Case Study in Process Mining in a Multinational Enterprise

Paul Taylor, Marcello Leida, Basim Majeed

▶ **To cite this version:**

Paul Taylor, Marcello Leida, Basim Majeed. Case Study in Process Mining in a Multinational Enterprise. 1st International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA), Jun 2011, Campione d'Italia, Italy. pp.134-153, 10.1007/978-3-642-34044-4_8 . hal-01515545

HAL Id: hal-01515545

https://inria.hal.science/hal-01515545

Submitted on 27 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Case Study in Process Mining in a Multinational Enterprise

Paul Taylor[1], Marcello Leida[2], and Basim Majeed[2]

[1] BT Innovate & Design, Adastral Park, Martlesham Heath, Ipswich, UK
[2] EBTIC (Etisalat BT Innovation Center), Khalifa University, P.O. Box 127788, Abu Dhabi, U.A.E.

Abstract. Process mining has become an active area of research and while there are numerous papers on approaches to process mining there are fewer detailing its application to real industrial scenarios and its applicability in these spaces. In this paper we introduce the approach to process mining used in a number of multinational enterprises and then reflect upon the issues that have been encountered during our ongoing work. In our opinion these issues are a clear example of the challenges that need to be addressed during business process discovery from heterogeneous data.

Key words: Process Mining, Data Driven Process Discovery, Industrial Application, Case Study, Process Improvement

1 Introduction

With the industrial revolution, the introduction of mass production forced a drastic shift from artisanal crafting of products, usually performed by one person, to the large scale production of goods, where each person was responsible for a single step in a process typically orchestrated by someone that was not involved in the execution. One of the first persons to define this series of steps as a formal process was Adam Smith [**?**], where he describes the series of steps required for the production of a pin.

With increasing demand for goods, the increase in competition and the need for a reduction in costs required the reengineering of the existing processes. In most major companies this was already being considered even before it was formally defined during the 1990s by Hammer [**?**] and Davenport [**?**]. They defined a procedural approach for improvement of business processes to ensure correctness and to improve effectiveness, efficiency, and compliance with statutes and protocols.

In the present day business processes are well understood and formally defined with standardized representations and associated reengineering procedures. Business analysts exploit these formal representations by defining performance measures and quality constraints to analyse and improve specific aspects of the enterprise. Such business improvement activity is based on the process captured

and defined using a formal language. However with the increase of the complexity of the formal languages used to model processes we witness also an increase in the distance between the formal process model and the process that is actually being executed [**?**, **?**].

Large enterprises are required to actively respond to market demands and market evolution, therefore it is necessary to ensure control of crucial activities and to facilitate the reengineering of the processes running across the enterprise. This activity is usually performed by defining strategic requirements over the process (i.e. reduced execution time, minimising the amount of repeated work, removing loops, and so on). This set of requirements is relatively easy to capture and monitor in a fully automated process execution environment, however the same cannot be said for complex processes running across multiple departments, or for processes involving human intervention.

Broadly speaking every activity in an Enterprise (or an SME) can be seen as a step in a more comprehensive and complex process. Most enterprises capture the trace of the execution of activities within a process in several different ways and for several different reasons. Most of the information systems keep track of the activities being executed within them: Enterprise Resource Planning (ERP) systems, Business-to-Business (B2B), Customer Relationship Management (CRM) systems are each able to generate activity logs. Others are explicitly captured by Workflow Management Systems (WMS) which log the beginning and the conclusion of activities. Some activities such as email exchanges are generated just to have a tangible record of an agreement established by word of mouth. Others are formal legal documents that establish some kind of relationship between the parties involved. Each of these pieces of information contains knowledge about the enterprise, which can be used in several different analyses. One way to exploit this information is to reconstruct the process executing in the enterprise; this allows the behaviour of those various processes and the actors involved to be monitored in order to identify reasons for bottlenecks, incorrect executions, rewinds, loops and other issues preventing the process from matching the desired strategic requirements. However the process of extracting measurable evidence from the enterprise knowledge base is a non-trivial activity, which requires understanding and analysis of the activities and their interactions to transform them in measurable models.

In this paper we present a series of case studies directly from our experience with the analysis of real process execution data collected from several different systems, each of which uses a different technique to capture data. We believe that the cases presented in this paper will help make the research community more aware of the issues that could arise when dealing with enterprise-scale data and influence the future research directions in the area of business process analysis. This paper is divided as follows: Section **??** presents some background information about the tool we used to perform the business process analysis described in the paper. This section will provide the knowledge required for a better understanding of the use cases presented. In Section **??** we discuss existing work detailing other real-cases of analysis of large-scale processes. In Section **??**

we present a set of relevant case studies regarding the application of business process mining techniques in a multinational enterprise, which is the focus of this work. We conclude the paper by bringing together some final considerations in Section **??**.

2 The Aperture Process Mining Tool

As introduced in the previous section, one of the major issues in process analysis is to capture and interpret data correctly. There are languages such as BPMN [**?**] and BPEL [**?**] that are used to explicitly define a process and capture execution information for fully automated systems (e.g. web service orchestration). However in most cases integration of BPEL engines in existing and on-going processes is a non-trivial undertaking and often is an effort that enterprises prefer to avoid unless a minimum Return of Investment (ROI) is guaranteed. This effort might also be undesirable where the process is not formally captured and it exists only in an idealized sense in the mind of the people involved; this is particularly the case for many SMEs. Therefore there are many situations where a process model is of very limited use or is simply not available; in which case the process model needs to be inferred from the information created during process execution.

