality and efficiency. *J Digit Imaging* 20:140–148, 2007
4. Lahiri A, Seidmann A: Analyzing the differential impact of radiology information systems across radiology modalities. *J Am Coll Radiol* 6:705–714, 2009
5. Redfern RO, Horii SC, Feingold E, Kundel HL: Radiology workflow and patient volume: effect of picture archiving and communication systems on technologists and radiologists. *J Digit Imaging* 13:97–100, 2000
6. Mariani C, Tronchi A, Oncini L, Pirani O, Murri R: Analysis of the X-ray work flow in two diagnostic imaging departments with and without a RIS/PACS system. *J Digit Imaging* 19:18–28, 2006
7. Horii SC, Feingold ER, Kundel HL, Nodine CF, Langlotz CP, Redfern RO, Grevera G, Brikman I, Muck J: PACS Workstation functions: usage differences between radiologists and MICU

- physicians. In *Medical Imaging*, 266–271. International Society for Optics and Photonics, 1996
8. Erickson BJ, Ryan WJ, Gehring DG, Beebe C: Clinician usage patterns of a desktop radiology information display application. *J Digit Imaging* 11:137–141, 1998
9. Erickson BJ, Ryan WJ, Gehring DG: Functional requirements of a desktop clinical image display application. *J Digit Imaging* 14:149–152, 2001
10. Noumeir R: Radiology interpretation process modeling. *J Biomed Inform* 39:103–114, 2006
11. Reiner B: Automating radiologist workflow, part 2: hands-free navigation. *J Am Coll Radiol* 5:1137–1141, 2008
12. van Der Aalst WM: *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media, 2008
13. Rebuge Á, Ferreira DR: Business process analysis in healthcare environments: a methodology based on process mining. *Inf Syst* 37:99–116, 2007
14. Bouarfa L, Dankelman J: Workflow mining and outlier detection from clinical activity logs. *J Biomed Inform* 45:1185–1190, 2012
15. Mans RS, van der Aalst WM, Vanwersch RJ, Moleman AJ: Process mining in healthcare: Data challenges when answering frequently posed questions. In *Process Support and Knowledge Representation in Health Care*, 140–153. Springer Berlin Heidelberg, 2013
16. Maruster L, Faber NR, Jorna RJ, van Haren RJ: A process mining approach to analyse user behaviour. In *WEBIST* 208–214, 2008
17. Poggi N, Muthusamy V, Carrera D, Khalaf R: Business process mining from e-commerce web logs. In *Business Process Management*, 65–80. Springer Berlin Heidelberg, 2013
18. Rubin VA, Mitsyuk AA, Lomazova IA, van der Aalst WM: Process mining can be applied to software tool. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, 2014
19. Günther CW, Verbeek, HMW: *XES-standard definition*, 2014
20. Mans RS, van der Aalst WM, Vanwersch RJ: *Process Mining in Healthcare: Evaluating and Exploiting Operational Healthcare Processes*, 2015
21. Weijters AJMM, van der Aalst WM, De Medeiros AA: Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven. Tech. Rep WP 166:1–34, 2006
22. Cardoso J: Control-flow complexity measurement of processes and Weyuker's properties. In *6th International Enformatika Conference*, 8:213–218, 2005
23. van der Aalst WM, Adriansyah A, van Dongen B: Replaying history on process models for conformance checking and performance analysis. *Wiley Interdiscip Rev Data Min Knowl Disc* 2:182–192, 2012