novática
Revista de la Asociación de Técnicos de Informática

Novática, founded in 1975, is the oldest periodical publication amongst those especialized in Information and Communication Technology (ICT) existing today in Spain. It is published by **ATI** (***Asociación de Técnicos de Informática***) which also publishes **REICIS** (*Revista Española de Inovación, Calidad e Ingeniería del Software*).

<http://www.ati.es/novatica/>
<http://www.ati.es/reicis/>

ATI is a founding member of **CEPIS** (*Council of European Professional Informatics Societies*), the Spain´s representative in **IFIP** (*International Federation for Information Processing*), and a member of **CLEI** (Centro Latinoamericano de Estudios en Informática) and **CECUA** (*Confederation of EuropeanComputer User Associations*). It has a collaboration agreement with **ACM** (*Association for Computing Machinery*) as well as with diverse Spanish organisations in the ICT field.

The opinions expressed by the autors are their exclusive responsability

Editorial Office, Advertising and Madrid Office
Plaza de España 6, 2ª planta, 28008 Madrid
Tlfn.914029391; fax.913093685 <novatica@ati.es>
Layout and Comunidad Valenciana Office
Av. del Reino de Valencia 23, 46005 Valencia
Tlfn. 963740173 <novatica_prod@ati.es>
Acounting, Subscriptions and Catalonia Office
Calle Ävila 48-50, 3a planta, local 9, 08005 Barcelona
Tlfn.934125235; fax 934127713 <secregen@ati.es>
Andalucía Office <secreand@ati.es>
Galicia Office<secregal@ati.es>
Subscriptións and Sales <novatica.subscripciones@atinet.es>
Advertising Plaza de España 6, 2ª planta, 28008 Madrid
Tlnf.914029391; fax.913093685 <novatica@ati.es>
Legal depósit: B 15.154-1975 -- ISSN: 0211-2124; CODEN NOVAEC
Cover Page: "Mineral, Vegetable, Animal" - Concha Arias Pérez / © ATI
Layout Dising: Fernando Agresta / © ATI 2003

Special English Edition 2013-2014
Annual Selection of Articles

summary

editorial

From the Chief Editor´Pen

monograph

Process Mining

Guest Editors: Antonio Valle-Salas and Anne Rozinat

Antonio Valle-Salas[1], Anne Rozinat[2]

Managing Partner of G2; Co-founder of Fluxicon

<avalle@gedos.es>, <anne@fluxicon.com>

Presentation. Introduction to Process Mining

During the last few decades information technology has touched every part of our lives. From cellular phones to the most advanced medical information processing systems, vending machines and PLCs in production lines, computerized components are everywhere. All these components generate vast amounts of information that are growing exponentially, Relatively few years ago the challenge was finding digitized information, whereas now the problem is being able to process and give meaning to all the information we generate.

In recent years we have seen how the information analysis industry has proposed various approaches to this problem. Some of them have been covered in one way or another in previous editions of ***Novática***: starting with *VLDB* (Very Large Databases) in volume 91 (1991) and *Datawarehouse* approaches attempting to discover patterns in these data stores with *Data Mining* in volume 138 (1999), then followed *Knowledge Management* in volume 155 (2002). We realized how complex the problem was in the monograph on *The Internet of Things* in volume 209 (2011) and how we could exploit this information in volume 211 on *Business Intelligence* (2011). Finally, the industry is also moving in a direction not yet covered in ***Novática*** but certain to be addressed in the near future: *Big Data.*

In the present volume of ***Novática*** we address a particularly interesting topic within this broad range of techniques for data analysis: *Process Mining.* This is a variant of data mining in which we focus on analyzing the information generated by the processes that have been computerized and whose executions have been traced. As ***Anne Rozinat*** and ***Wil van der Aalst*** explain in the opening article, we will see that the first traces are found in the late nineteenth century, although in terms of modern science we refer to the seminal work of Myhill / Nerod (1958), or Viterbi algorithms (1978).

In the late 90s there were already some specific research teams in universities around the world, especially the University of Colorado and *Technische Universiteit Eindhoven.* These teams developed their research defining algorithms and methods that allow the treatment of process execution traces for discovery, analysis and representation of the underlying processes.

Guest Editors

Antonio Valle-Salas is Managing Partner of G2 and a specialist consultant in ITSM (Information Technology Service Management) and IT Governance. He graduated as a Technical Engineer in Management Informatics from UPC (*Universitat Politécnica de Catalunya*) and holds a number of methodology certifications such as ITIL Service Manager from EXIN (Examination Institute for Information Science), Certified Information Systems Auditor (CISA) from ISACA, and COBIT Based IT Governance Foundations from IT Governance Network, plus more technical certifications in the HP Openview family of management tools. He is a regular collaborator with itSMF (IT Service Management Forum) Spain and its Catalan chapter, and combines consulting and project implementation activities with frequent collaborations in educational activities in a university setting (such as UPC or the *Universitat Pompeu Fabra*) and in the world of publishing in which he has collaborated on such publications as IT Governance: a Pocket Guide, Metrics in IT Service Organizations, *Gestión de Servicios TI. Una introducción a ITIL*, and the translations into Spanish of the books ITIL V2 Service Support and ITIL V2 Service Delivery.

Anne Rozinat has more than ten years of experience with process mining technology and obtained her PhD cum laude in the process mining group of Prof. Wil van der Aalst at the Eindhoven University of Technology in the Netherlands. Currently, she is a co-founder of Fluxicon and blogs at <http://www.fluxicon.com/blog/>.

However no tools that implemented these algorithms with appropriate degrees of usability had yet reached the market. By the end of 2003 the *processmining.org* specialized community (a working group of the TU/e) was created, and in early 2004 the first version of ProM was developed, a generic and open source framework for process mining that has become the primary tool for researchers and analysts, now at version 6.3 and including more than 500 plugins that implement state of the art in this field.

In 2009 a Task Force of the IEEE focused on process mining was created that now has members from over 20 countries including software vendors (such as Software AG, HP, IBM or Fluxicon, among many others), consulting firms and analysts (Process Sphere, Gartner and Deloitte, among others) and a wide range of educational and research institutions (TU/e, Universitat Politècnica de Catalunya or Universität zu Berlin, to name but a few). One of the key objectives of this task force is to spread the concepts, techniques and benefits of process mining. In 2011 they published the Process Mining Manifesto, a document signed by more than 50 professionals translated into 12 languages. We cannot include here the full text of the manifesto, but the reader will find the reference in the links section of this monograph.

For this present edition of ***Novática*** we have been privileged to have a group of authors that give us different perspectives on the matter. We begin with an introductory article in which ***Anne Rozinat*** and ***Wil van der Aalst*** set the context for process mining concepts and state, in a very enlightening process mining message, that it allows us to get an objective vision of our processes.

In the second paper ***Wil van der Aalst*** guides us through the different uses we can make of process mining: to create of a model of the process, to check the compliance of the model or to improve an existing model. Here another key message is presented: the use of process mining as X-rays that allow us to see "inside" the process, based on the analysis of real data from the execution of all cases (as opposed to the statistical sampling we would do in an audit, for example).

In the next article you will find the ***Josep Carmona***'s vision of the task of discovering a process from its traces. Here Josep makes an entertaining approach to how we could use the mining process to decrypt the message of an alien explaining his visit to planet Earth, while showing us the anatomy of the discovery process.

The introductory papers will be followed by a set of articles focusing on case studies. First ***Antonio Valle-Salas'*** article presents an application of process mining in a specific industry, focusing on the processes in an IT

Department and showing the different uses we can make of these techniques in the world of IT Service Management (ITSM)

Then ***Arjel D. Bautista***, ***Lalit M. Wangikar*** and ***Syed Kumail Akbar*** present the work done to optimize the loan approval process of a Dutch bank, a remarkable work that was awarded the BPI Challenge 2012 prize.

Finally ***Daniela Luengo*** and ***Marcos Sepúlveda*** give us a research perspective on one of the challenges stated in the manifesto: dealing with the concept drift. This term is used to refer to the situation in which the process is changing while it is being used. Detecting these changes and including these features in the analysis is essential when working on rapidly changing environments because, otherwise, it could lead to erroneous conclusions in analysis.

These authors have contributed with their articles to give a clearer vision of what process mining is, what it is useful for and what its future is. Process mining is a relatively new science but is already reaching the level of maturity required to become standard practice in companies and organizations, as reflected in the articles' practical uses. However, there are still many challenges ahead and a long way to go: Will we be able to overcome the problems introduced by the concept drift? Can we use process mining not only for knowing the past of a process but also to predict its future? Will we implement these techniques in the management systems of business processes in order to provide them with predictive systems or support operators?

We are sure we will see great advances in this area in the near future.

Useful References of "Process Mining"

In addition to the materials referenced by the authors in their articles, we offer the following ones for those who wish to dig deeper into the topics covered by the monograph:

- **W.M.P. van der Aalst.** *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer Verlag, 2011. ISBN 978-3-642-19344-6.
- **IEEE Task Force on Process Mining.** *Process Mining Manifesto* (en 12 idiomas). <http://www.win.tue.nl/ieeetfpm/doku.php?id=shared: process_mining_manifesto>.
- **Fluxicon TU/eProcess Mining Group.** *Introduction to Process Mining: turning (big) data into value* (video). <http://www.youtube.com/watch?v=7oat7MatU_U>.
- **Fluxicon.** *Process Mining News.* <http://fluxicon.com/s/newsarchive>.
- **TU/e Workgroup.** <http://www.processmining.org>.
- **Fluxicon.** *Process Mining Blog.* <http://fluxicon.com/blog/>.
- **IEEE Task Force on Process Mining.** <http://www.win.tue.nl/ieeetfpm/doku.php?id=start>.
- **LinkedIn.** *Process Mining* (community) <http://www.linkedin.com/groups/Process-Mining-1915049>.
- **TU/e.** *Health Analytics Using Process Mining.* <http://www.healthcare-analytics-process-mining.org>.

Anne Rozinat[1], Wil van der Aalst[2]

[1]Co-founder of Fluxicon, The Netherlands; [2]Technical University of Eindhoven, The Netherlands

<anne@fluxicon.com>,
<w.m.p.v.d.aalst@tue.nl>

Process Mining: The Objectification of Gut Instinct - Making Business Processes More Transparent Through Data Analysis

Abstract: *Big Data existed in the 19th Century. At least that might be the conclusion you would draw by reading the story of Matthew Maury. We draw a parallel with the first systematic evaluations of seafaring logbooks and we show how you can quickly and objectively map processes based on the evaluation of log files in IT systems.*

Keywords: *Big Data, Case Study, Log Data, Process Mining, Process Models, Process Visualization, Systematic Analysis.*

Authors

Anne Rozinat has more than eight years of experience with process mining technology and obtained her PhD cum laude in the process mining group of Prof. Wil van der Aalst at the Eindhoven University of Technology in the Netherlands. Currently, she is a co-founder of Fluxicon and blogs at <http://www.fluxicon.com/blog/>.

Wil van der Aalst is a professor at the Technical University in Eindhoven and with an H-index of over 90 points the most cited computer scientist in Europe. Well known through his work on the Workflow Patterns, he is the widely recognized "godfather" of process mining. His personal website is <http://www.vdaalst.com>.

1. Introduction

The archive of the United States Naval Observatory stored all the naval logbooks of the US Navy in the 19th century. These logbooks contained daily entries relating to position, winds, currents and other details of thousands of sea voyages. These logbooks lay ignored and it had even been suggested that they be thrown away until Mathew Fontaine Maury came along.

Maury (see **Figure 1**) was a sailor in the US Navy and from 1842 was the director of the United States Naval Observatory. He evaluated the data systematically and created illustrated handbooks which visually mapped the winds and currents of the oceans. These were able to serve ships' captains as a decision-making aid when they were planning their route.

In 1848 Captain Jackson of the W. H. D. C. Wright was one of the first users of Maury's handbooks on a trip from Baltimore to Rio de Janeiro and returned more than a month earlier than planned. After only seven years from the production of the first edition Maury's Sailing Directions had saved the sailing industry worldwide about 10 million dollars per year [1].

The IT systems in businesses also conceal invaluable data, which often remains completely unused. Business processes create the modern day equivalent of "logbook entries", which detail exactly which activities were carried out when and by whom, (see **Figure 2**). If, for example, a purchasing process is started in an SAP system, every step in the process is indicated in the corresponding SAP tables. Similarly, CRM systems, ticketing systems and even legacy systems record historical data about the processes.

These digital traces are the byproduct of the increasing automation and IT support of business processes [2].

2. From Random Samples to Comprehensive Analysis

Before Maury's manual on currents and tides, sailors were restricted to planning a route based solely on their own experience. This is also the case for most business processes: Nobody really has a clear overview of how the processes are actually executed. Instead, there are anecdotes, good feeling and many subjective (potentially contradicting) opinions which have to be reconciled.

The systematic analysis of digital log traces through so-called Process Mining techniques [3] offers enormous potential for all

Figure 1. Matthew Fontaine Maury (Source: Wikipedia).