To respond to this specific requirement we are using an internally developed process mining tool known as **Aperture**. It has been designed to provide an analysis that is focused not upon process mining itself but upon process analysis and business improvement. One of the design goals of Aperture is that it should be usable by those without experience or expertise in process mining, and should fit into the toolkits used by those people in the target enterprises. Aperture is effective with process data describing either sequential or parallel behaviour and requires a minimum of data to be effective (currently a process identifier for each process instance, a task identifier for each activity, the start time of the activity and the end time of each activity).

The important goals for the process mining algorithm used in Aperture are *predictability*, *plausibility*, and *traceability*. Predictability refers to the ability of the tool to produce the same model given the same input data, plausibility means that the generated models should be viewed with confidence by any person knowledgeable about the process, and traceability means the ability to trace the components of the generated model back to the source data. Traceability is very important in practice as it is quite regularly needed to convince skeptics of process mining within the business and operational units whose initial reaction is often to reject the mined model as being unrealistic, or just plain wrong. However the ability to trace each part back to the source builds confidence and leads to increased buy-in from the users.

Aperture is able to reconstruct the process model form data stored across heterogeneous sources of information, providing the users with a unified framework to analyse processes and tasks that are executed across the enterprise, that otherwise would be extremely difficult to monitor and improve.

Trained users can perform analysis across several business domains without knowing domain specific aspects. The more the data that is available, the greater the accuracy and flexibility of analysis that can be performed.

We will now briefly introduce the data model at the base of Aperture tool; this will help to fully understand the case studies presented in the next sections. The main elements of the back end data model are the *Process Instance* and the *Task Instance*:

– each *Process Instance* describes one job or order or process execution (service fulfillment or fault repair processes for example);
– each *Process Instance* consists of a number of *Task Instances* linked together by a temporal relations (activity `A` is executed before activity `B`).

The minimum information required to create a process model to be used in Aperture is a process instance with a minimum of one task instance. The minimum amount of information to create a task instance is the start time and end time of each task instance. Process instance start time and end time can be derived from the starting time of the first task and the ending time of the last task if they are not explicitly provided.

To convert the raw data from source systems and make it available to the users for analysis, the tool follows a computer assisted multi-phase approach (Figure **??**). The first phase is to acquire workflow data from the process under study. The identification of appropriate data and providing a source data set is performed manually. Acquiring the data is often challenging as it might come from multiple independent systems which may not have corresponding linking keys. Data acquisition is an extremely delicate phase since if there is misalignment with the process identification the resulting model will be of no use. The second phase is in many ways the most important: a competent computer programmer will implement a data importer that will extract and transform each process instance from the data acquired in the first phase into Aperture's internal format. This phase is supported by the process mining algorithm of Aperture that will be used to generate the structure of the process instance from each group of tasks.

The algorithm used by Aperture to connect tasks into a process instance is based upon minimum-spanning-tree algorithms [**?**]. We attempt to minimise the time spent in between tasks, thus satisfying the predictability, and traceability goals. To ensure that the models are plausible, domain knowledge can be incorporated into the importer thus placing appropriate constraints on the mining algorithm. An example of where this ability has been used is in a system handling complex orders which may have more than one sub-order, to achieve the plausibility goal it was necessary to ensure that each sub-order was connected independently to avoid cross-contamination and loss of conceptual clarity.

In the third and final phase the data is made available for interrogation via the provided user interface, which is a browser based application (Figure **??**) written using the Google Web Toolkit (GWT)[1]. This allows the user to view the

[1] `http://code.google.com/webtoolkit/`

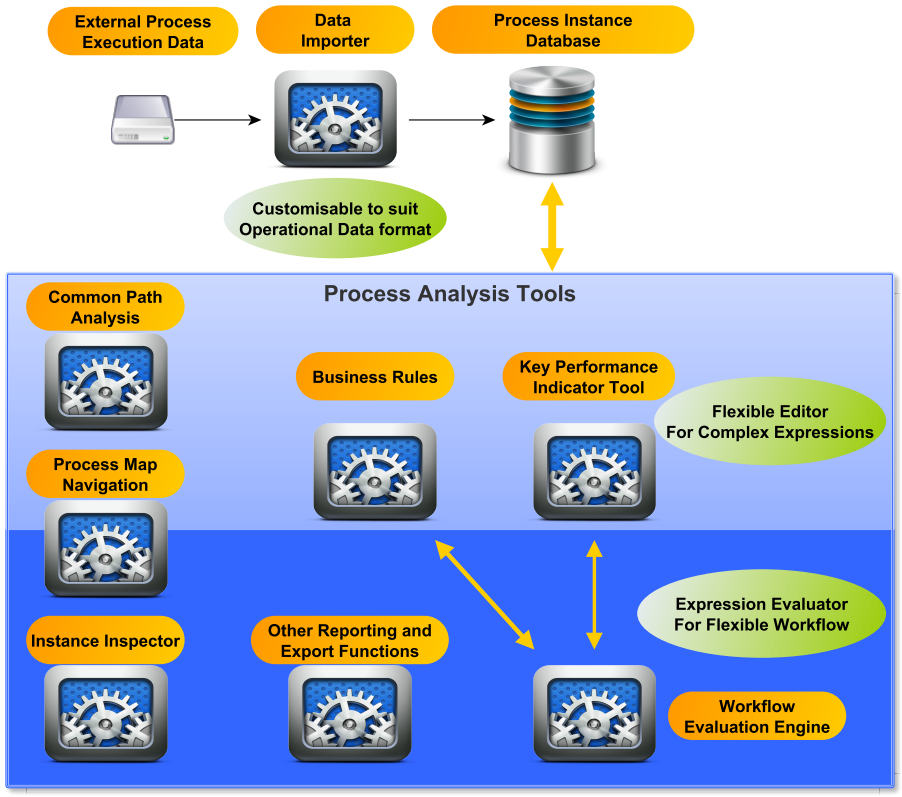

Fig. 1. The architecture of Aperture

data from the overall process model (Figure **??**) right down to the level of the individual process instances. In addition to the minimal information required, the rest of the source data is imported and attached to the processes as process and task attributes. These attributes can be used by the user to filter sub-groups of data, and to run calculations against the dataset without requiring a re-importing of the data which speeds up comparative work considerably.

