Monitoring Deployed Application Usage with Process Mining

C.W. Günther, A. Rozinat, and W.M.P. van der Aalst
Group of Information Systems
Eindhoven University of Technology, The Netherlands
{c.w.gunther,a.rozinat,w.m.p.v.d.aalst}@tue.nl

Kenny van Uden
Cardio/Vascular Development
Philips Healthcare, The Netherlands
kenny.van.uden@philips.com

Abstract

Increasingly, devices are connected to the Internet and are recording events. Computers, X-ray machines, high-end copiers, interactive kiosks, RFID scanners, etc. are examples of such devices. Typically, there are many devices running the same application (e.g., a computer running SAP R/3 or an X-ray machine running diagnosis software) and their event logs reflect the actual use of the application in the field. This way it is possible to systematically monitor real-life processes. Process mining aims at the analysis of such event logs, and we argue that proper log analysis can be used to increase reliability, usability, and serviceability of deployed applications. In this paper, we present a case study conducted within Philips Healthcare and based on the process mining tool ProM. Moreover, we address technical challenges of dealing with large, real-life data sets, and discuss generic methods of dealing with problems of diversity and complexity in unstructured environments.

1 Introduction

More and more information about processes is recorded by information systems in the form of so-called "event logs". A wide variety of *Process-Aware Information Systems* (PAISs) [10] is recording excellent data on actual events taking place. ERP (Enterprise Resource Planning), WFM (WorkFlow Management), CRM (Customer Relationship Management), SCM (Supply Chain Management), and PDM (Product Data Management) systems are examples of such systems. Moreover, also *Embedded Systems* (ESs) have started to record events. Examples are high-end copiers, medical imaging devices, RFID scanners, domotics systems, etc. Both PAISs and ESs have in common that a particular *application* is often running at many different locations. For example, Philips has thousands of X-rays machines deployed in hospitals all over the world and SAP has more than 120,000 installations of their ERP worldwide. The above applications have in common that they record events and that, in principle, these events can be transmitted over the Internet. This paper aims at an intelligent analysis of these events by applying *process mining to such deployed applications*.

Process mining techniques attempt to extract non-trivial and useful information about some real-world *process* from event logs. One aspect of process mining is control-flow discovery, i.e., automatically constructing a process model describing the causal dependencies between activities observed in reality. However, process mining is not limited to the control-flow perspective nor the discovery of models. Also other perspectives such as data, resources, time, etc. can be analyzed. Moreover, if there is already a model, then one may not want to discover a new model but measure the conformance of the existing model or extend the model with additional information (highlight bottlenecks, etc.). Process mining addresses the problem that most businesses have very limited information about what is actually happening in their organization. Similarly, vendors of PAISs and ESs are not aware of the actual usage of their deployed applications. In practice, there is often a significant gap between what is prescribed or supposed to happen with deployed applications, and what *actually* happens. Fortunately, a concise assessment of the actual use based on event logs, can assist in improving the situation.

In this paper, we focus on *deployed applications*, i.e., some application is used at different sites (e.g., a medical device of a particular type that is used within hundreds of hospitals or a CRM system that is used by thousands of organizations) and emits events that are gathered at some central point. At each site events are collected, so, conceptually one can think of the gathered data as a collection of event logs each describing the use of the application within a particular context. In such a setting process mining can support the following tasks.

- *Usage profiling.* By using process mining one can build models describing the actual use of the application. Such models may reveal that certain features are never used or that they are only used by particular customers under particular circumstances. This helps to characterize the different types of users and may influence further development, training, and marketing.

- *Reliability improvement.* The models discovered through process mining show the actual use of the application and also its failures. Insight into the actual use of the application can be used to test the system under realistic circumstances (sometimes this is even enforced by law). Moreover, a detailed analysis of failures using process mining can help to find root causes for reliability problems.

- *Usability improvement.* If the intended use deviates from the actual use, then this may point towards usability problems. Using conformance checking one can locate such deviations and quantify their impact.

- *Remote diagnostics and servicing.* Information in event logs can also be used more actively. For example, process mining can help to predict failures. By combining current data and historic information, it is possible to forecast likely problems. Moreover, if an error occurs, the event log may be used to find the core problem and take counter measures.