> “The manual discovery through classical workshops and interviews is costly and time-consuming, remaining incomplete and subjective ”

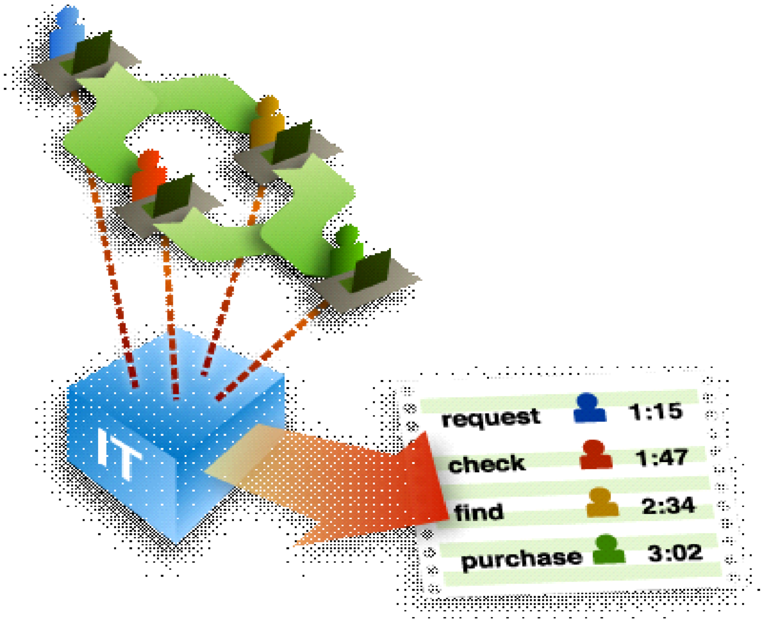


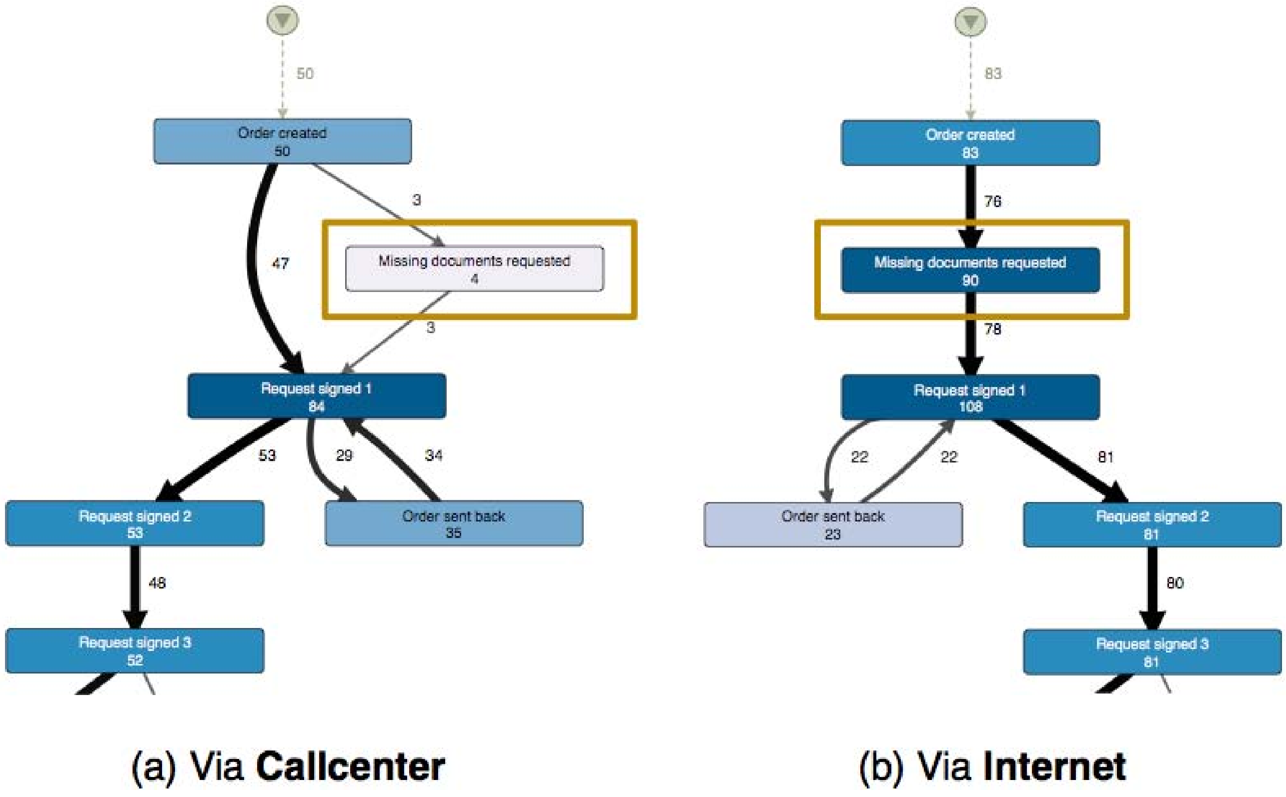
Figure 2. IT-supported Processes Record in Detail Which Activities Were Executed When and by Whom.

organizations currently struggling with complex processes. Through an analysis of the sequence of events and their time stamps, the actual processes can be fully and objectively reconstructed and weaknesses uncovered. The information in the IT logs can be used to automatically generate process models, which can then be further enriched by process metrics also extracted directly out of the log data (for example execution times and waiting times).

Typical questions that Process Mining can answer are:

- What does my process actually look like?
- Where are the bottlenecks?
- Are there deviations from the prescribed or described process?

In order to optimize a process, one must first understand the current process reality - the ‘As-is’ process. This is usually far from simple, because business processes are



(a) Via **Callcenter**

(b) Via **Internet**

Figure 3. Process Visualization of the Refund Process for Cases Were Started Via the Call Center (a) and Via the Internet Portal (b). In the case of the internet cases missing information has to be requested too often. In the call center-initiated process, however, the problem does not exist.

> “Like Maury did with the naval log books, objective process maps can be derived that show how processes actually work in the real world ”

performed by multiple persons, often distributed across different organizational units or even companies.

Everybody only sees a part of the process. The manual discovery through classical workshops and interviews is costly and time-consuming, remaining incomplete and subjective. With Process Mining tools it is possible to leverage existing IT data from operational systems to quickly and objectively visualize the As-Is processes as they are really taking place.

In workshops with process stakeholders one can then focus on the root cause analysis and the value-adding process improvement activities.

3. A Case Study

In one of our projects we have analyzed a refund process of a big electronics manufacturer. The following process description has been slightly changed to protect the identity of the manufacturer. The starting point for the project was the feeling of the process manager that the process had severe problems. Customer complaints and the inspection of individual cases indicated that there were lengthy throughput times and other inefficiencies in the process.

The project was performed in the phases: First, the concrete questions and problems were collected, and the IT logs of all cases from the running business year were extracted from the corresponding service platform. Then, in an interactive workshop involving the process managers the log data were analyzed.

For example, in **Figure 3** you see a simplified fragment of the beginning of the refund process. On the left side (a) is the process for all cases that were initiated via the call center. On the right side (b) you see the same process fragment for all cases that were initiated through the internet portal of the manufacturer. Both process visualizations were automatically constructed using Fluxicon's process mining software Disco based on the IT log data that had been extracted.

The numbers, the thickness of the arcs, and the coloring all illustrate how frequently each activity or path has been performed. For example, the visualization of the call center-initiated process is based on 50 cases (see left in **Figure 3**). All 50 cases start with activity Order created. Afterwards, the request is immediately approved in 47 cases. In 3 cases missing information has to be requested from the customer. For simplicity, only the main process flows are displayed here.

What becomes apparent in **Figure 3** is that, although missing information should only occasionally be requested from the customer,

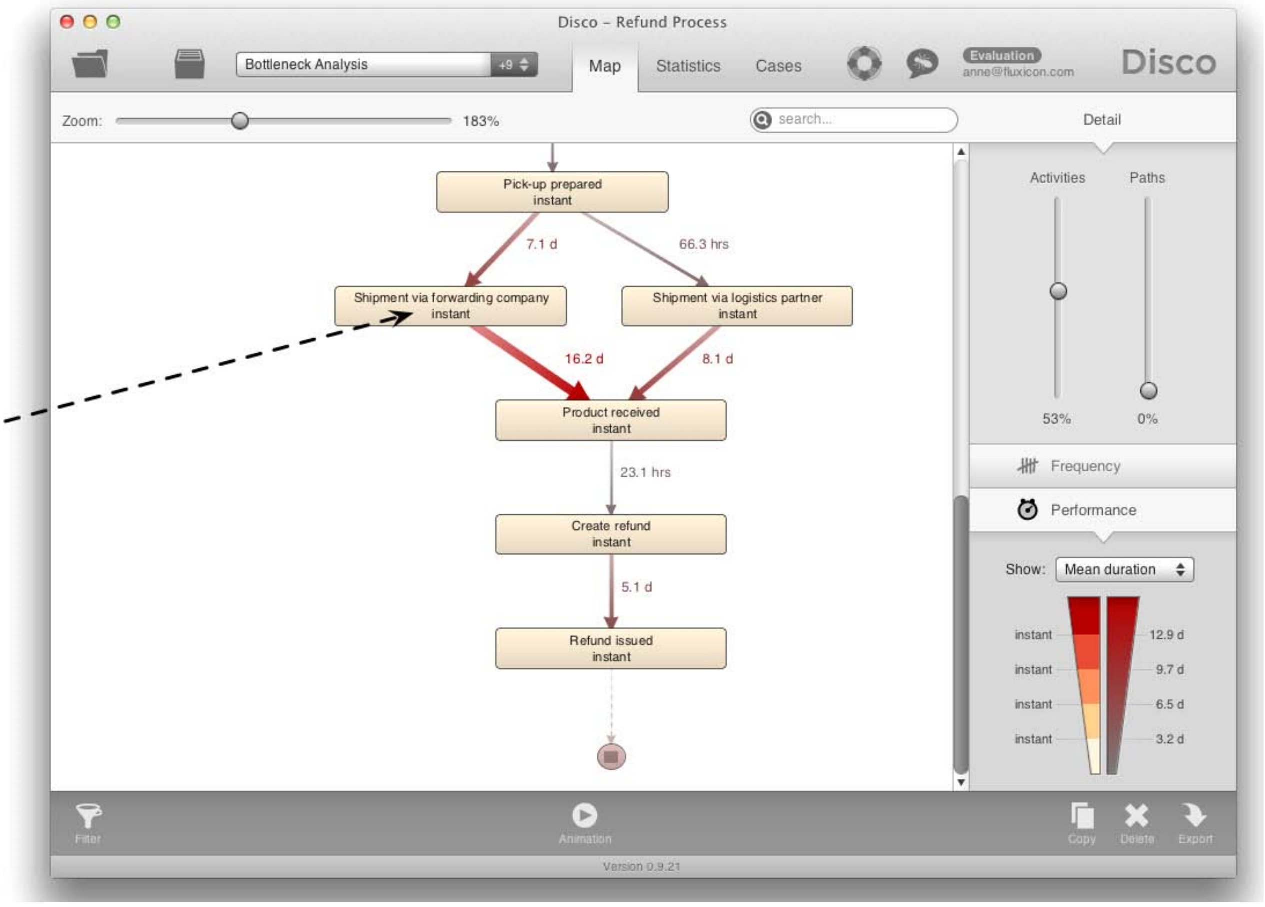


Figure 4. Screenshot of the Process Mining Software Disco in the Performance Analysis View. It becomes apparent that the shipment through the forwarding company causes a bottleneck.

this happens a lot for cases that are started via the internet portal: For 97% of all cases (77 out of 83 completed cases) this additional process step was performed. For 12 of the 83 analyzed cases (ca. 14%) this happened even multiple times (in total 90 times for 83 cases).

This process step costs a lot of time because it requires a call or an email from the service provider. In addition, through the external communication that is required, the process is delayed for the customer, who in a refund process has already had a bad experience. Therefore, the problem needs to be solved. An improvement of the internet portal (with respect to the mandatory information in the form that submits the refund request) could ensure that information is complete when the process is started.

Another analysis result was a detected bottleneck in connection with the pick-ups that were performed through the forwarding company. The process fragment in **Figure 4** shows the average waiting times between the process steps based on the timestamps in the historical data.

Also such waiting times analyses are automatically created by the process mining software. You can see that before and after the process step Shipment via forwarding company a lot of time passes. For example, it takes on average ca. 16 days between Shipment via forwarding company and Product received. The company discovered that the root cause for the long waiting times was that products were collected in a palette and the palette was shipped only when it was full, which led to delays particularly for those products that were placed in an almost empty palette. Also the actual refund process at the electronics manufacturer was taking too long (on average ca. 5 days). For the customer the process is only completed when she has her money back.

As a last result of the process mining analysis, deviations from the required process were detected. It is possible to compare the log data (and therewith the actual process) objectively and completely against required business rules, and to isolate those cases that show deviations. Specifically, we found that (1) in one case the customer received the refund twice, (2) in two cases the money was refunded without ensuring that the defect product had been received by the manufacturer, (3) in a few cases an important and mandatory approval step in the process had been skipped.

4. State of the Art

Process mining, which is still a young and relatively unknown discipline, is being made available by the first professional software tools on the market and supported by published case studies [4 |. The IEEE Task Force on Process Mining [5] was founded in 2009 to increase the visibility of process mining. In autumn 2011, it published a Process Mining Manifesto [6], which is available in 13 languages.

Companies already generate vast quantities of data as a byproduct of their IT-enabled business processes. This data can be directly analyzed by process mining tools. Like Maury did with the naval log books, objective process maps can be derived that show how processes actually work in the real world [7]. Developments in the field of Big Data are helping to store and access this data to analyze it effectively.

Matthew Fontaine Maury's wind and current books were so useful that by the mid-1850s, their use was even made compulsory by insurers [8] in order to prevent marine accidents and to guarantee plain sailing. Likewise, in Business process analysis and optimization, there will come a point when we can not imagine a time when we were ever without it and left to rely on our gut feeling.

References

[1] **Tim Zimmermann.** *The Race: Extreme Sailing and Its Ultimate Event: Nonstop, Round-the-World, No Holds Barred*. Mariner Books, 2004. ISBN-10: 0618382704.

[2] **W. Brian Arthur.** *The Second Economy*. McKinsey Quarterly, 2011.

[3] **Wil M.P. van der Aalst.** Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag, 2011. ISBN-10: 3642193447.

[4] **Alberto Manuel.** *Process Mining - Ana Aeroportos de Portugal*, 2012. BPTrends, <www.bptrends.com>.

[5] **IEEE Task Force on Process Mining.** <http://www.win.tue.nl/ieeetfpm/>.

[6] **IEEE Task Force on Process Mining.** Process Mining Manifesto. Business Process Management Workshops 2011, *Lecture Notes in Business Information Processing, Vol. 99*, Springer-Verlag, 2011.

[7] **Anne Rozinat.** *How to Reduce Waste With Process Mining*, 2011. BPTrends, <www.bptrends.com>.

[8] **Mark A. Thornton.** *General Circulation and the Southern Hemisphere*, 2005. <http://www.lakeeriewx.com/Meteo241/ResearchTopicTwo/ProjectTwo.html>.

Wil van der Aalst
Technical University of Eindhoven, The Netherlands

<w.m.p.v.d.aalst@tue.nl>

Process Mining: X-Ray Your Business Processes

Abstract: *Recent breakthroughs in* process mining *research make it possible to discover, analyze, and improve business processes based on event data. Activities executed by people, machines, and software leave trails in so-called* event logs. *Events such as entering a customer order into SAP, checking in for a flight, changing the dosage for a patient, and rejecting a building permit have in common that they are all recorded by information systems. Over the last decade there has been a spectacular growth of data. Moreover, the digital universe and the physical universe are becoming more and more aligned. Therefore, business processes should be managed, supported, and improved based on event data rather than subjective opinions or obsolete experiences. The application of process mining in hundreds of organizations has shown that both managers and users tend to overestimate their knowledge of the processes they are involved in. Hence, process mining results can be viewed as X-rays showing what is* really *going on* inside *processes. Such X-rays can be used to diagnose problems and suggest proper treatment. The practical relevance of process mining and the interesting scientific challenges make process mining one of the "hot" topics in Business Process Management (BPM). This article provides an introduction to process mining by explaining the core concepts and discussing various applications of this emerging technology.*

Keywords: *Business Intelligence, Business Process Management, Data Mining, Management, Measurement, Performance, Process Mining,*

Author

Wil van der Aalst is a professor at the Technical University in Eindhoven and with an H-index of over 90 points the most cited computer scientist in Europe. Well known through his work on the Workflow Patterns, he is the widely recognized "godfather" of process mining. His personal website is <http://www.vdaalst.com>.

1. Process Mining Spectrum

Process mining aims to *discover, monitor and improve real processes by extracting knowledge from event logs* readily available in today's information systems [1][2].