This approach differs significantly from that followed by other process mining tools (such as ProM [**?**] or BPM|one from Pallas Athena[2]) in that we do not process multiple process traces directly into a single model, rather we consider each process instance separately. The system then takes care of generating the compound process models and any associated analytics.

2.1 Analysis

The first and most intuitive of the analysis options offered by Aperture is the creation of the compound process model, as in the example in Figure **??**. This will

[2] `http://www.pallas-athena.com`

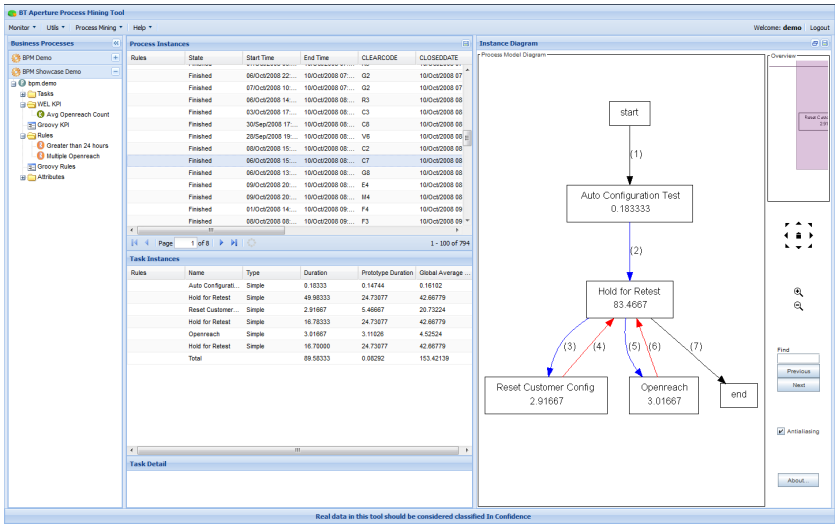

Fig. 2. A Screenshot of Aperture's main screen

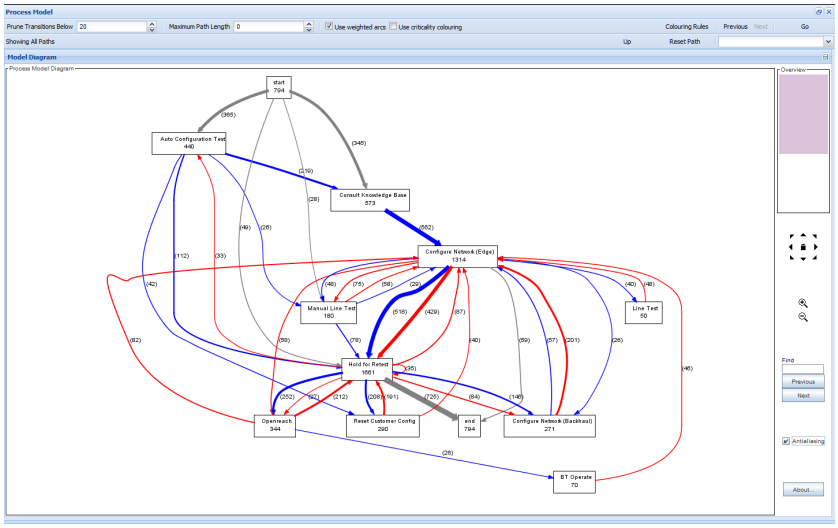

Fig. 3. A Screenshot of Aperture's process model view

combine the mined models for each process instance into a single graph showing the overall shape of the process. Weighting of the arcs of the graph is used to indicate frequency allowing users to see at a glance where the process is running *hottest*. In addition, the tool allows users to visually distinguish in the graph between the first time a task is executed in a process and where it is repeated, the first occurrence is connected using a blue (or solid) arc and the second using a red (or dashed arc). In a compound diagram this shows clearly where most

of the repetition (or *rework*) is occurring and therefore where efficiency savings could potentially be made.

A set of more complex analysis tools can be used starting from the process model graph: drill down facilities are provided on top of the model which allow the user to see, for example, if a particular problem transition is only associated with a particular group of process instances.

The second of the analysis options offered by Aperture is the extraction of the common paths through the process which are known as *prototypes*. Each prototype is a distinct concrete ordering of tasks (either sequential or parallel) as executed in the source system(s); in Aperture the process instances that have the same concrete sequence are considered to be in the same prototype. These prototypes can then be examined for frequency, and for the commonality of their attributes, allowing correlations between attributes of the process, and the execution path to be discovered.