This paper presents a case study where we have applied our process mining techniques to the *remote servicing network of Philips Healthcare* (formerly known as Philips Medical Systems). Using a custom implementation based on the ProM framework (our *Pro*cess *M*ining tool), we have analyzed the event logs of more than a thousand X-ray machines of Philips distributed all over the world. These systems record events at a fine-grained level and Philips is using this information to further improve the reliability, usability, and serviceability of their systems. We report on our experiences in this case study and show that there are many technical challenges (e.g., terabytes of data and many possible views). However, if one overcomes these challenges, then the potential impact is breathtaking and of great economic value.

The remainder of this paper is structured as follows. Section 2 describes the distributed application usage logging by Philips Healthcare's X-ray machines. Section 3 provides an overview of process mining and the ProM framework. The technical challenges encountered in this case study are described in Section 4. Section 5 shows how the diversity and complexity of such event logs can be addressed. Section 6 describes the concrete setting in which process mining has been applied in this case study so far, and Section 7 concludes the paper.

2 Distributed Application Usage Logging

Philips started its medical activities in 1918, when it first introduced a medical X-ray tube. In 1933 Philips started to manufacture other medical equipment in Europe and the United States. Today, *Philips Healthcare* (PH) is a global leader in diagnostic imaging systems, healthcare information technology solutions, and patient monitoring and cardiac devices. One type of diagnostic imaging systems manufactured by PH are cardiovascular imaging systems, which are part of the business line CardioVascular (CV). The business line CV makes products for two main clinical areas, *Cardio* and *Vascular*. Cardio refers to all medical issues concerning the heart, and Vascular refers to all medical issues concerning blood vessels. These clinical areas are targeted by the *Allura Xper* productline. The Allura Xper productline consists of X-ray systems designed to diagnose and assist in the treatment of, for example, heart and lung diseases, by generating images of the internal body. An example Allura Xper system is depicted in Figure 1.

Figure 1. Allura Xper system.

The *Philips Remote Services* (PRS) is a system for the active monitoring of all systems connected and for the supply of services via the intranet. Philips' guaranteed, secure broadband connection delivers remote technical support, monitoring, diagnostics, application assistance and many added value services. Throughout the whole world, currently more than 1400 Allura Xper systems are connected to the PRS. Via the PRS network, event logs are downloaded to PH. Using the *Remote Analysis, Diagnostics And Reporting* (RADAR) system, the event logs are converted to an XML format and stored in an internal database of RADAR. RADAR monitors PH systems and gathers service data via the PRS network.

Analyzing these event logs in a structured way can deliver new insights about the actual usage of the Allura Xper systems in the hospitals. These insights are relevant in various phases of the development process. For example, in *testing* it helps to concentrate the test efforts on the most relevant (e.g., most used) functions of the product. However, also other areas, such as product *development* and *support*, could benefit from this knowledge.

3 Process Mining and ProM

Process mining is a field of analysis techniques that focus on mining behavioral aspects from data. Since the mid-nineties several groups have been concentrating on the discovery of process models (i.e., structures that model behavior) from event-based data [4, 5, 6, 7, 8]. In [3] an overview is given of the early work in this domain. The idea to apply process mining in the context of workflow management systems was introduced in [5]. In parallel, Datta [7] looked at the discovery of business process models. Cook et al. investigated similar issues in the context of software engineering processes [6]. Herbst [13] was one of the first to tackle more complicated processes, e.g., processes containing duplicate tasks. Despite the many relations to some work discussed in the Machine learning domain, there are also many differences as the targeted process models reside at the net level (e.g., Petri nets) rather than sequential or lower level representations (e.g., Markov chains, finite state machines, or regular expressions). Therefore process mining needs to deal with various forms of concurrency.

Over the years the notion of process mining has broadened to the general idea of discovering, monitoring and improving *real* processes (i.e., not assumed processes) by extracting knowledge from event logs. Assuming that we are able to log events, a wide range of process mining techniques comes into reach: we can use process mining to (1) *discover* new models (e.g., constructing a Petri net that is able to reproduce the observed behavior), (2) check the *conformance* of a model by checking whether the modeled behavior matches the observed behavior, and (3) *extend* an existing model by projecting information extracted from the logs onto some initial model (e.g., show bottlenecks in a process model by analyzing the event log).