Although event data are omnipresent, organizations lack a good understanding of their actual processes. Management decisions tend to be based on PowerPoint diagrams, local politics, or management dashboards rather than an careful analysis of event data. The knowledge hidden in event logs cannot be turned into actionable information. Advances in data mining made it possible to find valuable patterns in large datasets and to support complex decisions based on such data. However, classical data mining problems such as classification, clustering, regression, association rule learning, and sequence/episode mining are *not* process-centric.

Therefore, Business Process Management (BPM) approaches tend to resort to hand-made models. Process mining research aims to bridge the gap between data mining and BPM. Metaphorically, process mining can be seen as taking X-rays to diagnose/predict problems and recommend treatment.

An important driver for process mining is the incredible growth of event data [4][6]. Event data is everywhere – in every sector, in every economy, in every organization, and in every home one can find systems that log events. For less than $600, one can buy a disk drive with the capacity to store all of the world's music [6]. A recent study published in *Science*, shows that storage space grew from 2.6 optimally compressed exabytes (2.6 x 10^{18} bytes) in 1986 to 295 compressed exabytes in 2007. In 2007, 94 percent of all information storage capacity on Earth was digital. The other 6 percent resided in books, magazines and other non-digital formats. This is in stark contrast with 1986 when only 0.8 percent of all information storage capacity was digital. These numbers illustrate the exponential growth of data.

The further adoption of technologies such as RFID (Radio Frequency Identification), location-based services, cloud computing, and sensor networks, will further accelerate the growth of event data. However, organizations have problems effectively using such large amounts of event data. In fact, most organizations still diagnose problems based on fiction (Powerpoint slides, Visio diagrams, etc.) rather than facts (event data). This is illustrated by the poor quality of process models in practice, e.g., more than 20% of the 604 process diagrams in SAP's reference model have obvious errors and their relation to the actual business processes supported by SAP is unclear [7]. Therefore, it is vital to turn the massive amounts of event data into relevant knowledge and reliable insights. This is where process mining can help.

The growing maturity of process mining is illustrated by the *Process Mining Manifesto* [5] recently released by the *IEEE Task Force on Process Mining.* This manifesto is supported by 53 organizations and 77 process mining experts contributed to it. The active contributions from end-users, tool vendors, consultants, analysts, and researchers illustrate the significance of process mining as a bridge between data mining and business process modeling.

Starting point for process mining is an *event log.* Each event in such a log refers to an *activity* (i.e., a well-defined step in some process) and is related to a particular *case* (i.e., a *process instance*). The events belonging to a case are *ordered* and can be seen as one "run" of the process. Event logs may store additional information about events. In fact, whenever possible, process mining techniques use extra information such as the *resource* (i.e., person or device) executing or initiating the activity, the *timestamp* of the event, or *data elements* recorded with the event (e.g., the size of an order).

Event logs can be used to conduct three types of process mining as shown in **Figure** 1 [1]. The first type of process mining is *discovery.* A discovery technique takes an event log and produces a model without using any a-priori information. Process discovery is the most prominent process

> “ Conformance checking can be used to check if reality, as recorded in the log, conforms to the model and vice versa ”

mining technique. For many organizations it is surprising to see that existing techniques are indeed able to discover real processes merely based on example behaviors recorded in event logs.

The second type of process mining is *conformance*. Here, an existing process model is compared with an event log of the same process. Conformance checking can be used to check if reality, as recorded in the log, conforms to the model and vice versa. The third type of process mining is *enhancement*. Here, the idea is to extend or improve an existing process model using information about the actual process recorded in some event log. Whereas conformance checking measures the alignment between model and reality, this third type of process mining aims at changing or extending the a-priori model. For instance, by using timestamps in the event log one can extend the model to show bottlenecks, service levels, throughput times, and frequencies.

2. Process Discovery

As shown in **Figure** 1, the goal of process discovery is to learn a model based on some event log. Events can have all kinds of attributes (timestamps, transactional information, resource usage, etc.). These can all be used for process discovery. However, for simplicity, we often represent events by activity names only. This way, a case (i.e., process instance) can be represented by a *trace* describing a sequence of activities.

Consider for example the event log shown in **Figure** 1 (example is taken from [1]). This event log contains 1,391 cases, i.e., instances of some reimbursement process. There are 455 process instances following trace *acdeh*. Activities are represented by a single character: *α = register request*, *b = examine thoroughly*, *c = examine casually*, d = *check ticket*, *e = decide*, *f = reinitiate request*, *g = pay compensation*, and *h = reject request*. Hence, trace *acdeh* models a reimbursement request that was rejected after a registration, examination, check, and decision step. 455 cases followed this path consisting of five steps, i.e., the first line in the table corresponds to 455 x 5 = 2,275 events. The whole log consists of 7,539 events.

Process discovery techniques produce process models based on event logs such as the one shown in **Figure** 2. For example, the classical *α*-algorithm produces model M_1 for this log. This process model is represented as a *Petri net*. A Petri net consists of *places* and *transitions*. The state of a Petri net, also referred to as *marking*, is defined by the distribution of *tokens* over places.

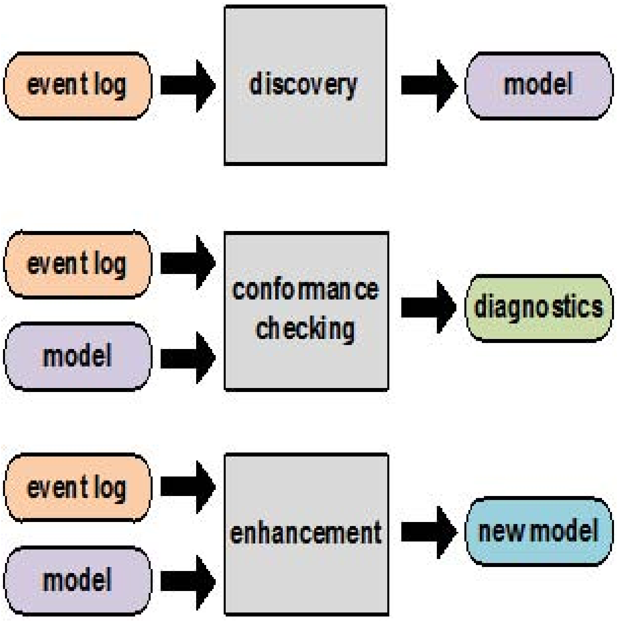


Figure 1. The Three Basic Types of Process Mining Explained in Terms of Input and Output.

A transition is *enabled* if each of its input places contains a token. For example, a is enabled in the initial marking of M_1, because the only input place of a contains a token (black dot). Transition e in M_1 is only enabled if both input places contain a token. An enabled transition may *fire* thereby consuming a token from each of its input places and producing a token for each of its output places. Firing a in the initial marking corresponds to removing one token from *start* and producing two tokens (one for each output place). After firing a, three transitions are enabled: b, c, and d. Firing b will disable c because the token is removed from the shared input place (and vice versa). Transition d is concurrent with b and c, i.e., it can fire without disabling another transition. Transition e becomes enabled after d and b or c have occurred. After executing e three transitions become enabled: f, g, and h. These transitions are competing for the same token thus modeling a choice. When g or h is fired, the process ends with a token in place *end*. If f is fired, the process returns to the state just after executing a.

Note that transition d is concurrent with b and c. Process mining techniques need to be able to discover such more advanced process patterns and should not be restricted to simple sequential processes.

It is easy to check that all traces in the event log can be reproduced by M_1. This does not hold for the second process model in **Figure** 2. M_2 is only able to reproduce the most frequent trace $acdeh$. The model does not *fit* the log well because observed traces such as $abdeg$ are not possible according to M_2. The third model is able to reproduce the entire event log, but M_3 also allows for traces such as ah and $adddddddg$.

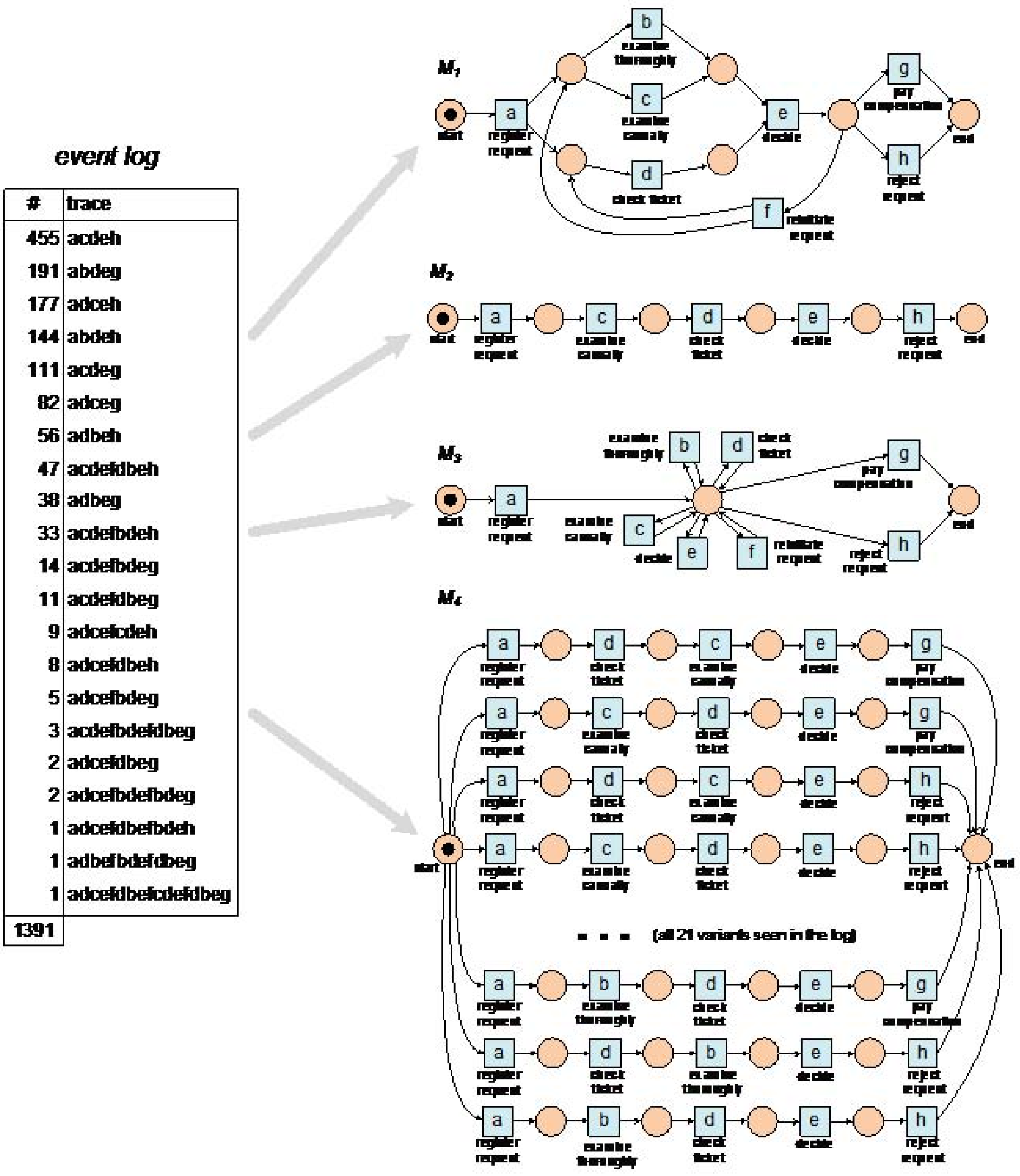
Therefore, we consider M_3 to be "underfitting"; too much behavior is allowed because M_3 clearly overgeneralizes the observed behavior. Model M_4 is also able to reproduce the event log. However, the model simply encodes the example traces in the log. We call such a model "overfitting" as the model does not generalize behavior beyond the observed examples.

In recent years, powerful process mining techniques have been developed that can automatically construct a suitable process model given an event log. The goal of such techniques is to construct a simple model that is able to explain most of the observed behavior without "overfitting" or "underfitting" the log.

3. Conformance Checking

Process mining is not limited to process discovery. In fact, the discovered process is merely the starting point for deeper analysis. As shown in Figure 1, conformance checking and enhancement relate model and log. The model may have been made by hand or discovered through process discovery. For conformance checking, the modeled behavior and the observed behavior (i.e., event log) are compared. When checking the conformance of M_2 with respect to the log shown in Figure 2 it is easy to see that only the 455 cases that followed $acdeh$ can be replayed from begin to end. If we try to replay trace $acdeg$, we get stuck after executing $acde$ because g is not enabled. If we try to replay trace $adceh$, we get stuck after executing the first step because d is not (yet) enabled.

There are various approaches to diagnose and quantify conformance. One approach is to find an *optimal alignment* between each trace in the log and the most similar behavior in the model. Consider for example process model M_1, a fitting trace $\sigma_1 = adceg$, a non-fitting trace $\sigma_2 = abefdeg$, and the three alignments shown in Table 1. γ_1 shows a perfect alignment between σ_1 and M_1: all moves of the trace in the event log (top part of alignment) can be followed by



#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Figure 2. One Event Log and Four Potential Process Models (M1, M2, M3 and M4) Aiming to Describe the Observed Behavior.

> A Petri net consists of *places* and *transitions*. The state of a Petri net, also referred to as *marking*, is defined by the distribution of *tokens* over places

$$\gamma_1 = \begin{array}{|c|c|c|c|c|} a & d & c & e & g \\ \hline a & d & c & e & g \end{array} \quad \text{and} \quad \gamma_2 = \begin{array}{|c|c|c|c|c|c|c|c|c|} a & b & \gg & e & f & d & \gg & e & g \\ \hline a & b & d & e & f & d & b & e & g \end{array} \quad \text{and} \quad \gamma_3 = \begin{array}{|c|c|c|c|c|c|c|} a & b & e & f & d & e & g \\ \hline a & b & \gg & \gg & d & e & g \end{array}$$

Table 1. Examples of Alignment Between the Traces in the Event Log and the Model.

moves of the model (bottom part of alignment). γ_2 shows an optimal alignment for trace σ_2 in the event log and model M_1.

The first two moves of the trace in the event log can be followed by the model. However, e is not enabled after executing just a and b. In the third position of alignment γ_2, we see a d move of the model that is not synchronized with a move in the event log. A move in just the model is denoted as $(\gg, d)$. In the next three moves model and log agree. In the seventh position of alignment γ_2 there is just a move of the model and not a move in the log: $(\gg, b)$. γ_3 shows another optimal alignment for trace σ_2. Here there are two situations where log and model do not move together: $(e, \gg)$ and $(f, \gg)$. Alignments γ_2 and γ_3 are both optimal if the penalties for "move in log" and "move in model" are the same. In both alignments there are two $\gg$ steps and there are no alignments with less than two $\gg$ steps.

Conformance can be viewed from two angles: (a) the model does not capture the real behavior ("the model is wrong") and (b) reality deviates from the desired model "the event log is wrong"). The first viewpoint is taken when the model is supposed to be *descriptive*, i.e., capture or predict reality. The second viewpoint is taken when the model is *normative*, i.e., used to influence or control reality.