Furthermore the tool provides ad-hoc querying for Key Performance Indicators (KPIs) using either a custom workflow oriented language called Workflow Execution Language (WEL) [**?**] or more complex measures defined through the Groovy[3] scripting language, interacting directly with the Aperture data model. In conjunction with the ability of the tool to rapidly extract subsets of data this allows the user to rapidly get an insight into the relative performance of those subsets in terms that are familiar from strategic reporting (once the formula for the KPIs has been entered into the tool).

3 Related Work

Existing literature describing systems able to infer the process model from arbitrary process execution data is not extensive, furthermore we do not intend to focus on the differences between Aperture and report systems based on top of WMS, ERP, CMS because they are out of the scope of this paper.

In the set of available works, we identify the ProM framework [**?**] as the pioneering and leading research project for business process mining from execution logs and a system closer to our approach; however the definition of a new process mining tool is not the reason for writing this paper. We introduced our system in Section **??** for the sake of completeness and clarity, but the reasons that led us to collect a set of case studies of real applications of a process mining tool are led by the experience our group gained in the deployment of Aperture tool in enterprise-scale environments, during the last few years.

The list of papers describing case studies of the application of process discovery techniques in enterprise or government-scale level is also not extensive as we have only been able to identify few published realistic pieces of work.

In [**?**] the authors describe the use of the ProM system for the analysis of the Dutch National Public Works Department. The paper described how the logs, generated by a WMS were analysed and how the ProM tool was used to extract

[3] `http://groovy.codehaus.org/`

the process model, the social network of the organization and the process logic based on decision trees. However the step of importing the data into ProM is not described in detail: the authors mention that they converted the original log to MXML[4] [**?**] format used in ProM. But the conversion process is not reported and if any issue arose during the import of the data, there is no record of it in the paper.

In [**?**] an application of the ProM framework in the Health sector shows how the tool is used to extract the process model from event logs. In this particular case an application of the clustering technique is used to extract clusters of similar process models in order to filter wrong models or models representing outliers. The raw data used contains information about 627 gynecological oncology patients, collected by the billing system of the Academic Medical Center (AMC) hospital in Amsterdam which is converted into MXML for use with ProM. In this case the authors mention a problem with the billing system caused by the fact that the timestamps associated to the events refers only to the day the event happened. The authors mention in the paper that this situation had led to wrong ordering of events happening in the same day. Moreover the data import task has been preceded by a preprocessing of the MXML document to aggregate events by department, this way the resulting process model was simpler to understand and analyse. However in the rest of the paper the authors do not mention if and how the problem influenced the resulting analysis.

The case studies reported in the literature focus mainly on the process mining aspect with little or no mention of the issues relating to the data collection process. The set of examples in this paper will also address the possible problems and issues that need to be considered before importing the data into a process mining tool. This will provide valuable experience to process mining practitioners as they tackle new data sets by helping to avoid common mistakes, leading to improved results.

4 Case Study Issues

This main section of the paper, discusses the issues and common patterns encountered while process mining across a variety of processes in a number of diverse multinational organisations. The most significant challenges are to understand and reconcile data models from systems which can be very old, and to acquire appropriate process data from these systems when the data that they record was defined in another era for other less rigorous approaches.

These cases do not form an exhaustive list of all issues that we have encountered, but those presented have occurred with sufficient frequency that we are confident that they are likely to be present in most enterprises to a greater or lesser degrees.

[4] `http://www.processmining.org/logs/mxml`

4.1 Variation in Standardized Processes

The first use case to present is the analysis of how a process that is standardized across the entire enterprise is executed with different performance for reasons that cannot be inferred from the process execution data directly. The process we are focusing on is a service provisioning process of a multinational enterprise.

In the provision of services globally one approach to ensure the performance of provisioning tasks would be to use a centrally defined process model that is well supported and well understood by all the departments that execute it worldwide. One of the processes we had the opportunity to analyse with the tool we developed was of this nature. The process improvement team examining this process requested the analysis of the process execution data in order to extract the process workflow.

The goal of the analysis process was to identify areas for performance improvement in the global process model which could then be tested and deployed around the world.

Once the process model was extracted from the process execution data provided by the enterprise, it was clear that what was understood to be a single process had a large amount of unexpected variation. As-designed executions were only a small percentage of the overall executions of the processes analyzed; all the remaining process instances deviated significantly from the original process.

Even though the original process model was lost in a plethora of different executions, our goal was to identify areas of inefficiency such and looping, repeating tasks and what is known as *rewind* or *rework*: the situation where the execution has to be returned to an earlier part of the process because work there has been left undone. Unfortunately the size and complexity of the generated model were such that it was not possible to identify any such issues immediately.

This being the case, we carried out a process mining exercise in order to identify which criteria was best able to group the process executions into models based on the issues we have previously discussed: such as bottlenecks, repeated work and so on. The separation of the various process models would allow the process improvement teams to conduct a more detailed analysis and interviews with the people involved in the execution of the process to uncover the underlying reasons behind the variation, so that it could be addressed.

For example, if the process model for a particular product type, say product A showed that in almost all executions the credit checking task was performed multiple times while the process model for every other product showed only a single execution, then the process improvement team could investigate that specific task to try to uncover the reason behind the repeat for product A. This shows our general approach of using process mining as a tool to inform, rather than a tool to indicate potential solutions directly.

From the analysis performed using Aperture it was possible to highlight that the most striking of the variations in the set of performance metrics defined to evaluate the process executions, was the difference between the processes executed in each of the order management teams around the world. Therefore

the criterion that was used to separate the process model was the geographical location of the various order management teams.