Traditionally, process mining applications have dealt with event logs stemming from relatively rigid, and thus structured processes, such as an administrative process that is enacted on a workflow system [2]. However, today it can be observed that more and more events are being logged in situations where the information system is not directly noticeable, where processes are more flexible, and thus of an *unstructured*, *ad-hoc* nature. Examples are mobile phones, car entertainment systems, production systems (e.g., wafer steppers), copiers, and the Allura Xper X-ray machines of PH analyzed in this paper. As far as we know, process mining techniques have not been used for analyzing deployed applications. Applying process mining to these unstructured processes poses a number of challenges due to their *diversity* and *complexity*, as we will explain in Section 5.

ProM [1] is a toolset that supports a wide variety of process mining techniques[1]. Furthermore, due to its plug-able architecture, it enables researchers to quickly implement new algorithms without spending any efforts on common tasks such as loading logs, displaying the graphs of discovered models etc. To account for the special needs of the domain while leveraging existing algorithms and techniques, a custom plug-in was developed for PH in the context of ProM. Since ProM uses the *M*ining *XML* (MXML) format to read event logs, the logs collected by the RADAR system (cf. Section 2) needed to be converted into this format. The basic structure of MXML is very simple: a process log consists of a set of process instances, which in turn each contain a sequence of events. A process instance, also referred to as case, trace, or audit trail, is one particular realization of the process, while events correspond to concrete steps that are undertaken in the context of that process for the particular process instance. Furthermore, each event carries a timestamp and many contain additional data.

Note that already *during log conversion different views on the process can be taken*, since the definition of what is to be considered a process instance determines the scope of the process to be analyzed. For example, in the context of the usage process of the Allura Xper system one could consider the usage of the system over the course of a whole day, as well as at the treatment session of a single patient. Furthermore, the logs contain references to certain domain-specific usage contexts, such as the so-called *procedures* and *applications*. Since these procedures and applications provide pre-configured parameters for specific, well-defined application scenarios, it is interesting to analyze the usage of the machines within such a particular context. For example, one might want to focus on commands that were triggered in the context of a procedure that is optimized for brain imaging. Thus, we leverage different conversions to obtain different views on the process.

4 Technical Challenges

The event log data recorded by Alura Xper systems, and aggregated in the PRS / RADAR infrastructure, have some characteristics, which make them difficult to analyze. Events are recorded on a very *low level*, representing single *commands* of the system, such as moving the examination table or capturing a single image. Logs also contain events referring to boundaries of *higher-level abstractions* in the system, such as *procedures* (e.g., analyzing a specific blood vessel) and *applications* (e.g., a general type of analysis, such as vascular analysis). There are many choices to consider when converting such ambiguous logs, which renders this step very involved, and makes it crucial for the quality of later analysis.

Another difficulty is the sheer amount of data available for analysis, which is the result of a combination of many log-recording systems (currently more than 1400), and of

[1] The software including source code and documentation can be freely downloaded from *prom.sf.net* (see also our website *processmining.org*).

the fine-grained nature of logging, producing large data sets. Loading a representative set of logs, which may contain from *hundreds of thousands up to millions of events*, consumes considerable time even using contemporary hardware. To illustrate this problem, when one wants to analyze 200 X-ray systems, each recording around 200.00 events per day, over 100 days, one needs to analyze 400 million events. This excludes any kind of interactive analysis, since most mining algorithms scale with the size of the log.

In order to alleviate these performance problems, it has been decided to convert the input data on a number of levels of abstraction, each producing a separate log file. The PH-specific plugin in ProM provides a *dashboard* surface, which allows the user to navigate the set of logs available, and to select an appropriate subset for actual mining. On the top level, X-ray systems (i.e., specific installations) are identified, and can be selected by e.g. their country of installation. From there, the user can drill down to selecting specific patient treatments, applications and procedures used, down to the sequence of invoked commands. Since each level of abstraction is represented by a separate set of log files, and logs are compiled from a large set of available cases interactively, ProM had to be extended by a mechanism for supporting log aggregation. This mechanism prevents that unnecessary data is loaded, and improves access performance to log data considerably.