There are various types of conformance and creating an alignment between log and model is just the starting point for conformance checking [1]. For example, there are various *fitness* (the ability to replay) metrics. A model has fitness 1 if all traces can be replayed from begin to end. A model has fitness 0 if model and event log "disagree" on all events. Process models M_1, M_3 and M_4 have a fitness of 1 (i.e., perfect fitness) with respect to the event log shown in **Figure 2** Model M_2 has a fitness 0.8 for the event log consisting of 1,391 cases. Intuitively, this means that 80% of the events in the log can be explained by the model. Fitness is just one of several conformance metrics.

Experiences with conformance checking in dozens of organizations show that real-life processes often deviate from the simplified Visio or PowerPoint representations used by process analysts.

4. Model Enhancement

It is also possible to extend or improve an existing process model using the alignment between event log and model. A non-fitting process model can be corrected using the diagnostics provided by the alignment. If the alignment contains many $(e, \gg)$ moves, then it may make sense to allow for the skipping of activity e in the model. Moreover, event logs may contain information about resources, timestamps, and case data. For example, an event referring to activity "register request" and case "992564" may also have attributes describing the person that registered the request (e.g., "John"), the time of the event (e.g., "30-11-2011:14.55"), the age of the customer (e.g., "45"), and the claimed amount (e.g., "650 euro"). After aligning model and log it is possible to replay the event log on the model. While replaying one can analyze these additional attributes.

For example, as **Figure 3** shows, it is possible to analyze waiting times in-between activities. Simply measure the time difference between causally related events and compute basic statistics such as averages, variances, and confidence intervals. This way it is possible to identify the main bottlenecks. Information about resources can be used to discover roles, i.e., groups of people frequently executing related activities. Here, standard clustering techniques can be used. It is also possible to construct social networks based on the flow of work and analyze resource performance (e.g., the relation between workload and service times).

Standard classification techniques can be used to analyze the decision points in the process model. For example, activity e ("decide") has three possible outcomes ("pay", "reject", and "redo"). Using the data known about the case prior to the decision, we can construct a decision tree explaining the observed behavior.

Figure 3 illustrates that process mining is not limited to control-flow discovery. Moreover, process mining is not restricted to offline analysis and can also be used for predictions and recommendations at runtime. For example, the completion time of a partially handled customer order can be predicted using a discovered process model with timing information.

5. Process Mining Creates Value in Several Ways

After introducing the three types of process mining using a small example, we now focus on the practical value of process mining. As mentioned earlier, process mining is driven by the exponential growth of event data. For example, according to MGI, enterprises stored more than 7 exabytes of new data on disk drives in 2010 while consumers stored more than 6 exabytes of new data on devices such as PCs and notebooks [6].

In the remainder, we will show that process mining can provide value in several ways. To illustrate this we refer to case studies where we used our open-source software package *ProM* [1]. ProM was created and is maintained by the process mining group at Eindhoven University of Technology. However, research groups from all over the world contributed to it, e.g., University of Padua, Universitat Politècnica de Catalunya, University of Calabria, Humboldt-Universität zu Berlin, Queensland University of Technology, Technical University of Lisbon, Vienna University of Economics and Business, Ulsan National Institute of Science and Technology, K.U. Leuven, Tsinghua University, and University of Innsbruck. Besides ProM there are about 10 commercial software vendors providing process mining software (often embedded in larger tools), e.g., Pallas Athena, Software AG, Futura Process Intelligence, Fluxicon, Businesscape, Iontas/Verint, Fujitsu, and Stereologic.

> “It is also possible to extend or improve an existing process model using the alignment between event log and model”

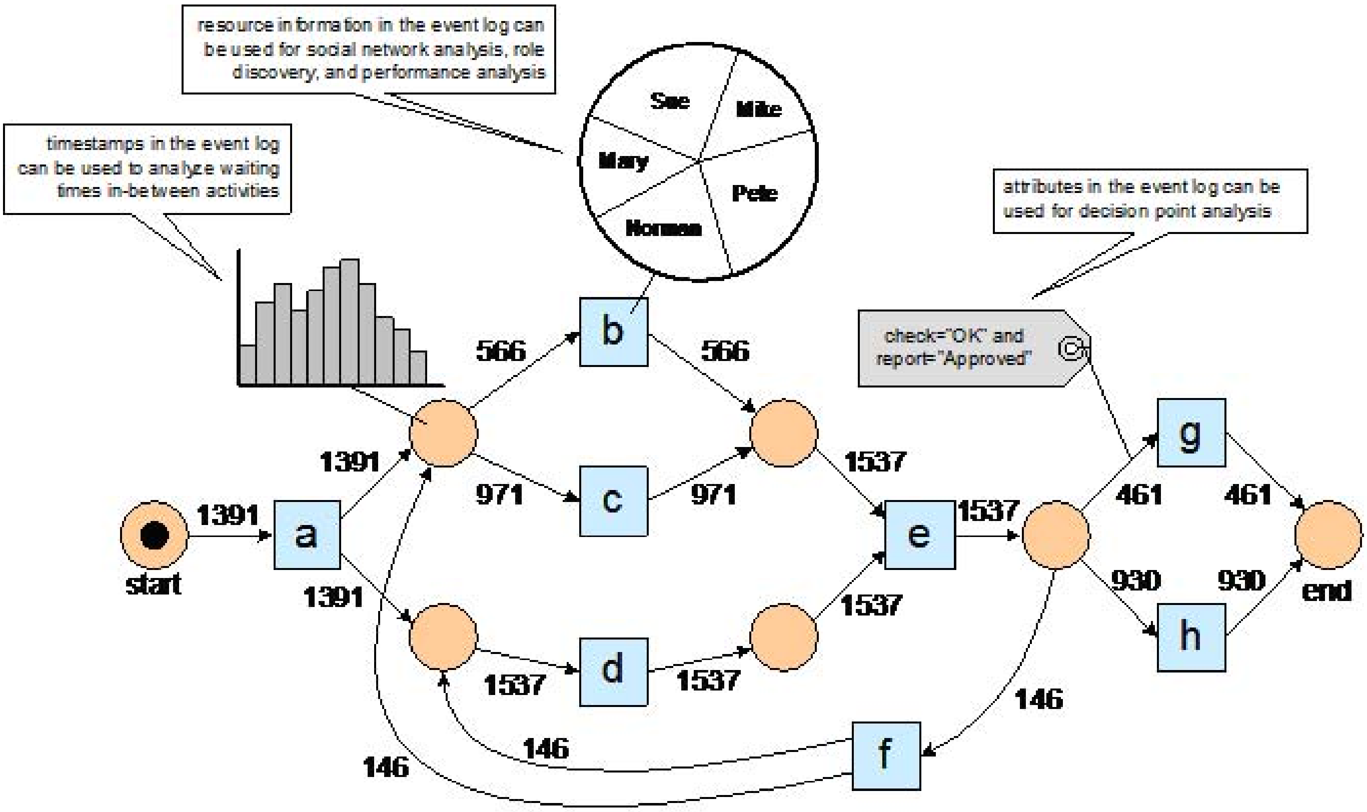

resource information in the event log can be used for social network analysis, role discovery, and performance analysis
timestamps in the event log can be used to analyze waiting times in-between activities
attributes in the event log can be used for decision point analysis
Sue
Mike
Mary
Pete
Norman
check="OK" and report="Approved"
start
end
a
b
c
d
e
f
g
h
1391
1391
1391
566
566
971
971
1537
1537
1537
1537
1537
1537
461
461
930
930
146
146
146

Figure 3. The Process Model Can Be Extended Using Event Attributes Such as Timestamps, Resource Information and Case Data. The model also shows frequencies, e.g. 1,537 times a decision was made and 930 cases where rejected.!

5.1. Provide Insights

In the last decade, we have applied our process mining software ProM in over 100 organizations. Examples are municipalities (about 20 in total, e.g., Alkmaar, Heusden, and Harderwijk), government agencies (e.g., Rijkswaterstaat, Centraal Justitieel Incasso Bureau, and the Dutch Justice department), insurance related agencies (e.g., UWV), banks (e.g., ING Bank), hospitals (e.g., AMC hospital and Catharina hospital), multinationals (e.g., DSM and Deloitte), high-tech system manufacturers and their customers (e.g., Philips Healthcare, ASML, Ricoh, and Thales), and media companies (e.g., Winkwaves). For each of these organizations, we discovered some of their processes based on the event data they provided. In each discovered process, there were parts that surprised some of the stakeholders. The variability of processes is typically much bigger than expected. Such insights represent a tremendous value as surprising differences often point to waste and mismanagement.

5.2. Improve Performance

As explained earlier, it is possible to replay event logs on discovered or hand-made process models. This can be used for conformance checking and model enhancement. Since most event logs contain timestamps, replay can be used to extend the model with performance information.

Figure 4 illustrates some of the performance-related diagnostics that can be obtained through process mining. The model shown was discovered based on 745 objections against the so-called WOZ ("*Waardering Onroerende Zaken*") valuation in a Dutch municipality. Dutch municipalities need to estimate the value of houses and apartments. The WOZ value is used as a basis for determining the real-estate property tax. The higher the WOZ value, the more tax the owner needs to pay. Therefore, many citizens appeal against the WOZ valuation and assert that it is too high.

Each of the 745 objections corresponds to a process instance. Together these instances generated 9,583 events all having timestamps. **Figure 4** shows the frequency of the different paths in the model. Moreover, the different stages of the model are colored to show where, on average, most time is spent. The purple stages of the process take most time whereas the blue stages take the least time. It is also possible to select two activities and measure the time that passes in-between these activities.

As shown in **Figure 4**, on average, 202.73 days pass in-between the completion of activity "*OZ02 Voorbereiden*" (preparation) and the completion of "*OZ16 Uitspraak*" (final judgment). This is longer than the average overall flow time which is approx. 178 days. About 416 of the objections (approx. 56%) follow this route; the other cases follow the branch "*OZ15 Zelf uitspraak*" which, on average, takes less time.

Diagnostics as shown in **Figure 4** can be used to improve processes by removing

> “ Often such a ‘PowerPoint reality’ has little in common with the real processes that have much more variability. However, to improve conformance and performance, one should not abstract away this variability ”

bottlenecks and rerouting cases. Since the model is connected to event data, it is possible to "drill down" immediately and investigate groups of cases that take more time than others [1].

5.3. Ensure Conformance

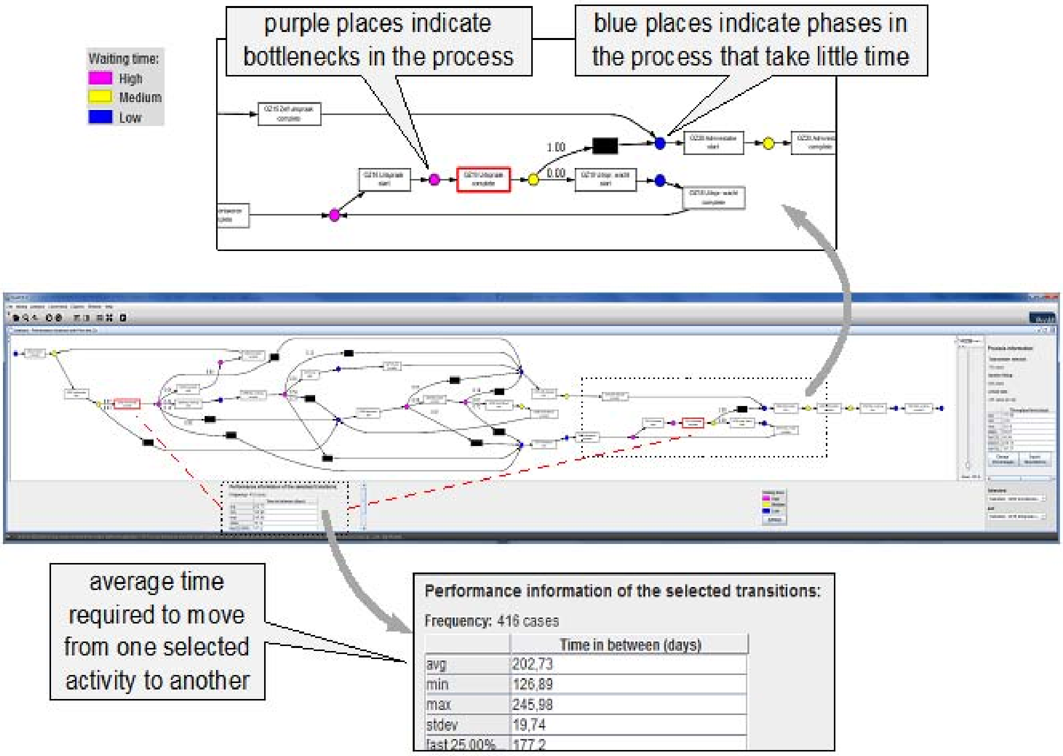
Replay can also be used to check conformance as is illustrated by **Figure 5**. Based on 745 appeals against the WOZ valuation, we also compared the normative model and the observed behavior: 628 of the 745 cases can be replayed without encountering any problems. The fitness of the model and log is 0.98876214 indicating that almost all recorded events are explained by the model. Despite the good fitness, ProM clearly shows all deviations. For example, "*OZ12 Hertaxeren*" (reevaluate property) occurred 23 times while this was not allowed according to the normative model (indicated by the "-23" in **Figure 5**). Again it is easy to "drill down" and see what these cases have in common.

The conformance of the appeal process just described is very high (about 99% of events are possible according to the model). We also encountered many processes with a very low conformance, e.g., it is not uncommon to find processes where only 40% of the events are possible according to the model. For example, process mining revealed that ASML’s modeled test process strongly deviated from the real process [9]. The increased importance of corporate governance, risk and compliance management, and legislation such as the Sarbanes-Oxley Act (SOX) and the Basel II Accord, illustrate the practical relevance of conformance checking. Process mining can help auditors to check whether processes are executed within certain boundaries set by managers, governments, and other stake holders [3]. Violations discovered through process mining may indicate fraud, malpractice, risks, and inefficiencies. For example, in the municipality where we analyzed the WOZ appeal process, we discovered misconfigurations of their eiStream workflow management system. People also bypassed the system. This was possible because system administrators could manually change the status of cases [8].

5.4. Show Variability

Hand-made process models tend to provide an idealized view on the business process that is modeled. Often such a "PowerPoint reality" has little in common with the real processes that have much more variability. However, to improve conformance and performance, one should not abstract away this variability.

In the context of process mining we often see Spaghetti-like models such as the one shown in **Figure 6**. The model was discovered based on an event log containing 24,331 events referring to 376 different activities. The event log describes the diagnosis and treatment of 627 gynecological oncology


Waiting time:
High
Medium
Low
purple places indicate bottlenecks in the process
blue places indicate phases in the process that take little time
1.00
0.00
average time required to move from one selected activity to another

Performance information of the selected transitions:

Frequency: 416 cases

	Time in between (days)
avg	202,73
min	126,89
max	245,98
stdev	19,74
fast 25.00%	177,2

Figure 4. Performance Analysis Based on 745 Appeals against the WOZ Valuation.