In order to confirm this assumption, we used our process analysis tool to generate models of the process as it was followed in each of the separate order management teams. Once each of these models had been generated we proceeded to visually identify the areas of largest variation, together with the analysis of the performance indexes, and the areas exhibiting the undesirable properties we have previously discussed. Examples of two of the models created using the geographical location as clustering criteria can be seen in Figure **??** and Figure **??**.

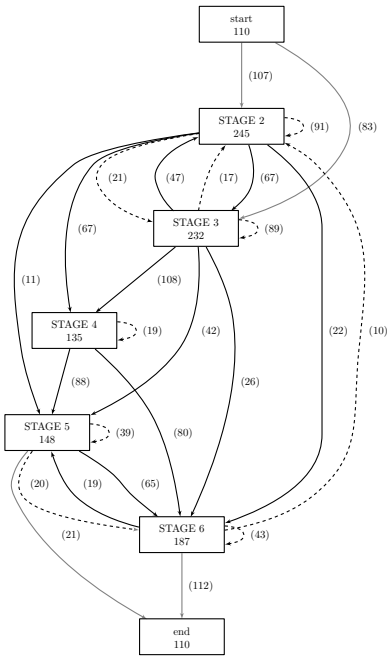

Fig. 4. Process Executed in France

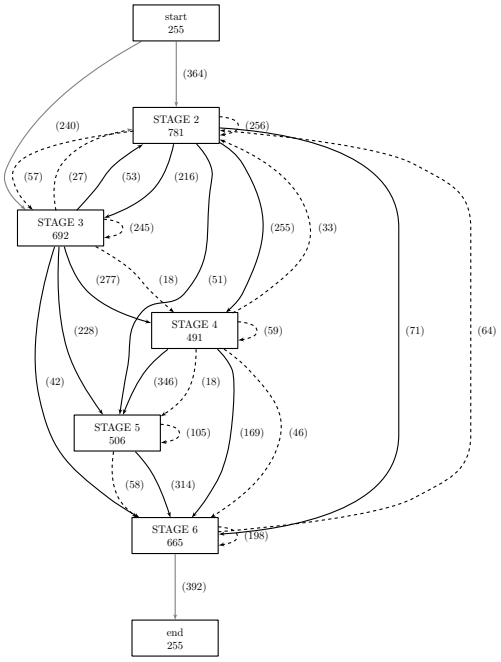

Fig. 5. Process Executed in Germany

As it is possible to notice from Figure **??** and Figure **??** there is a marked contrast between the processes executed in France and in Germany. To reiterate: the process model that should be executed both in France and Germany should be the same in this case.

For the period that these models are referring to, the evaluation of the performances of the processes was markedly better in France than Germany. This gives a significant hint that we could improve the performance of the German team if best practice from the more efficient French implementation is adopted. The outcome from this analysis has been to gather representatives of the two regions and examine some meaningful orders identified by our process analysis

tool and discuss the rationale behind the decisions made during execution to identify the root cause of the differences and the deviation from the standard process.

The complete analysis indicated that while variation between the regions was common; the most conformant process with the standard process model and also the one with the minimum of process issues was the process model followed by the French team, while the process showing the most frequent process issue was the one followed by the German team.

Since this process is standardised the process improvement teams started to look at potential reasons for the differences that were not captured by process execution data, this may have included variations in the process inputs or setup (e.g. exact product configurations, equipment requirements, expertise of the employee performing the installation), insufficient or incomplete documentation of the original process model or simply the way the local process was interpreted and executed.

The process improvement teams took this information away to one of the regular meetings of representatives from the regional teams so they could present our findings and try to gain some insight from the operational level as to the reasons for the features identified. Discussions with the process team following this meeting revealed that the process models, and diagrams we produced had been invaluable in drawing out additional information in the discussions.

4.2 Documentation Out of Synchronization with Reality

This case study highlights the importance of collecting the process execution data in a way that is aligned with the process model that was defined and put in place. This example is useful to point out the fact that extracting the process model from execution data is a not straightforward task.

Process improvement projects that do not capture process execution data for analysis by a process mining tool normally perform analysis on the designed process model when looking for areas to improve. This is a problem when the designed process is significantly different to the process that is being executed. In our experience we have observed many situations where this is the case and this happens for a number of very different reasons, for example:

- when a change in the system that executes the process, forces a change to the sequence of constituent tasks;
- when an exceptional circumstance arises (natural disaster, strike action, etc.) leading to temporary process changes which are unintentionally adopted permanently as staff assimilate the temporary process;
- when an upstream process drastically changes its performance, leading to a change in the downstream process execution, e.g. change of the task execution order;
- when a process execution includes constraints from local conditions that are not captured in the process definition;

– when a process work-around that is discovered by operators becomes the local standard as it is perceived to be more efficient than the designed process.

However from our experience we identified one example that stands out as being a recurrent issue for process mining tools and practitioners to be aware of.

This issue is the misalignment between the level the process designers are working at and the level that the process executors are working at; including the systems that are being used to manage the workflow though the execution phase. To be clear, this is not an issue with systems, or with the data itself, or data collection step; it is a process design issue.

First, let us introduce queues. A queue is simply a collection of orders that are waiting for action. Each queue in the workflow system is assigned to a particular team for them to take work from, this means that there is an approximate $1:1$ mapping between queue and team. In situations where each team has only a single task to do in the process, data on when a process entered a queue and exited a queue can be used as an approximation of the time that team spent working on that task.