Due to the architecture of ProM, log objects need to be duplicated very frequently, for tasks such as filtering and analysis. Since the PH plugin makes excessive use of large logs and filtering, these duplications were having a serious performance impact on the usability of the system. Addressing these problems, a number of systematic performance improvements were implemented. In the following, we provide two representative examples for a number of changes which had to be implemented throughout the ProM framework, and in the ProM*import* framework used for log conversion [11].

- *Lazy Duplication*. Duplication of logs in ProM occurs when the receiving party may modify the log, in order to protect the original. This behavior has been changed to create *soft copies* of cases, which initially point to the original case. Only when actual modification occurs will the respective case be copied. Since modifications are relatively infrequent, this change significantly improved log handling performance.

- *Modification Window*. In order to free memory for analysis tasks, logs are stored as binary streams on disk by ProM. When events within such a stream are modified, the whole stream needs to be copied in case the modified event has grown in size. The storage of events in streams has been changed to initially include some spare space after each event, so that modified events often still fit into their initial window within the storage stream. This change prevents a large number of expensive file copy transactions, and has led to significant improvements in log filtering speed throughout the framework.

From a process mining perspective, there are two important lessons to be learned from our experience: First, event log data from a real domain comes in sizes that makes it necessary to provide sophisticated means for structuring it, and to select an appropriate subset. And second, even a reduced subset of data may have up to millions of events in one log. This can pose significant performance problems when analyzing the underlying process.

5 Managing Diversity and Complexity

The use of a deployed application like the Allura Xper system can be characterized as an *ad-hoc process*. While short episodes of using the system may be well-structured (due to e.g. implementation of user interfaces, medical best practices, or common sense), the process of a doctor using an X-ray appliance is mainly *unstructured*. Further, the systems analyzed in this study provide a large set of functionality, which is recorded on a very low level of abstraction.

As such, this domain perfectly exemplifies two major challenges of analyzing unstructured processes. *Diversity* describes the problem, that several cases (i.e., usage sessions of the system) may be structured entirely differently, thus making any comparison hard. *Complexity* refers to the high level of detail and richness in potential behavior, which contradicts the analyst's desire to extract high-level information and general trends from the data.

The following subsections address the problems of diversity and complexity in more detail, and describe the methods which were applied to counter these effects.

5.1 Trace Clustering

If logs from a number of different processes are analyzed as if they had been derived from a single process, the result is typically difficult to interpret because the merging of processes blurs the underlying process structures. The typical problems observed in resulting process models are an overwhelming number of task nodes, and equally large numbers of relations, resulting in the so-called "spaghetti models". These problems are also observed, when an event log is very *diverse*, i.e. single cases differ significantly from one another. It is, in such cases, a valid assumption that there are a number of *tacit process variants* hidden within one event log.

The problem with diversity, i.e. that subsets of cases are structured very differently, obviously grows with the number of cases being analyzed. Thus, one solution to this problem is to reduce the number of cases which are analyzed at once. Tacit processes will usually not be explicitly known, i.e. there is no available knowledge on how to partition the set of cases. One can, however, measure the similarity of cases and use this information to divide the set of cases into more homogeneous subsets.

Trace clustering implements this *divide-and-conquer* approach in a systematic manner. There are a number of *profiles*, each measuring a number of *features* for each case from a specific perspective. For example, one profile measures the number of occurrences for each event class per case. Based on these *feature matrices*, a number of *distance metrics* can be applied to compute the relative distance between any two cases in the log. Finally, any standard data clustering algorithm can be applied, grouping closely related cases into subsets. These subsets can subsequently be analyzed independently from one another.

We have implemented *trace clustering* with a diverse set of profiles and distance metrics in the ProM framework. While this implementation could be used successfully to reduce diversity in the input data and to improve mining results, we have used domain knowledge in this study to derive a customized implementation. The analyst can select a level of abstraction for feature selection, e.g. one can cluster cases based on common procedures. Based on this simplified feature set, the Agglomerative Hierarchical Clustering (AHC) algorithm is used, which successively combines clusters into larger ones [9].