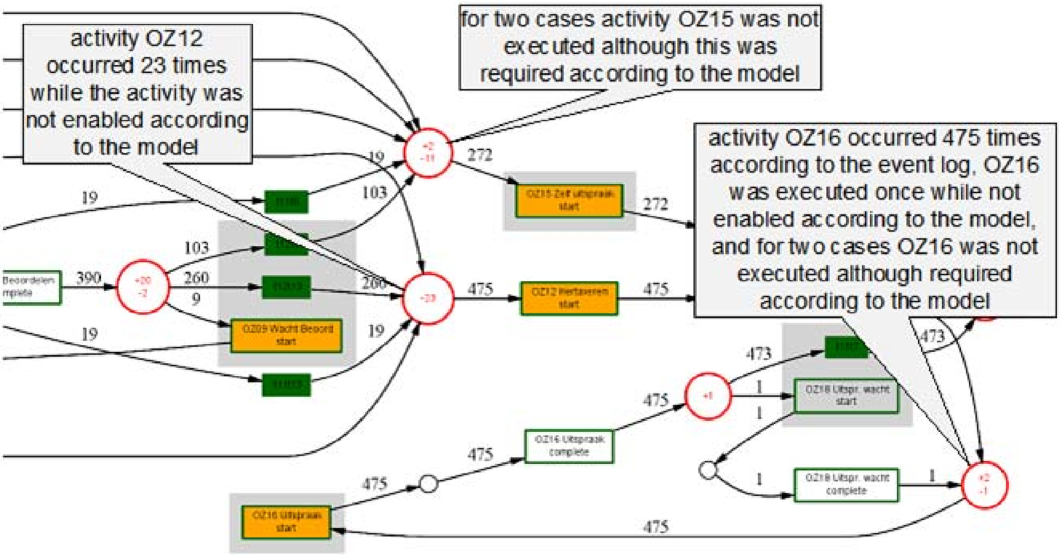

activity OZ12 occurred 23 times while the activity was not enabled according to the model
for two cases activity OZ15 was not executed although this was required according to the model
activity OZ16 occurred 475 times according to the event log, OZ16 was executed once while not enabled according to the model, and for two cases OZ16 was not executed although required according to the model
OZ15 Zelf uitspraak start
OZ12 Vertaxeren start
OZ09 Wacht Beoord start
OZ16 Uitspraak complete
OZ18 Uitspr. wacht start
OZ18 Uitspr. wacht complete
OZ16 Uitspraak start
Beoordelen complete
390
260
103
19
9
272
475
473
200
1
+2 -11
+20 -2
-23
+1
+2 -1

Figure 5. Conformance Analysis Showing Deviations between Eventlog and Process.

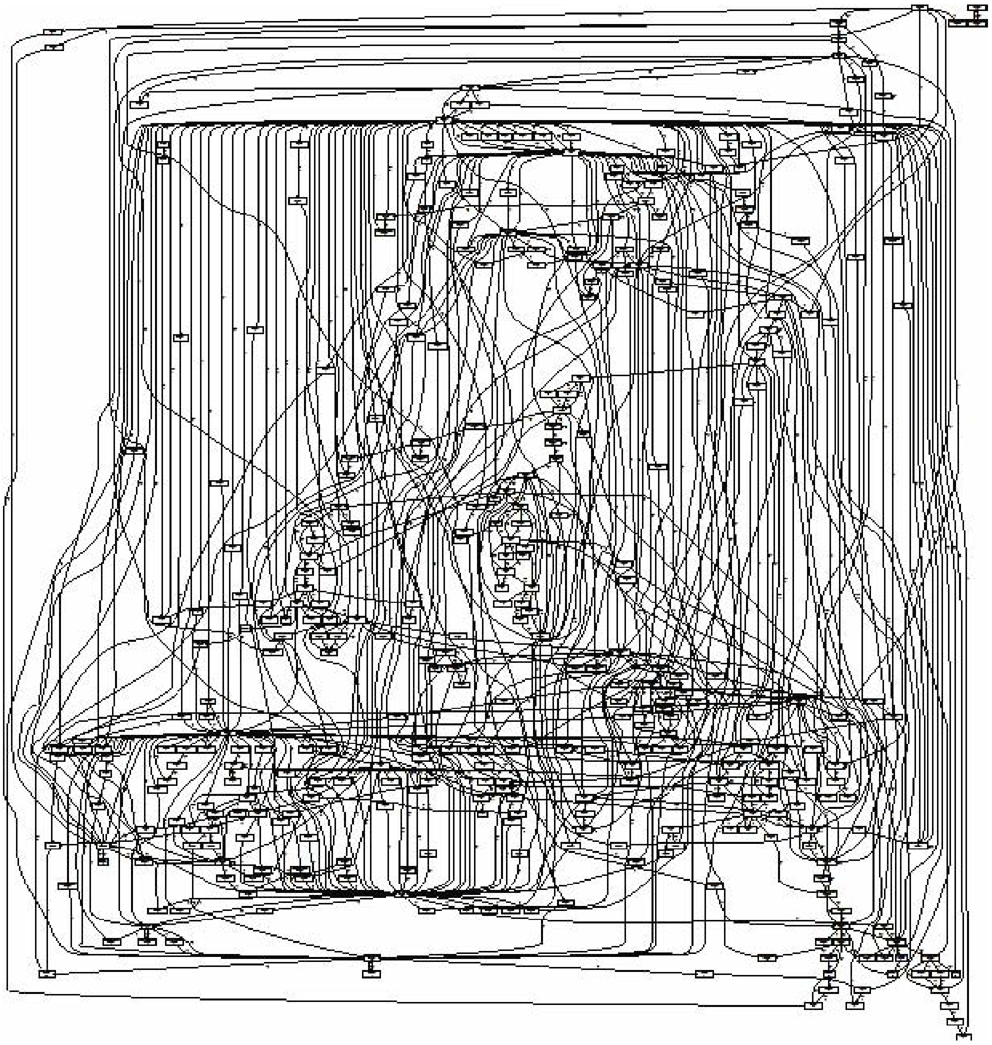


Figure 6. Process Model Discovered for a Group of 627 Gynecological Oncology Patients.

patients in the AMC hospital in Amsterdam. The Spaghetti-like structures are not caused by the discovery algorithm but by the true variability of the process.

Although it is important to confront stakeholders with the reality as shown in Fig. 6, we can also seamlessly simplify Spaghetti-like models. Just like using electronic maps it is possible to seamlessly zoom in and out [1]. While zooming out, insignificant things are either left out or dynamically clustered into aggregate shapes – like streets and suburbs amalgamate into cities in Google Maps. The significance level of an activity or connection may be based on frequency, costs, or time.

5.5. Improve Reliability

Process mining can also be used to improve the reliability of systems and processes. For example, since 2007 we have been involved in an ongoing effort to analyze the event logs of the X-ray machines of Philips Healthcare using process mining [1]. These machines record massive amounts of events. For medical equipment it is essential to prove that the system was tested under realistic circumstances. Therefore, process discovery was used to construct realistic test profiles. Philips Healthcare also used process mining for fault diagnosis. By learning from earlier problems, it is possible to find the root cause for new problems that emerge. For example, using ProM, we have analyzed under which circumstances particular components are replaced. This resulted in a set of signatures. When a malfunctioning X-ray machine exhibits a particular "signature" behavior, the service engineer knows what component to replace.

5.6. Enable Prediction

The combination of historic event data with real-time event data can also be used to predict problems. For instance, Philips Healthcare can anticipate that an X-ray tube in the field is about to fail by discovering patterns in event logs. Hence, the tube can be replaced before the machine starts to malfunction.

Today, many data sources are updated in (near) real-time and sufficient computing power is available to analyze events as they occur. Therefore, process mining is not restricted to off-line analysis and can also be used for online operational support. For a running process instance it is possible to make predictions such as the expected remaining flow time [1].

6. Conclusion

Process mining techniques enable organizations to X-ray their business processes, diagnose problems, and get suggestions for treatment. Process discovery often provides new and surprising insights. These can be used to redesign processes or improve management. Conformance checking can be used to see where processes deviate. This is very relevant as organizations are required to put more emphasis on corporate governance, risks, and compliance. Process mining techniques offer a means to more rigorously check compliance while improving performance.

This article introduced the basic concepts and showed that process mining can provide value in several ways. The reader interested in process mining is referred to the first book on process mining [1] and the process mining manifesto [5] which is available in 12 languages. Also visit <www.processmining.org> for sample logs, videos, slides, articles, and software.

The author would like to thank the members of the IEEE Task Force on Process Mining and all that contributed to the Process Mining Manifesto and the ProM framework.

References

[1] **W. van der Aalst.** *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin, 2011. ISBN: 978-3-642-19345-3.

[2] **W. van der Aalst.** Using Process Mining to Bridge the Gap between BI and BPM. *IEEE Computer 44,* 12, pp. 77–80, 2011.

[3] **W. van der Aalst, K. van Hee, J.M. van Werf, M. Verdonk.** Auditing 2.0: Using Process Mining to Support Tomorrow's Auditor. *IEEE Computer 43,* 3, pp. 90–93, 2010.

[4] **M. Hilbert, P.Lopez.** The World's Technological Capacity to Store, Communicate, and Compute Information. *Science 332,* 6025, pp. 60–65, 2011.

[5] **TFPM Task Force on Process Mining.** Process Mining Manifesto. *Business Process Management Workshops*, F. Daniel, K. Barkaoui, and S. Dustdar, Eds. Lecture Notes in Business Information Processing Series, vol. 99. Springer-Verlag, Berlin, pp. 169–194, 2012.

[6] **J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, A. Byers.** Big Data: The Next Frontier for Innovation, Competition, and Productivity. McKinsey Global Institute, 2011. <http://www. mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation>.

[7] **J. Mendling, G. Neumann, W. van der Aalst.** Understanding the Occurrence of Errors in Process Models Based on Metrics. Proceedings of the OTM Conference on Cooperative information Systems (CoopIS 2007). En F. Curbera, F. Leymann, and M. Weske, Eds. *Lecture Notes in Computer Science Series, vol. 4803*. Springer-Verlag, Berlin, pp. 113–130, 2007.

[8] **A. Rozinat, W. van der Aalst.** Conformance Checking of Processes Based on Monitoring Real Behavior. *Information Systems 33, 1*, pp. 64–95, 2008.

[9] **A. Rozinat, I. de Jong, C. Günther, W. van der Aalst.** Process Mining Applied to the Test Process of Wafer Scanners in ASML. *IEEE Transactions on Systems, Man and Cybernetics, Part C 39,* 4, pp. 474–479, 2009.

The Process Discovery Journey

Josep Carmona
Software Department, Technical University of Catalonia, Spain

<jcarmona@lsi.upc.edu>

Abstract: *Process models are an invaluable element of an IT system: they can be used to analyze, monitor, or improve the real processes that provide the system's functionality. Technology has enabled IT systems to store in file logs the footprints of process executions, which can be used to derive the process models corresponding with the real processes, a discipline called Process Discovery. We provide an overview of the discipline together with some of the alternatives that exist nowadays.*

Keywords: *Formal Methods, Process Discovery, Software Engineering.*

Author

Josep Carmona received his MS and PhD degrees in Computer Science from the Technical University of Catalonia, in 1999 and 2004, respectively. He is an associate professor in the Software Department of the same university. His research interests include formal methods, concurrent systems, and process and data mining. He has co-authored more than 50 research papers in conferences and journals.

1. Introduction

The speed at which data grows in IT systems [1] makes it crucial to rely on automation in order to enable enterprises and institutions to manage their processes. Automated techniques open the door for dealing with large amounts of data, a mission unthinkable for a human's capabilities. In this paper we discuss one of these techniques: the discovery of process models. We now illustrate the main task behind process discovery by means of a (hopefully) funny example.

2. A Funny Example: The Visit of an Alien

Imagine that an alien visits you (see **Figure** 1) and, by some means, it wants to communicate the plan it has regarding its visit to the Earth. For obvious reasons, we cannot understand the alien's messages, that look like the one shown in **Figure** 2.

Although not knowing the meaning of each individual letter in the message above, one may detect that there are some patterns, e.g., a repetition for the sequence *I A C D M E* (first and last six letters in the sequence). So the question is: how can we represent the behavior of the aliens without knowing exactly the meaning of each single piece of information?

Figure 1. Our Imaginary Alien.

Process discovery may be a good solution for this situation: a process discovery algorithm will try to produce a (formal) model of the behavior underlying a set of sequences. For instance, the following formal model in the Business Process Modeling Notation (BPMN) [2] shown in **Figure 3** represents very accurately the behavior expressed in the alien's sequences. For those not familiar with the BPMN notation, the model above describes the following process: *after I occurs, then ('x' gateway) either branch B followed by X occurs, or branch A followed by C and D in parallel('+' gateway), and then M occurs. Both branches activate E which in turn reactivates I.* Clearly, even without knowing anything about the actions taken from the alien, the global structuring of these activities becomes very apparent from a simple inspection of the BPMN model.

Now imagine that at some point the meaning of each letter is decrypted: *evaluate the amount of energy in the Earth (I), high energy (B), invade the Earth (X), low energy (A), gather some human samples (C), learn the human reproduction system (D), teach humans to increase their energy resources (M), communicate the situation to the aliens in the closest UFO (E).* In the presence of this new information, the value of the model obtained is significantly incremented (although maybe one may not be relaxed after realizing the global situation that the model brings into light).

3. Anatomy of a Simple Process Discovery Algorithm

The previous example illustrates one of the main tasks of a process discovery algorithm: given a set of traces (called *log*) corresponding to a particular behavior under study, derive a formal model which represents faithfully the process producing these traces. In its simplest form, process discovery algorithms focus on the *control-flow* perspective of the process, i.e., the ordering activities are performed in order to carry out the process tasks. The previous example has considered this perspective.

A log must contain enough information to extract the sequencing of the activities that are monitored. Typically, a trace identifier, an activity name and a time stamp are required to enable the corresponding sequencing (by the time stamp) for the activities belonging to a given trace (determined by the trace identifier). Other information may be required if the discovery engine must take into account additional information, like resources (*what quantity was purchased?*), activity originator (*who performed that activity?*), activity duration (*how long does activity X last?),* among others. An example of a discovery algorithm that takes into account other dimension is the *social network miner* [3], that derives the network of collaborators that carry out a given process.

I A C D M E I B X E I A D C M E I B X E I A C D M E

Figure 2. A Message Sent by the Alien.

> “The core of a process discovery algorithm is the ability to extract the necessary information required to learn a model that will represent the process”

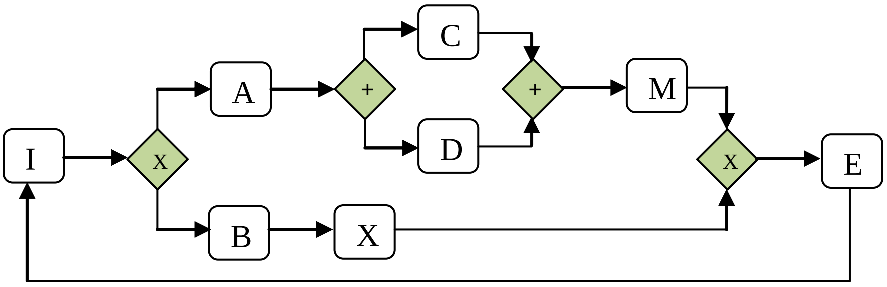


Figure 3. A Formal Model of Behavior in the Alien’s Sequences in BPMN.