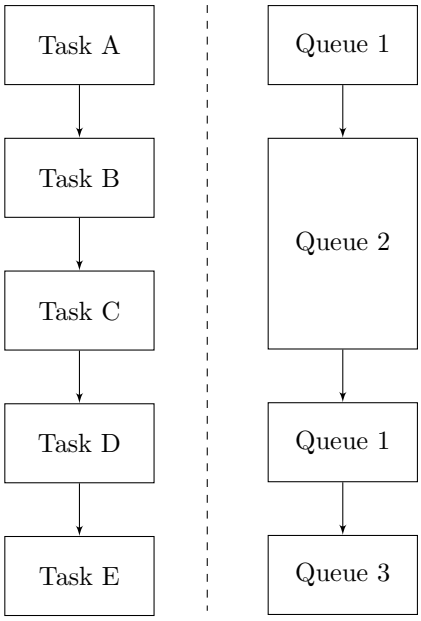

Fig. 6. An example showing how a process instance may look if data about tasks, or queues is stored

For example, imagine we have `Task A` to `Task E` where `Task A` and `Task D` are serviced via `Queue 1`, `Task B` and `Task C` by `Queue 2`, and `task E` by `Queue 3`. Figure **??** shows the difference between the process models according to tasks (left) and queues (right). It is possible to notice in this simple example

how information is lost, making it extremely difficult to interpret the results in terms of analysis of the process under investigation.

We have now seen that the queue abstraction can mask data about the process from being recorded in the logs of the workflow system, and this is a widespread issue particularly in older workflow systems. If the process designers take this into account when designing the process and ensure that there is a $1:1$ mapping between task and queue, then the process can be recovered with a reasonable degree of accuracy, however if the process designers ignore this abstraction then the process designs will not map onto the data that is provided by the workflow system, meaning that the process model discovered by process mining cannot match the process as designed.

One concrete circumstance in which we have often observed this pattern is where a product has many specific configurations that the customer might choose from. In this situation the teams creating the workflow will create product specific queues for each team e.g. Order Management (Product X), Field (Product X), since there is just one product. The problem arises when the process designers, who are a separate team, then treat each specific configuration (erroneously) as its own specific product, so if there are five configurations, the workflow team will encode a single process into the queues in the workflow system, but the process designers will provide five separate process designs for the actual users of the system to utilise.

This is a major problem for process mining because the generated models will be using the queue abstraction from the workflow system and that will not correspond to the process model that is provided to the process analysis teams. This means that the analysts may propose/implement solutions that will not be possible to implement or that will not generate the expected outcome. The easiest way to mitigate this problem is to maintain a mapping between the queues and the tasks performed in those queues and use the mapping to convert process execution data with workflows aligned with the conceptual model.

However in practice this mapping either does not exist or it can get quickly outdated for the reasons we described previously. Moreover organisational changes and previous process improvement efforts may have caused multiple queues to perform the same task, or a single queue to perform multiple disparate activities. An example mapping for the process in Figure **??** is given in Table **??**.

Table 1. An example mapping of tasks to queues

Task	Queue
Task A	Queue 1
Task B	Queue 2
Task C	Queue 2
Task D	Queue 1
Task E	Queue 3

In addition to having to maintain this mapping through business as usual process improvement and process change work, a difficulty is introduced if there is reorganisation within the business. Let us imagine that we have 2 products in this workflow system, and each of these products has 5 different configurations that have different process designs in the way described above (so 1 workflow system, 10 process designs). Since each of these designs requires that an engineer is dispatched, the team responsible for dispatching is assigned a common queue called FIELD. There is a simple mapping between the dispatch task in the process design and the field queue so there is no issue. Now the business is reorganised, and it is mandated that the field teams for each of the 2 products are to be managed separately, this means that we have to create a new queue and change the assignment of each queue to the appropriate new team, however we would also have to update 10 separate mapping documents (one for each of the process designs) to take this change into account. In a real system containing hundreds of products, and potentially thousands of configurations maintaining an accurate mapping between queues and tasks is a practical impossibility.

The lesson we have taken from this is that it is extremely important to have an accurate and consistent mechanism to translate between the different abstractions used (where they must be used) otherwise process analysis will be performed on incorrect models and the resulting decisions based on faulty assumptions. The ideal solution is that the components of the workflow system match the components of the design so that no mapping is required and there are no conflicting abstractions.

4.3 Deficiencies in Workflow Systems/Recorded Data

It is extremely important to have correct and consistent information available for process mining to derive accurate models and measures for the process under study, and often such data is not available. In most cases this is due to a deficiency in workflow management systems that cause misleading or incorrect data to be provided for process mining. This may not be due to a faulty workflow management system, as much as an issue with the way it is being used and how it records data.

We came across this issue during the analysis of a process based on the grouping of tasks into stages of execution. These were then organised in the process flow in a way that a stage was able to start when all of the tasks in the previous stage were completed. If the workflow system is able to record the times these tasks are actually performed (i.e. started and concluded) then no issue exists since we can analyse the workflow in the standard way.

However if the workflow system, like in the case we analysed, is able to collect only the time the task becomes ready for execution, then the data generated is misleading as it looks as though the various tasks in a stage are all running concurrently. This situation is exacerbated if some of the tasks have a long execution time, or a long lead time (such as tasks awaiting equipment, or an appointment), this means that a task to confirm the delivery of an item which

takes only few minutes of effort, could show in the data collected by the workflow system as having a duration of several weeks.