The benefit of using this approach for this study is, that the clustering procedure is based on concepts the user of the mining tool (who is intimately familiar with the domain) understands intuitively. Usage sessions of X-ray machines are mainly characterized by which functionality is used, and this is captured precisely in the set of procedures and applications, which the user may choose from. The AHC algorithm further allows the user of the mining tool to fine-tune the size of clusters, by adjusting the similarity threshold. This ability to navigate the diversity of a large set of cases with a precise, yet user-friendly (since simple and domain-aware) method is important for efficient mining analysis and meaningful results.

5.2 Fuzzy Mining

Another challenge for process mining, which is orthogonal to diversity, is *complexity*. We can observe complexity as an *overwhelming number of artifacts* (i.e., events in a log, or nodes in a process model), combined with a large number of ordering relations (i.e., edges in a process model). While diversity can always be attributed to a lack of kinship between cases, complexity can be observed even within one single case.

Diversity can be explained by an *inappropriate starting point* for process mining, i.e. by analyzing an overly diverse set of cases at once, one is essentially comparing apples to oranges. Regarding complexity, a complex process model can be considered a correct and faithful representation of a complex process. While being correct, however, the process model is *inappropriate* for the goal of analysis, since it cannot be understood efficiently.

Process mining techniques which are suitable for complex environments need to be able to provide a high-level view on the process, abstracting from undesired details. The field of cartography has always been faced with a quite similar challenge, namely to simplify highly complex and unstructured topologies. Activities in a process can be related to locations in a topology (e.g. towns or road crossings) and precedence relations to traffic connections between them (e.g., railways or motorways). Two concepts from cartography which are successfully used to simplify complex topologies are *aggregation*, i.e. hiding details in opaque clusters (cf. the cities on a road map), and *abstraction*, i.e. removing less-interesting artifacts (cf. not showing dirt-roads on a map).

The *fuzzy mining* approach has been developed to deliver such simplified process models of complex environments, using the concepts of aggregation and abstraction [12]. Two metrics have been defined to guide the simplification of process models. *Significance* describes the relative importance of (i.e., the relative interest one has in) an event, or a precedence relation between two events. *Correlation*, on the other hand, measures how closely related two events following one another are. The fuzzy mining algorithm *preserves* highly significant behavior. Behavior which is less significant, but highly correlated, is *aggregated*, i.e. hidden in cluster nodes. Finally, behavior which is both less-significant and lowly-correlated is *abstracted from*, i.e. removed from the model.

Figure 2 shows the implementation of the fuzzy mining approach in the ProM framework. A small excerpt of a larger *fuzzy model* is shown, highlighting some concepts used for visualization. Rectangular nodes represent preserved activities, while green octagonal nodes represent clusters of aggregated activities. Edges between nodes, representing preserved precedence relations, are colored by correlation (i.e., the darker the edge, the more correlated the endpoints) and have a width corresponding to their significance. One important feature of the fuzzy mining approach is that it allows for *adaptation and exploration*. The analyst can interactively modify parameters controlling the aggregation and abstraction algorithms, thereby exploring different *perspectives* on the process. This is a crucial distinction from earlier approaches, which implicitly assume the existence of *one perfect model*. Interactive exploration makes efficient use of the user's domain knowledge to yield a truly appropriate representation of the process.

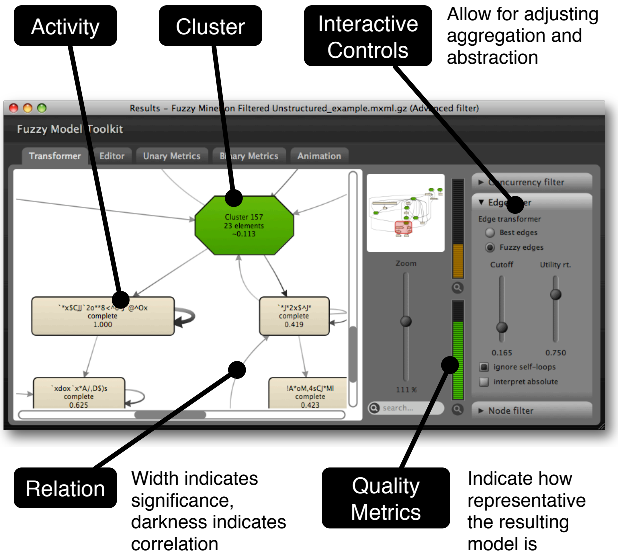

Figure 2. Fuzzy mining in ProM.