The core of a process discovery algorithm is the ability to extract the necessary information required to learn a model that will represent the process. Process discovery is often an *unsupervised learning task*, since the algorithm is usually exposed only to positive examples, i.e., successful executions of the process under study: in the example of the introduction, we were only exposed to what the alien plans to do, but we do not know what the alien does not plan to do. This complicates the learning task, since process discovery algorithms are expected to produce models that are both *precise* (the model produced should not deviate much from the behavior seen) and *general* (the model should generalize the patterns observed in the log) [4]. Obviously, the presence of negative examples would help the discovery algorithm into improving these two quality metrics, but negative information is often not available on IT logs.

How to learn a process model from a set of traces? Various algorithms exist nowadays for various models (see **Section 4**). However, let us use the alien’s example to reason on the discovery of the BPMN model above. If we focus on the first letter of the sequence (I), it is sometimes followed by A and sometimes by B, and always (except for the first occurrence) preceded by E. These observations can be expressed graphically as shown in **Figure 4**.

In BPMN notation, the *or-exclusive* relation between the occurrences of either A or B after I is modeled by using the ‘x’ gateway. The precedence between E and I is modeled by an edge connecting both letters in the model. Symmetrically, E is preceded either by M or by X. Also, following A both C and D occur in any order. The well-known *alpha* algorithm [5] can find most of these pairwise ordering relations in the log, and one may use them to craft the BPMN model as **Table 1** illustrates.

Table 1 can be read as follows: if in the log A precedes B always but B is unique (there is no other letter preceded by A), then a directed arc between A and B is created. If in contrast there is always more than one letter preceded by A, then an ‘+’ gateway is inserted between A and the letters preceded by A. The sometimes relation can be read similarly.

Hence one can scan the log to extract these relations (worst-case quadratic in the size of the log) and use the table to create the BPMN model. However, this is a very restrictive way of discovery since other relations available in the BPMN notation can also be hidden in the log, like the *inclusive-or* relation, but the algorithm does not consider them. Process discovery algorithms are always in a trade-off between the complexity of the algorithm and the modeling capacity: the algorithm proposed in this section could be extended to consider also inclusive-or gateways, but that may significantly complicate the algorithm. Below we address informally these and other issues.

4. Algorithms and Models

There are several models that can be obtained through different process discovery algorithms: *Petri nets*, *Event-driven Process Chains*, *BPMN*, *C-Nets*, *Heuristic Nets*, *Business Process Maps*, among others. Remarkably, most of these models are supported by replay semantics that allow one to simulate the model in order to certify its adequacy in representing the log.

To describe each one of these models is out of the scope of this article, but I can briefly comment on Petri nets, which is a model often produced by discovery algorithms, due to its formal semantics and ability to represent concurrency. For the model of our running example, the corresponding Petri net that would be discovered by most of the Petri net discovery algorithms will be as shown in **Figure 5**.

Those readers familiar with Petri nets will find a perfect match between the underlying behavior of the Petri net and the alien’s trace. Notice that while in the BPMN model, apart from the units of information (in this case letters of the alphabet), there are other model components (gateways) whose semantics define the way the model represents the log traces.

The same happens with the Petri net above, where the circles correspond to the global behavior of the model, which is distributed among the net (only some circles are marked). While the discovery algorithm for BPMN needs to find both the connections and gateways, the analogous algorithm for Petri nets must compute the circles and connections.

Several techniques exist nowadays to accomplish the discovery of Petri nets, ranging from the log-ordering relations extracted by the alpha algorithm, down to very complex graph-based structures that are computed on top of an automaton representing the log traces.

What process discovery algorithm/modeling notation to choose? This is in fact a very good question that can only be answered partially: there is no one model that is better than the rest, but instead models that are better than others only for a particular type of behaviors. Actually, deciding the best

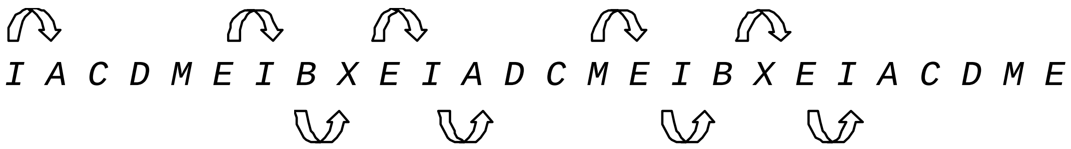


Figure 4. Patterns Observed in the Alien's Messages.

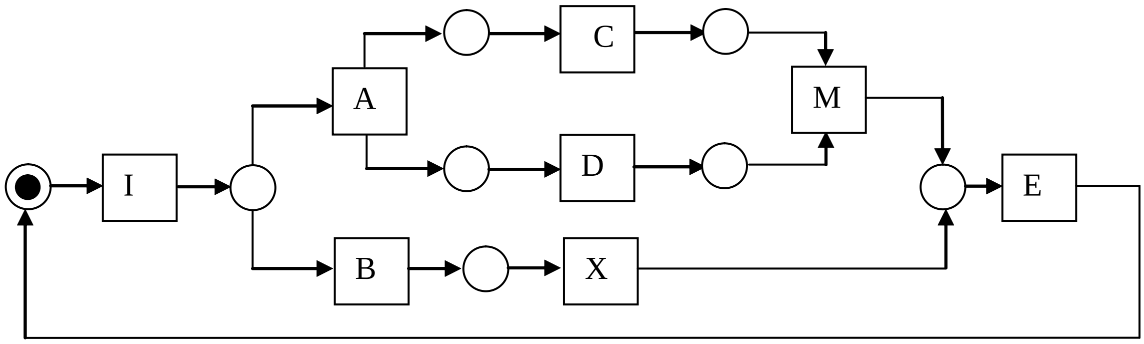


Figure 5. Petri Net for the Model of Our Running Example.

modeling notation for a log is a hard problem for which research must provide techniques in the next decade (a problem called *representational bias selection*). From a pragmatic point of view, one must select those process modeling notations one is familiar with, and expect the discovery algorithms for that notation to be good enough for the user needs.

As said before, other perspectives different from the control-flow view may be considered by process discovery algorithms: time, resources, organizational, etc.

The reference book [6] may be consulted in order to dig into these other process discovery algorithms.

5. Tools

Process discovery is a rather new discipline, if compared with related areas such as data mining or machine learning. In spite of this, one can find process mining tools both in academia (mostly) but also in industry.

The following classification is by no means exhaustive, but instead reports some of the prominent tools one can use to experience with process discovery tools:

■ ACADEMIA: the ProM Framework, from Technical University of Eindhoven (TU/e) is the reference tool nowadays. It is the result of a great academic collaboration among several universities in the world to gather algorithmic support for process mining (i.e., not only process discovery). Additionally, different groups have developed several academic stand-alone tools that incorporate modern process discovery algorithms.

■ INDUSTRY: some important companies have invested an effort into building process discovery tools, e.g., Fujitsu (APD), but also medium-sized or start-ups that are more focused on process mining practices, e.g.,

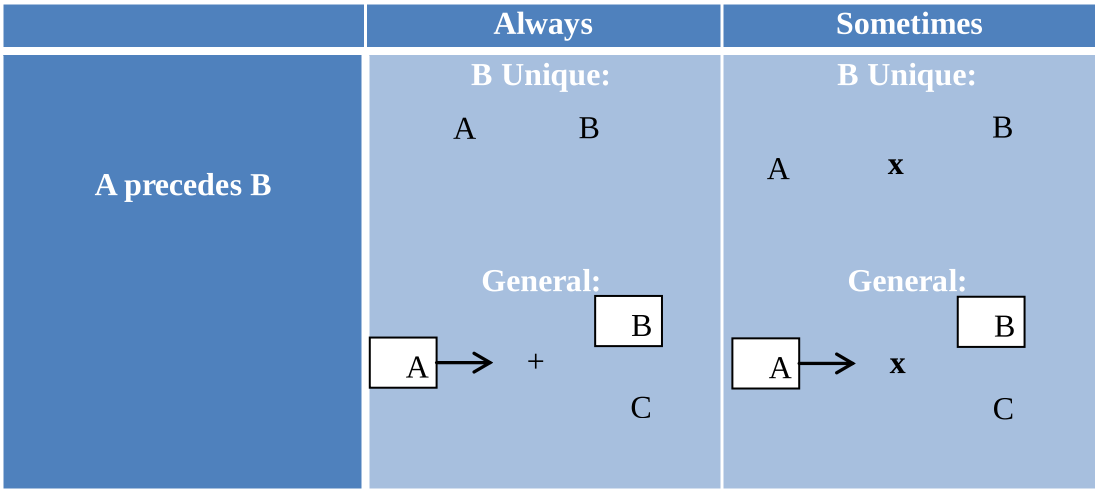

Always
Sometimes
A precedes B
B Unique:
A
B
B Unique:
A
x
B
General:
A
+
B
C
General:
A
x
B
C

Table 1. BPMN Model Built from Patterns in the Alien's Messages.

"Actually, deciding the best modeling notation for a log is a hard problem for which research must provide techniques in the next decade"

Pallas Athena (ReflectOne), Fluxicon (Disco), Perspective Software (BPMOne, Futura Reflect), Software AG (ARIS Process Performance Manager), among others.

6. Challenges

The task of process discovery may be aggravated if some of the aspects below are present:

- *Log incompleteness:* the log often contains only a fraction of the total behavior representing the process. Therefore, the process discovery algorithm is required to guess part of the behavior that is not present in the log, which may be in general a difficult task.
- *Noise*: logged behavior may sometimes represent infrequent exceptions that are not meant to be part of the process. Hence, process discovery algorithms may be hampered when noise is present, e.g., in control-flow discovery some relations between the activities may become contradictory. To separate noise from the valid information in a log is a current research direction.
- *Complexity*: due to the magnitude of current IT logs, it is often difficult to use complex algorithms that may either require loading the log into memory in order to derive the process model, or apply techniques whose complexity are not linear on the size of the log. In those cases, high level strategies (e.g., *divide-and-conquer*) are the only possibility to derive a process model.

§ *Visualization*: even if the process discovery algorithm does its job and can derive a process model, it may be hard for a human to understand it if it has more than a hundred elements (nodes, arcs). In those cases, a hierarchical description, similar to the *Google Maps* application were one can zoom in or out of a model's part, will enable the understanding of a complex process model.

Acknowledgements

I would like to thank David Antón for creating the alien's drawing used in this paper.

References

[1] **S. Rogers.** Data is Scaling BI and Analytics-Data Growth is About to Accelerate Exponentially - Get Ready. *Information and Management - Brookfield*, *21(5)*:p. 14, 2011.

[2] **D. Miers, S.A. White.** *BPMN Modeling and Reference Guide: Understanding and Using BPMN.* Future Strategies Inc., 2008. ISBN-10: 0977752720.

[3] **W. M. P. van der Aalst, H. Reijers, M. Song.** Discovering Social Networks form Event Logs. *Computer Supported Cooperative Work*, *14(6)*:pp. 549-593, 2005.

[4] **A. Rozinat, W. M. P. van der Aalst.** Conformance Checking of Processes Based on Monitoring Real Behavior. *Information Systems*, *33(1)*:pp. 64-95, 2008.

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

[6] **W.M.P. van der Aalst.** *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer, 2011. ISBN-10: 3642193447.

Antonio Valle-Salas
Managing Partner of G2

<avalle@gedos.es>

Using Process Mining in ITSM

Abstract: *When it comes to information systems, ranging from copiers to surgical equipment or enterprise management systems, all the information about the processes executed using those systems are frequently stored in logs. Specifically for IT Service Management processes (ITSM), it is quite common for the information systems used to execute and control those processes to keep structured logs that maintain enough information to ensure traceability of the related activities. It would be interesting to use all that information to get an accurate idea of ??how the process looks like in reality, to verify if the real process flow matches the previous design, and to analyze the process to improve it in order to become more effective and efficient. This is the main goal of process mining. This paper explores the different capabilities of process mining and its applicability in the IT Service Management area.*

Keywords: *Change Management, ITSM, Process Management Tools, Process Mining, Service Desk, Services.*

Author

Antonio Valle-Salas is Managing Partner of G2 and a specialist consultant in ITSM (Information Technology Service Management) and IT Governance. He graduated as a Technical Engineer in Management Informatics from UPC (*Universitat Politécnica de Catalunya*) and holds a number of methodology certifications such as ITIL Service Manager from EXIN (Examination Institute for Information Science), Certified Information Systems Auditor (CISA) from ISACA, and COBIT Based IT Governance Foundations from IT Governance Network, plus more technical certifications in the HP Openview family of management tools. He is a regular collaborator with itSMF (IT Service Management Forum) Spain and its Catalan chapter, and combines consulting and project implementation activities with frequent collaborations in educational activities in a university setting (such as UPC or the *Universitat Pompeu Fabra*) and in the world of publishing in which he has collaborated on such publications as IT Governance: a Pocket Guide, Metrics in IT Service Organizations, *Gestión de Servicios TI. Una introducción a ITIL*, and the translations into Spanish of the books ITIL V2 Service Support and ITIL V2 Service Delivery.

1. Roles and Responsibilities in ITSM Models

All models, standards or frameworks used in the ITSM industry are process oriented. This is because process orientation helps structure the related tasks and allows the organization to formalize the great variety of activities performed daily: which activities to execute and when, who should carry them out, who owns what responsibilities over those tasks, which tools or information systems to use and what are the expected objectives and outcomes of the process.

One model commonly used to represent the different components of the process is the ITOCO model [1] **Figure 1** that represents the fundamental elements of a process: Inputs, Outputs, Tasks, Control parameters and Outcomes.

This model allows us to differentiate between three different roles needed for the correct execution of any process: process *operators*, who are responsible for executing the different tasks; process *managers*, who warrantee that the process execution meets the specifications and ensure that both inputs and outputs match the expectations (within the specified control parameters); and process *owners*, who use a governance perspective to define the process, its outcomes and the applicable controls and policies, as well as being responsible to obtain and allocate the resources needed for the right execution of the process.

The *process manager*'s job is the execution of the control activities (also called the control process) over the managed process, acting on the deviations or the quality variations of the results, and managing the allocated resources to obtain the best possible results. Therefore, this role requires a combination of skills from diverse professional disciplines such as auditing, consulting and, chiefly, continuous improvement.

2. ITSM Process Management

The ITSM industry has traditionally used a number of methodological tools to enable the process manager do the job:

- Definition of metrics and indicators (usually standardized from the adopted frameworks).
- Usage of Balanced Scorecards to show and follow those indicators.
- Definition of management reports (daily, weekly, monthly).
- Usage of various kinds of customer and/or user satisfaction surveys.
- Performance of internal or external compliance audits.

These tools allow the process manager to gain knowledge about the behavior of the processes she is in charge of, and to make decisions to set the correct course of tasks and activities. However these tools are commonly rather rigid whereas the process manager needs a deeper analysis of the process behaviour.