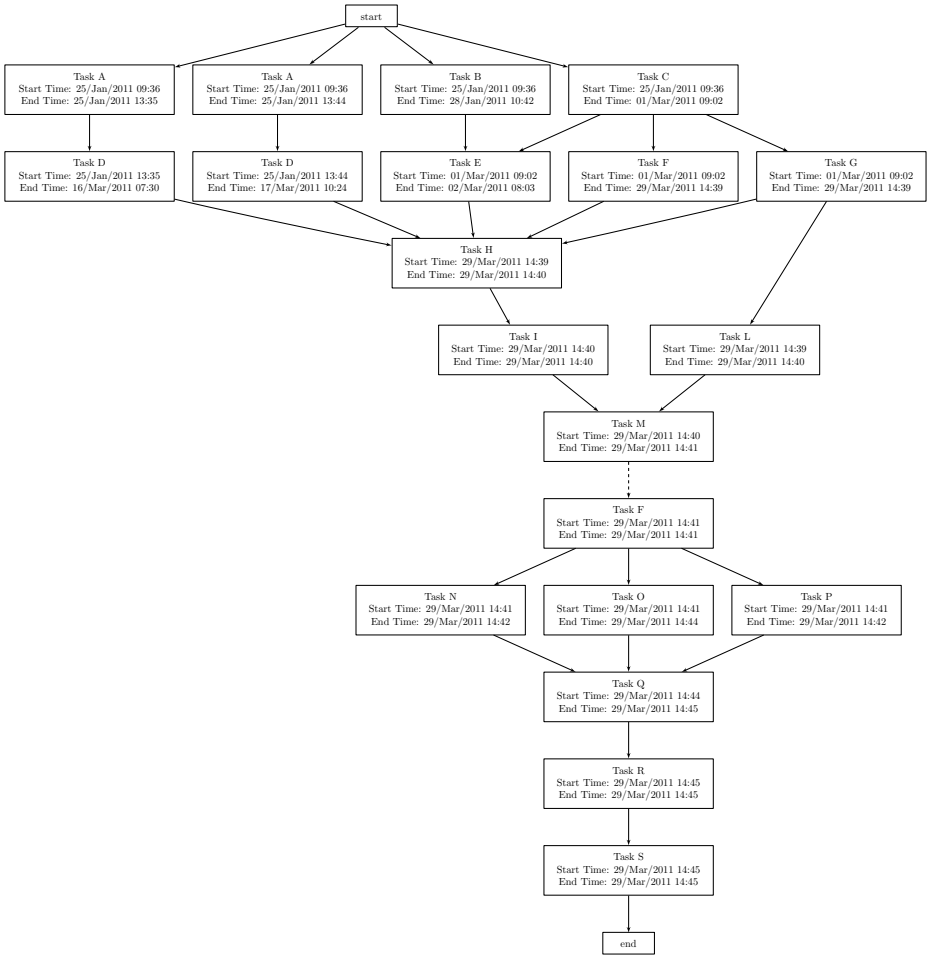

Fig. 7. Diagram showing pattern caused by staged executions

Figure **??** shows an example of such a workflow: as it is possible to see there is an initial stage composed of four tasks which are immediately opened for execution with identical start times which refers to the start time of the stage, the end times of these tasks however show a more linear sequence, despite this, because the start times are all equal it is not possible to extract a sequential flow of the execution. In the later part of the diagram you can visually identify an example of this where the start of the final stage where a group of 3 tasks are started when Task F is finished, each of the 3 tasks has an identical start time because they all become available when the stage is opened.

Ideally the workflow system could be changed to record the real starting times of the tasks so a more accurate duration could be derived. Unfortunately such a change is often impractical in large entrenched or legacy systems. There are then two alternatives available; either treat the data as accurate, leading to the situation in Figure **??**, or we could force the tasks into a sequential sequence, this however is prone to errors and therefore misleading results. Therefore the evaluation of Key Performance Indicators (KPIs) or decisions taken on this process model needs to be carefully evaluated before they are put in place, since the KPIs may be measuring a situation that is not reflecting reality.

4.4 Inadequate Data Quality

The fourth case study that we identified is related to the quality of the data acquired by the enterprise. It is important to recall, in order to fully understand this case, that the process analysis tool we developed and applied in the use cases described in this paper, works mainly with the start and end times of the task executions, and most of the cycle time analysis and performance metrics are derived from these measures.

In this case, and many others that we have investigated, the data being available for analysis contained a number of issues such as the start and end times resulting in negative process durations, missing execution times, invalid dates, and ambiguous or misreported task identifiers. When issues like this are found it is important to investigate if the data collection process put in place by the enterprise has been properly defined or not.

In order to identify the reason for these errors it is possible to use external information from users, or other data sources to mitigate them. Such corrective information can be incorporated into the Aperture importer, which transforms the raw data to be analysed into an appropriate format for the tool. However, as for the previous case, it is important to ensure that the users are aware of the compromises made when they are evaluating the results of the process analytics. In addition, it would also be advisable to validate the measures obtained from the tool against metrics that are calculated outside it to give meta-information that can be used as a quality measure for the data. This is important in order to ensure that the business management is not being misled through low quality data.

Table 2. An example of some source data with quality issues.

ID	Task Name	Start Time	End Time	Order Status
1	Task A	2010-01-01 12:00	2010-01-01 12:00	CLOSED
2	Task B	2010-01-01 12:00	2009-01-12 12:00	CLOSED
3	Equipment ordered by Sandra for delivery on Thursday	2010-01-02 12:00	2010-01-03 12:00	CLOSED
4	Task D	2010-01-03 12:00		CLOSED
5	Task E		2010-02-04 12:00	CLOSED

An example of data provided in this situation is shown in Table **??** which is based on a real data set we have processed. Each row in the table shows a sample of an encountered data quality issue:

Row 1. This row shows information that at first sight seems to be correct. However it is important to analyse if the start time and end time captured by the system are correct. The resulting task might cause a problem for the evaluation of some metrics since the duration is zero.