During this study, the fuzzy mining approach has proven to allow analysts, who are familiar with the application domain, to create suitable and insightful representations of usage processes. We have further extended and adapted the approach for this specific domain, taking into account additional knowledge about the input data. While the fuzzy mining approach allows the analyst to *impose an adjustable hierarchy* on the otherwise "flat" process model, domain knowledge has been used to impose and also display an *explicit hierarchy*. The analyst can navigate the process in a top-down fashion, starting from identifying the procedures performed in a clinical examination, down to the content of those procedures and atomic events. Applications and procedures are depicted as blue octagonal cluster nodes, ensuring that the user is aware of hierarchy. Another extension uses the knowledge about explicit hierarchy to ensure that clusters only contain nodes, which are part of the same concept on the above hierarchy level (e.g., only commands from the same procedure are clustered). This technique is also used to assign meaningful names to cluster nodes, i.e. the name of the common concept. Since one goal of the study was tracking specific cases, their path can also be highlighted within the process model, further adding to its understandability.

6 Application

For PH the main goals of applying process mining techniques in this case study so far have been (i) to gain insight into actual application usage in general, and (ii) the extraction of typical usage patterns as input for automatic reliability growth testing. Insight into actual application usage yields knowledge that is relevant in various phases of the product development process, and reliability is one of the key quality characteristics that must be tested before a medical device can be released on the market. The System Integration & Test department (SI&T) of PH and the process mining group at Eindhoven University of Technology collaborated to address both goals [14] by developing a PH-specific plug-in in the context of ProM:

(i) The PH plug-in allows for interactive discovery of process models on different levels of abstraction. As described earlier, existing techniques like trace clustering and fuzzy mining have been leveraged but also enhanced by domain-specific knowledge to create results that are as appropriate and understandable as possible for the domain experts. The top-most view on the usage process is a so-called *user profile*, which characterizes the session of a patient treatment. It is started by a preparation phase and finished by a completion phase. In between, commands are executed in the context of different domain-specific procedures.

(ii) One technique that is used for reliability testing within PH is the so-called "real-life test". The real-life tests are executed by automatic test scripts on real X-ray machines and the objective of these real-life tests is to discover defects that can occur in the final clinical use (i.e. in the hospital or clinic). Therefore, the test scripts should reflect user behavior as accurately as possible. The PH plug-in supports the export of selected user profiles to use actually observed usage sequences extracted from log data to achieve this.

In the following, we summarize the overall functionality of the PH plug-in, describe an example analysis session, and report on the status of adoption within the organization.

Filtering Simply due to the sheer mass of data—but also because application usage may differ, e.g., per region—it is necessary to be able to restrict the analysis to a subset of the monitored systems. The PH plug-in supports the filtering according to a variety of relevant, domain-specific characteristics, such as system type, region, country, release, and date interval.

Clustering These patient examination log traces can now be clustered according to domain-relevant parameters (e.g., based on their procedures) to further group them into more homogeneous subsets before starting the process model discovery. Relative procedure or application frequencies with respect to the patient examinations that took place on the selected systems in the chosen time period can be visualized in pie charts.

Profiling Based on one or more of these clusters, domain-specific hierarchical process models can now be discovered. Hierarchical process models are a suitable way of visualizing and navigating through the particular usage scenarios on different levels of abstraction.

Furthermore, particular user profiles can be selected and they will be highlighted in the process model.

Exporting The PH plug-in allows to export selected user profiles (e.g., especially representative usage sequences), so that they can be easily included in test scripts. As a consequence of using actually observed usage sequences as input for these test scripts, the system can be tested based on *real behavior* (i.e., not assumed behavior) already prior to product launch.