Still, there are two key aspects of any continuous improvement model: to know what the current situation is - as the starting point for the improvement trip - and to understand what the impact of the improvement initiatives will be on the process and its current situation. Both aspects are represented in **Figure 2**.

At these initial stages many questions arise regarding the daily activities of the process manager, namely:

- Which is the most common flow?
- What happens in some specific type of request?
- How long are the different cases at each state of the flow?
- Can we improve the flow?
- Where is the flow stuck?
- Which are the most repeated activities?
- Are there any bottlenecks?
- Are the process operators following the defined process?
- Is there segregation of duties in place?

Moreover, in ITSM we usually find that most processes defined using frameworks do not fully match the real needs of daily operations; a standard and rigid approach to processes does not meet the needs of those activities in which the next steps are not known in advance [2].

One clear case of this type of processes in ITSM is the problem management process. Here, to be able to execute the diagnostics and identification of root causes, the operator

> "The role of the process manager requires a combination of skills from diverse professional disciplines such as auditing, consulting and, chiefly, continuous improvement"

will have to decide the next step according to the results of the previous analysis. Thus, problem management is, by nature, a non-structured process whose behavior will totally differ from a strict process such as request management.

3. Process Mining & ITSM

The first and most delicate task when using process mining techniques is obtaining a log of good quality, representative of the process we want to analyze, and with enough attributes to enable filtering and driving subsequent analysis steps as shown in **Figure 3**.

Fortunately enough, most ITSM process management tools have logs that allow the actions executed by the various process actors to be traced. These logs (e.g. **Figure 4**) are usually between maturity levels IV and V on the scale proposed by the Process Mining Manifesto [3].

The following steps of discovery and representation are those in which the use of process mining techniques provides immediate value.

The designed processes are usually different to the real execution of activities. This is caused by various factors, amongst which we find too generalist process designs (to try to cover non-structured processes), flexibility of the management tools (that are frequently configured to allow free flows instead of closed flows) and process operator's creativity (they are not always comfortable with a strict process definition).

For this reason, both the *process owner* and the p*rocess manager*usually have an idealized view of the process, and so are deeply surprised the first time they see a graphic representation of the process from the analysis of the real and complete information.

For instance, as mentioned in USMBOK [4], the different types of request a user can log into a call center will be covered by a single concept of *Service Request* that will then follow a different flow or *Pathway* as shown in **Figure 5**. This flow will be "fitted" within a common flow in the corresponding module of the management tool used by the Service Desk team.

In order to fit this wide spectrum of different types of requests into a relatively general flow we usually avoid a closed definition of the process and its stages (in the form of a deterministic automat) but we allow an open flow as shown in **Figure 6** in which any operator decides at any given time the next state or stage of the corresponding life cycle [2].

That is why, when we try to discover and represent these types of activities, we find what in process mining jargon is called "spaghetti model" as shown in **Figure 7**. In this model, even with a reduced number of cases, the high volume and heterogeneous transactions between states makes the diagram of little (if any) use.

Therefore, to facilitate analysis, we need to use some techniques to divide the problem into smaller parts [5]. We can use clustering, or simply filtering the original log, in order to select the type of pathway we want to analyze.

Previous to the discovery and representation tasks, it is recommended that the log is enriched with any available information that will later allow segmenting the data set according to the various dimensions of analysis.

For instance, in this case we will need to have an attribute indicating the request type or pathway to be able to break down the model by requests, segmenting the data set

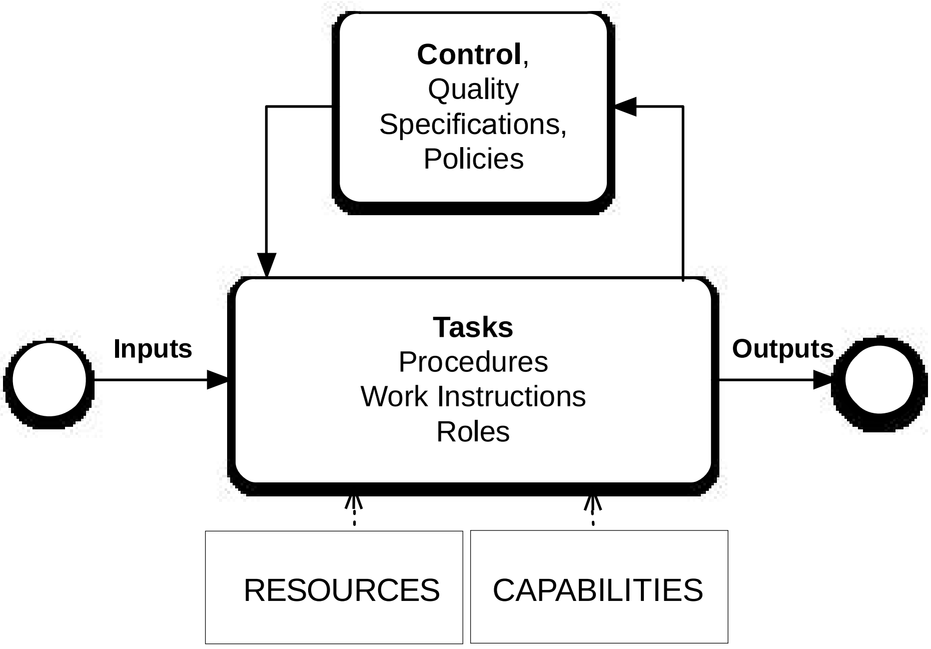


Figure 1. The ITOCO Model.

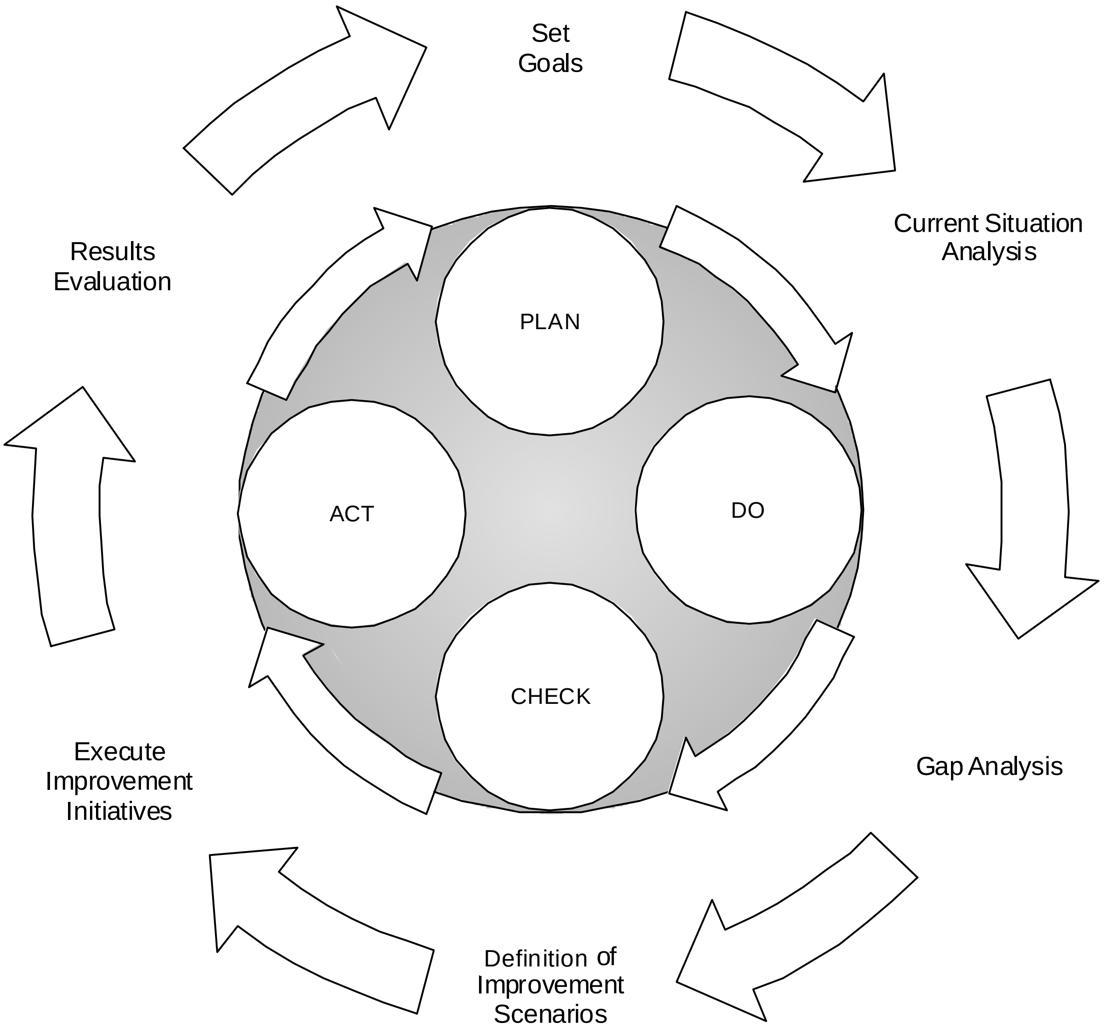

Set Goals
Current Situation Analysis
Gap Analysis
Definition of Improvement Scenarios
Execute Improvement Initiatives
Results Evaluation
PLAN
DO
CHECK
ACT

Figure 2. Continuous Improvement Cycle.

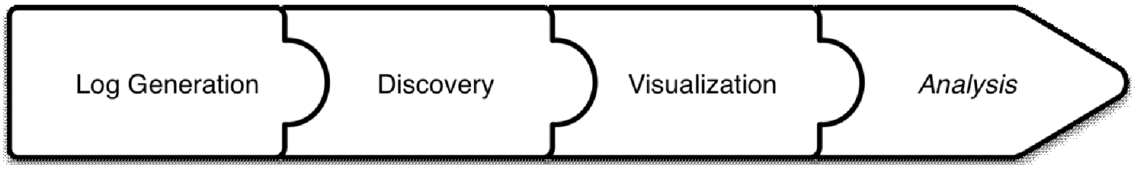
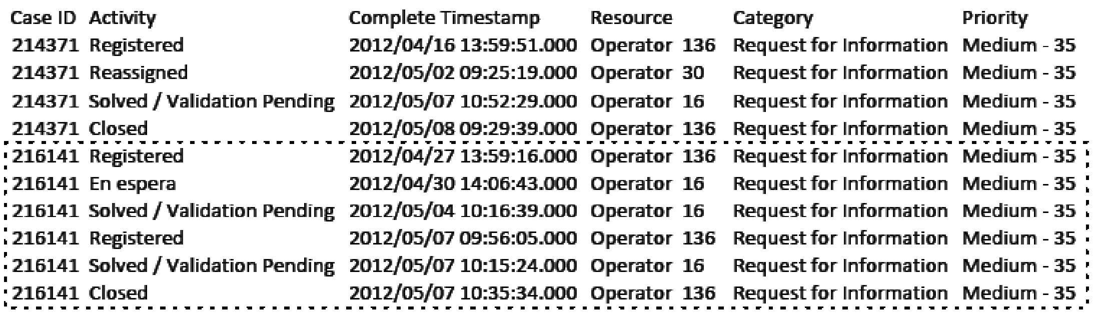


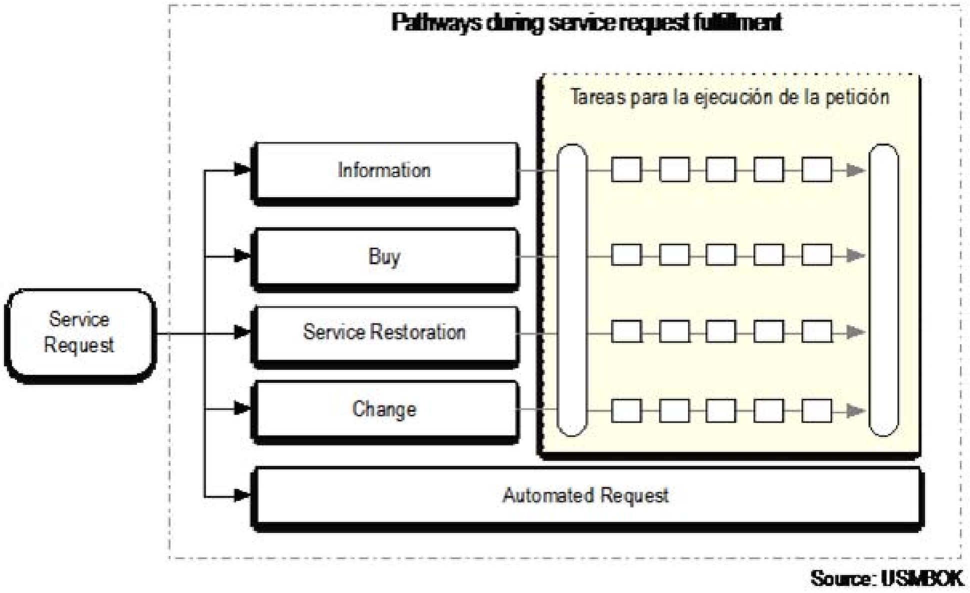
Figure 3. Sequence of Process Mining Steps.



Case ID	Activity	Complete Timestamp	Resource	Category	Priority
214371	Registered	2012/04/16 13:59:51.000	Operator 136	Request for Information	Medium - 35
214371	Reassigned	2012/05/02 09:25:19.000	Operator 30	Request for Information	Medium - 35
214371	Solved / Validation Pending	2012/05/07 10:52:29.000	Operator 16	Request for Information	Medium - 35
214371	Closed	2012/05/08 09:29:39.000	Operator 136	Request for Information	Medium - 35
216141	Registered	2012/04/27 13:59:16.000	Operator 136	Request for Information	Medium - 35
216141	En espera	2012/04/30 14:06:43.000	Operator 16	Request for Information	Medium - 35
216141	Solved / Validation Pending	2012/05/04 10:16:39.000	Operator 16	Request for Information	Medium - 35
216141	Registered	2012/05/07 09:56:05.000	Operator 136	Request for Information	Medium - 35
216141	Solved / Validation Pending	2012/05/07 10:15:24.000	Operator 16	Request for Information	Medium - 35
216141	Closed	2012/05/07 10:35:34.000	Operator 136	Request for Information	Medium - 35

Figure 4. Sample Log.

> “The first and most delicate task when using process mining techniques is obtaining a log of good quality, representative of the process we want to analyze”


Pathways during service request fulfillment
Tareas para la ejecución de la petición
Service Request
Information
Buy
Service Restoration
Change
Automated Request

Source: USMBOK

Figure 5. Pathways, According to USMBOK.