Row 2. This row shows a task where the end time occurs before the start time. This could have been caused by many reasons: for example, because a user edited task data by adding a new note to this task after it was finished, or a metadata update was performed on it after the task execution was concluded. Another example would be where the data collection system, for some task types, swapped the start and end times. Using Aperture it is straightforward to identify this type of error since the duration of the task will be negative.

Row 3. This row shows an example where the task name has been replaced when the user performing the task has added a note or other information to the process or task. In such situations, depending on the workflow engine it may or may not be possible to determine the original task name.

Row 4. This row shows an open task called `Task D` as part of a completed process. The problem is that from the data, it appears as though the task has never finished, since the end time is missing. This can be mitigated if we assume the task was immediately completed, but it would be equally valid to assume this task ran until the recorded completion time for the entire process. In order to associate the most suitable information with the end time value, knowledge of the underlying process and workflow engine is required.

Row 5. This row shows a case which is similar to the previous one, where the `Task E` is in a closed state, with related end time, but according to the data the task was never started. The same assumptions made for the previous row are also valid here.

If these issues are ignored or go undetected it is easy to see that they could lead to errors being introduced into the process models and resulting analyses.

The approach that we followed to tackle these issues is to liaise with the business improvement team that commissioned the analysis, to identify a suitable solution; however this may not be always possible so a fallback position would be to remove the entire job from consideration to ensure it does not pollute the model. In cases where this approach would be too drastic, then the problematic task could be removed however this is a riskier strategy since it could introduce false dependencies between the remaining tasks

4.5 Undocumented Process Evolution

In the last case study we want to present in this paper, we highlight an example of evolution in the process model that was not driven by a decision in the process design team. In this case we have to deal with the variation from the original model that was due to a temporary change in the workflow caused by the ongoing replacement of machinery.

In this case we noticed that even when the machinery was restored to service some process executions were still following the temporary pattern. In the analysis of the overall process model it was possible to notice by examining the process model before and after the restoration that two process models existed, without management being aware of the second process.

Another process we had the chance to analyse was a process model that was not fully constrained. Due to several reasons (change of the employees, delayed training and so on) the process being executed was the process that was understood by the employee, which was in many cases different from the designed process. For example, tasks in the process were sometimes skipped because they were considered unnecessary by a process actor. Moreover in many cases the performance was better in the modified process than the normal one, but there were also many cases where the process execution flow was generating many rewinds, and other repetition.

The reason was that the task that was skipped was useful in very specific cases and in such cases the process would go into an uncontrolled situation, causing loops and rewinds, since the required task had not been executed. In case of an expensive and long process this is not an insignificant issue.

In these cases our tool allowed the extraction and documentation of these deviations from the standard process model and was also able to evaluate the impact of such deviations in the enterprise. The evaluation of the impact was an important step to enforce the designed process model, since some teams were claiming better performance without taking into account the drastic performance loss in the other executions, causing overall process performances to be worse. In addition, this behaviour can lead to many outlier execution instances, which in some cases are extremely risky.

5 Conclusion

The area of process mining is not simply concerned with finding and visualising the best model that fits the work execution data. In our opinion, a process mining system must have a rich set of tools that allow the practitioners to carry out a variety of performance measurements and other analytics. In addition, process mining techniques that focus on the dominant execution paths and ignore outlier instances lead to the reason behind the outliers staying undiscovered and thus capable of appearing again in the future causing persistent performance issues. In this paper we presented a set of case studies that we consider interesting for the community of professionals involved in the practice of business process

management. We intentionally avoided considering situations where the original process design was not optimal because this is an issue of process modelling rather than analysis; instead we preferred to focus on the issues arising from the system used to capture the data and how this data is interpreted in the extraction of the process model.

The most important issue that we have found in our work is that **data quality is paramount**: techniques for knowledge extraction such as process mining can only be mislead by incidences of erroneous data. This is, of course, the well known garbage-in-garbage-out phenomenon; however we have identified some data quality issues that are particularly troubling for process mining (Sections **??** & **??**).

The other major conclusion from our work is that having access to someone with process knowledge is extremely important for the accurate consumption and evaluation of the data. For example, throughout this paper we have highlighted repeated task execution as a process problem to be identified and avoided, however in some processes such repetition could be beneficial or even necessary. Consider a fault resolution process where the fault is handed to a different business unit in some circumstances, there it would be helpful to test the problem before it is referred to avoid unnecessary referrals and also helpful to test after the fault is received back to ensure the fault is actually resolved. This highlights the necessity for analysis of the results of process mining to be carried out by people who have, or who have ready access to knowledge about the process under examination. If this information is not available, suboptimal decisions can be made both in preparing the data for mining and in the interpretation of the results.

As a result of all the data issues that were encountered during these case studies, it has become clear that the data import process, i.e. transforming the raw systems data into a viable process model, needs to be enhanced by adding domain knowledge that constrains the algorithms in order to first highlight any inconsistencies and also resolve the issues that are discovered by bringing into action a set of business rules that act on the data accordingly. In fact, one of the major benefits that our partners in the business teams have found in going through the process mining exercise is to discover many data quality issues of which they were not aware previously, and that affected the existing reporting process by giving erroneous results to many of the performance measurements.