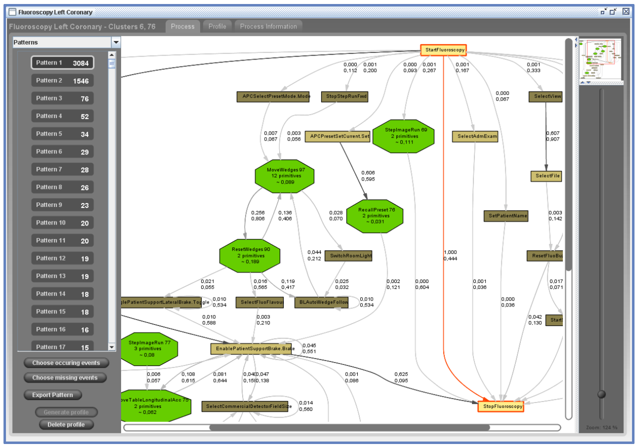

Figure 3. Screenshot of the PH plug-in: After filtering and clustering the patient examination traces, process models are created on different levels of abstraction.

Consider the following example of using this plug-in: We select all systems of the type FD20 in the United States and Canada for the release Rocket-A and between 1 January and 31 March in 2007. As result, 41 Systems are covered, which have in total 1801 logged days. During these 1801 days, 6012 patients are clinically examined. An average of 3,34 patients are examined per day. For example, after we have started the analysis based on two selected clusters, which cover 438 patient examinations, we find out that the most frequent user profile (with 59 out of these 438 examinations) is the sequence *PatientPreparation*, *Procedure Left Coronary*, *Procedure Abdomen*, and *PatientCompletion*. The user can then further expand procedure nodes to investigate their lower-level command structure. A process model on the lowest abstraction level is currently shown in Figure 3, where all fluoroscopy runs of all patient examinations in the context of the Left Coronary procedure are visualized. The most frequent command pattern is selected, and thus highlighted in the model. As a result, we learn that the vast majority of the fluoroscopy runs (3084 runs correspond to almost 70%) of these patient examinations only consist of the commands *StartFluoroscopy* (cf. top-right rectangle) directly followed by *StopFluoroscopy* (cf. bottom-right rectangle). This is a surprising result as the testers expected to see actions in between. Thus, they would have tested the system most of the time with other commands in between, while this does not reflect the usage in real-life.

The PH plug-in has been used and evaluated by Philips employees in the SI&T department throughout the whole project. Next to surprising results as described in the example, they could also confirm their expectation that the machines are generally used in very diverse ways, i.e., no real basic flow, or a few dominant main flows, can be determined over all the systems. Furthermore, the used functionality varies, for example, with respect to the geographical region in the world. At this point in time, up to 90 people within PH have been trained to use the ProM plug-in for analyzing the actual application usage in the field, and from July 2008 on the automatic tests scripts will be enhanced by usage sequences extracted from real-life log data as described. It is expected, that this will increase the reliability of the systems in the field. In particular, it is now possible to measure whether reliability requirements (e.g., in terms of MTBF) are met *before* releasing the product on the market. Furthermore, the awareness of the importance and the potential of log data has been increased within the organization. Follow-up projects are planned to, for example, investigate potential correlations of certain usage patterns and observed software and hardware failures (which can also be extracted from the log data).

7 Outlook and Future Work

For any sufficiently advanced and complex application or machine, the manufacturer typically has a very limited understanding of how exactly the system is being used in practice. This is especially true for generic applications (e.g., tools) and highly configurable, interactive applications (e.g., software). In this study, we have developed a process mining solution for a deployed application which is highly complex: an interactive X-ray system with a high degree of freedom for users. We have discussed the challenges of converting and handling large amounts of ambiguous input data, and we have shown how the problem of process diversity and complexity can be efficiently countered by interactive methods like trace clustering and fuzzy mining. Further, it has been exemplified that available domain knowledge can be leveraged to adjust and extend these methods.

PH is now actively using this customized tool for exploring and analyzing their deployed systems' behavior, and plans to extend the use of process mining also to other areas. Monitoring of deployed applications is not limited to PH and can also be used by other organizations to help focus and improve the development of new products. It is obvious that there are many manufacturers which could equally benefit from an improved insight into their products' usage. Many advanced systems are connected to the internet, which allows for efficient log distribution. We think that monitoring deployed applications will become more commonplace in a few years. For many products, great amounts of resources are spent on implementing functionality which is hardly ever used in practice. Once the manufacturer has a clear understanding of what parts of their products are most important, they can efficiently focus their resources, reduce cost, and improve quality at the same time. For hardware or embedded systems, a concise view on system usage can also help improving their safety and reliability. Finally, monitoring deployed applications complements structured usability testing of interactive systems (e.g., frequent action flows might be supported by the user interface in less steps). Process mining has proven to be a valuable tool for analyzing deployed application usage. Furthermore, the concept of interactive tools, which enable exploration by the analyst, has been found very appropriate for real-life analysis by domain experts. Future process mining approaches should focus on improving the visualization and simplification of complex, unstructured behavior.

The monitoring of deployed application usage holds great potential value for many scenarios and purposes. Especially for development-intensive industries with flexible products, it has the potential to revolutionize the complete product lifecycle, from inception, to development, up to service and diagnostics. Process mining can deliver a powerful set of solutions for providing factual and appropriate insights into product usage. These insights are the foundations upon which more efficient and customer-focused product design and maintenance processes can be built.

Acknowledgements

This research is supported by the IOP program of the Dutch Ministry of Economic Affairs and by the Technology Foundation STW. The authors would like to thank Wim van Rooij from PH for his continuous support and enthusiasm.

References

[1] W. Aalst, B. Dongen, C. Günther, R. Mans, A. Medeiros, A. Rozinat, V. Rubin, M. Song, H. Verbeek, and A. Weijters. ProM 4.0: Comprehensive Support for Real Process Analysis. In J. Kleijn and A. Yakovlev, editors, *Application and Theory of Petri Nets and Other Models of Concurrency (ICATPN 2007)*, volume 4546 of *Lecture Notes in Computer Science*, pages 484–494. Springer-Verlag, Berlin, 2007.

[2] W. Aalst, H. Reijers, A. Weijters, B. van Dongen, A. Medeiros, M. Song, and H. Verbeek. Business Process Mining: An Industrial Application. *Information Systems*, 32(5):713–732, 2007.

[3] W. Aalst, B. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering*, 47(2):237–267, 2003.

[4] W. Aalst, A. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[5] R. Agrawal, D. Gunopulos, and F. Leymann. Mining Process Models from Workflow Logs. In *Sixth International Conference on Extending Database Technology*, pages 469–483, 1998.

[6] J. Cook and A. Wolf. Discovering Models of Software Processes from Event-Based Data. *ACM Transactions on Software Engineering and Methodology*, 7(3):215–249, 1998.

[7] A. Datta. Automating the Discovery of As-Is Business Process Models: Probabilistic and Algorithmic Approaches. *Information Systems Research*, 9(3):275–301, 1998.

[8] B. Dongen and W. Aalst. Multi-Phase Process Mining: Building Instance Graphs. In P. Atzeni, W. Chu, H. Lu, S. Zhou, and T. Ling, editors, *International Conference on Conceptual Modeling (ER 2004)*, volume 3288 of *Lecture Notes in Computer Science*, pages 362–376. Springer-Verlag, Berlin, 2004.

[9] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, New York, NY, USA, 2000.

[10] M. Dumas, W. Aalst, and A. Hofstede. *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley & Sons, 2005.

[11] C. Günther and W. Aalst. A Generic Import Framework for Process Event Logs. In J. Eder and S. Dustdar, editors, *Business Process Management Workshops, Workshop on Business Process Intelligence (BPI 2006)*, volume 4103 of *Lecture Notes in Computer Science*, pages 81–92. Springer-Verlag, Berlin, 2006.

[12] C. Günther and W. Aalst. Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In G. Alonso, P. Dadam, and M. Rosemann, editors, *International Conference on Business Process Management (BPM 2007)*, volume 4714 of *Lecture Notes in Computer Science*, pages 328–343. Springer-Verlag, Berlin, 2007.

[13] J. Herbst. A Machine Learning Approach to Workflow Management. In *Proceedings 11th European Conference on Machine Learning*, volume 1810 of *Lecture Notes in Computer Science*, pages 183–194. Springer-Verlag, Berlin, 2000.

[14] K. van Uden. Extracting user profiles with Process Mining at Philips Medical Systems. Master's thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2008.