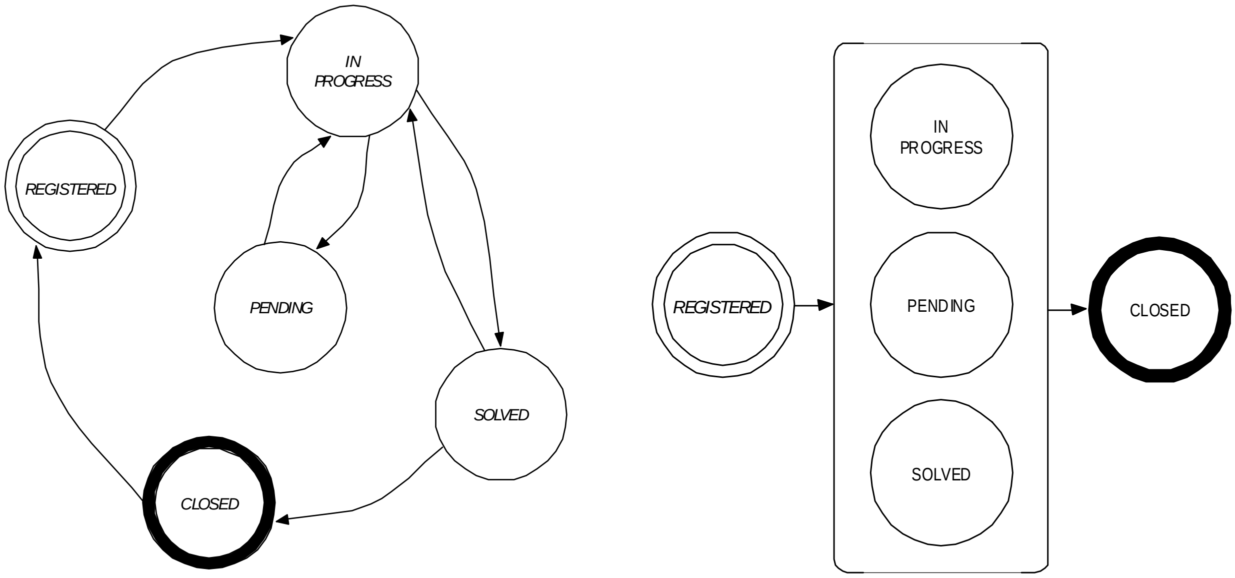


Figure 6. Strict Flow vs. Relaxed Flow.

Figure 7. Spaghetti Model.

and carrying on the analysis of a specific kind of request (see **Figure** 8).

On the other hand, we need to remember that process mining techniques are independent of the process activity. Instead, they focus in analyzing the changes of state.

At this point, we can be creative and think about the process flow as any "change of state within our information system", so we can use these techniques to analyze any other transitions such as the task assignment flow amongst different actors, the escalation flow amongst different specialist groups or (even lesser related to the common understanding of a process) ticket priority changes and re classifications (see **Figure** 9).

Finally, at the analysis stage it is time to answering questions about the process behaviour. To do this we have a broad array of tools:

- Enrich the visual representation: for example in **Figure** 9 we can observe that longer transactions between operators are represented in a thicker line, or in **Figure** 8 we show most frequent states in darker color.
- Graphs and histograms: to represent volume or time-related information. Typical situations of this kind of analysis are graphic representations of the number of open cases over time (backlog evolution) and histograms showing the distribution of duration and/or events per case.
- In more analytic fields, we can obtain a

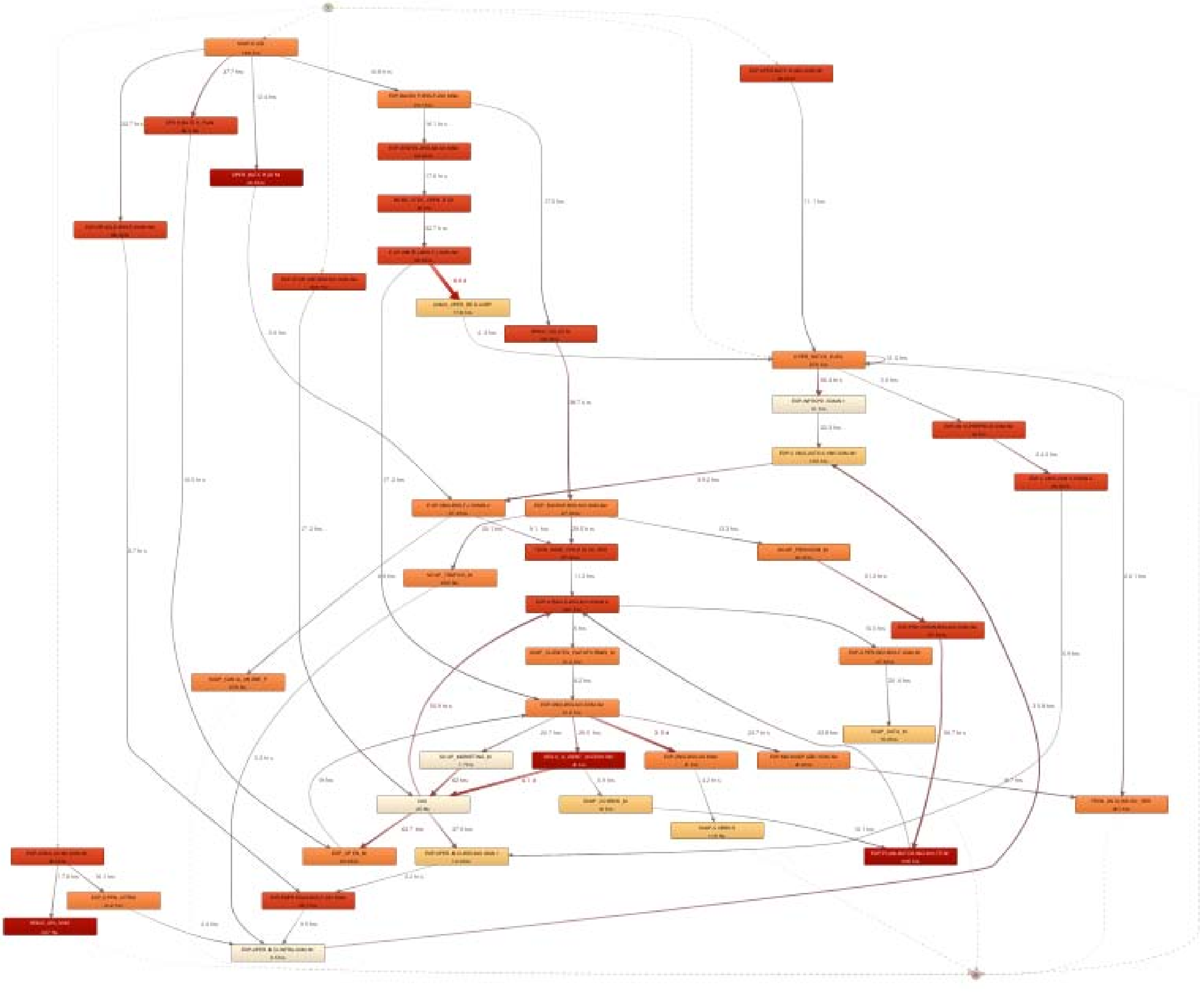


Figure 8. Filtered Spaghetti Model.

> “To facilitate analysis we need to use some techniques to divide the problem into smaller parts”

diagram showing a Márkov Chain for our process (see **Figure 10**). It will depict the probability for a particular transition to happen, to help answer questions like "what is the probability that a closed ticket will be re-opened?" We can also complement this information with case attributes: affected item, contact person, request type, organization etc. so that the model for analysis is richer.

So far we have covered methodological tools and mechanisms intended for quantitative and statistical analysis of processes and their behaviour. However, there is another side of the analysis focusing in the specific area of execution, answering questions such as "are there clear patterns of behavior in my process?", "is the process execution meeting previous definitions or corporate policies?" [6].
To answer the first question we will use the concept of "variant". We can describe a variant as the set of cases executed following the same trace of sequence of events. Thus, it is possible that some types of requests are always completed in a common pattern. We will easily check this by analyzing the variants of our process as shown in **Figure 11** (right side), where we see 79% of cases following the same flow: Registered à Completed / Validation à Closed.

To answer the second questions about process conformance we must have a formal model of the process to compare with its real execution. Once we have this piece, we can carry out different approaches to the problem of validation of conformance, as described by Anne Rozinat in her paper *Conformance Checking of Processes Based on Monitoring Real Behavior* [7]:

- Fitness analysis: answers the question "is the observed process complying with the process flow specified in the model?"
- Appropriateness analysis: answers the question "does the process model describe the observed process appropriately?"

Nevertheless, calculating some fitness index to a particular model will not be enough when doing analysis or audits; in those situations we will need ways to do complex consultations of the log [8]. To be able to know in which situations activity A is executed before activity B, or when did operator X executes activities A and B, will be of great importance to unveil violations of business rules or policies that govern the execution of the process.

If these techniques are applied to ITSM processes, we can provide an interesting application to ensure segregation of duties in change management for those organizations needing to comply with SOX or similar regulations. Next step would be continuously monitoring of these rules [9].

4. Conclusions

Process mining is presented as a set of tools that facilitate to a large degree process owners' and process managers' tasks, from acquiring knowledge about the real behaviour of the process to audits and continuous improvement. They allow many analyses that would be practically impossible or extremely costly to perform using traditional strategies like reporting, dashboarding or measuring indicators.

Although, generally speaking, one of the biggest difficulties we find in process mining is the lack of information or logs to analyze, in the specific area of IT service management

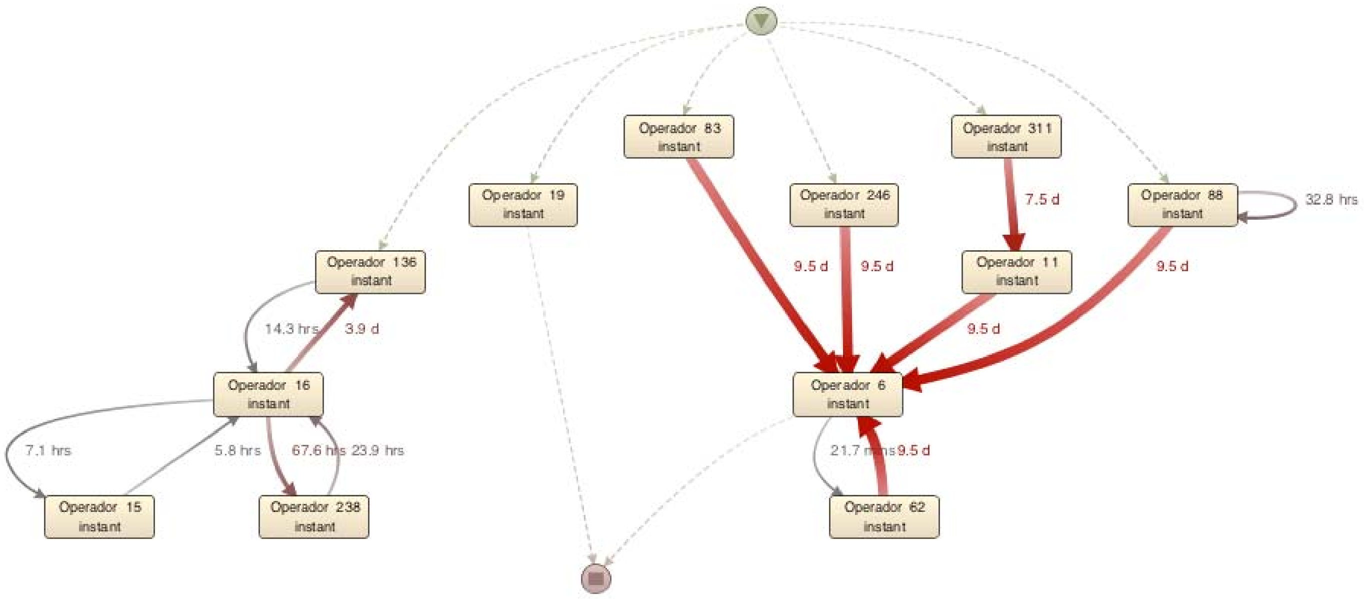


Figure 9. Social Map: How Cases Flow through the Different Process Operators.

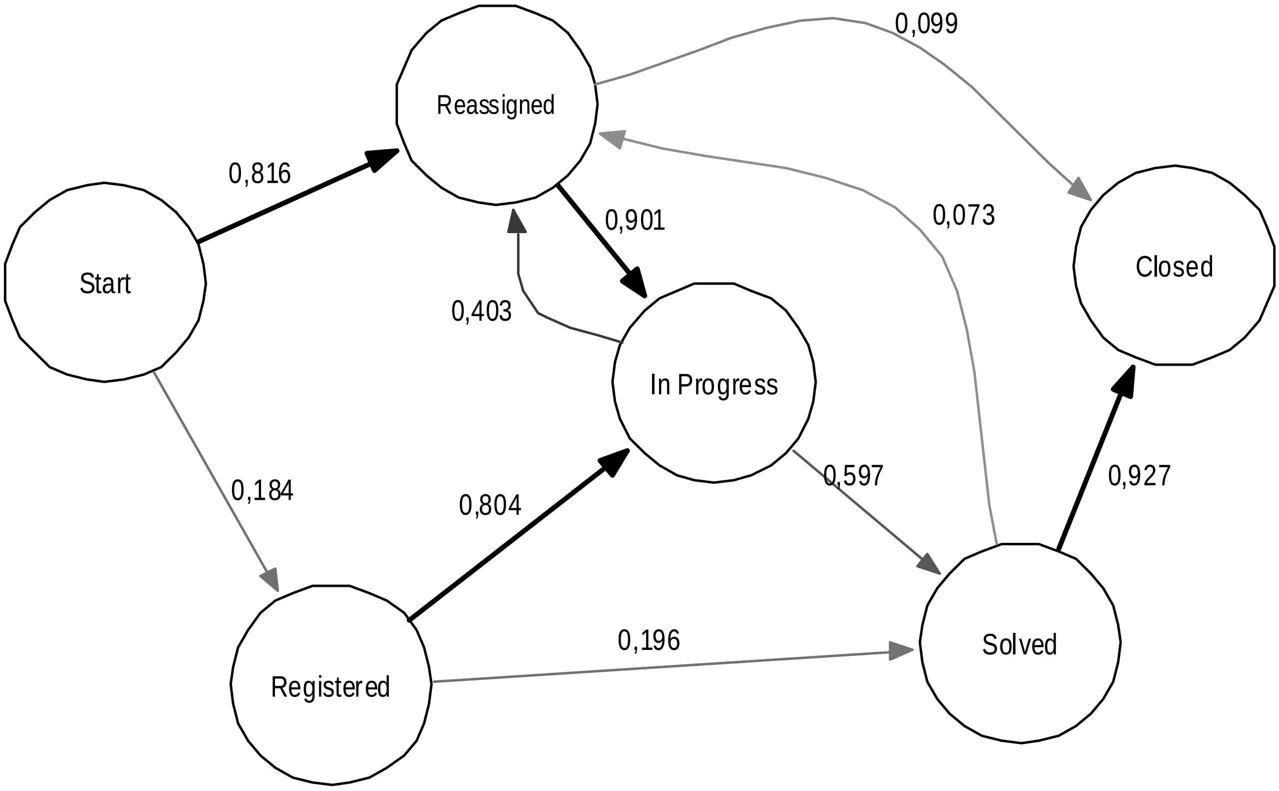
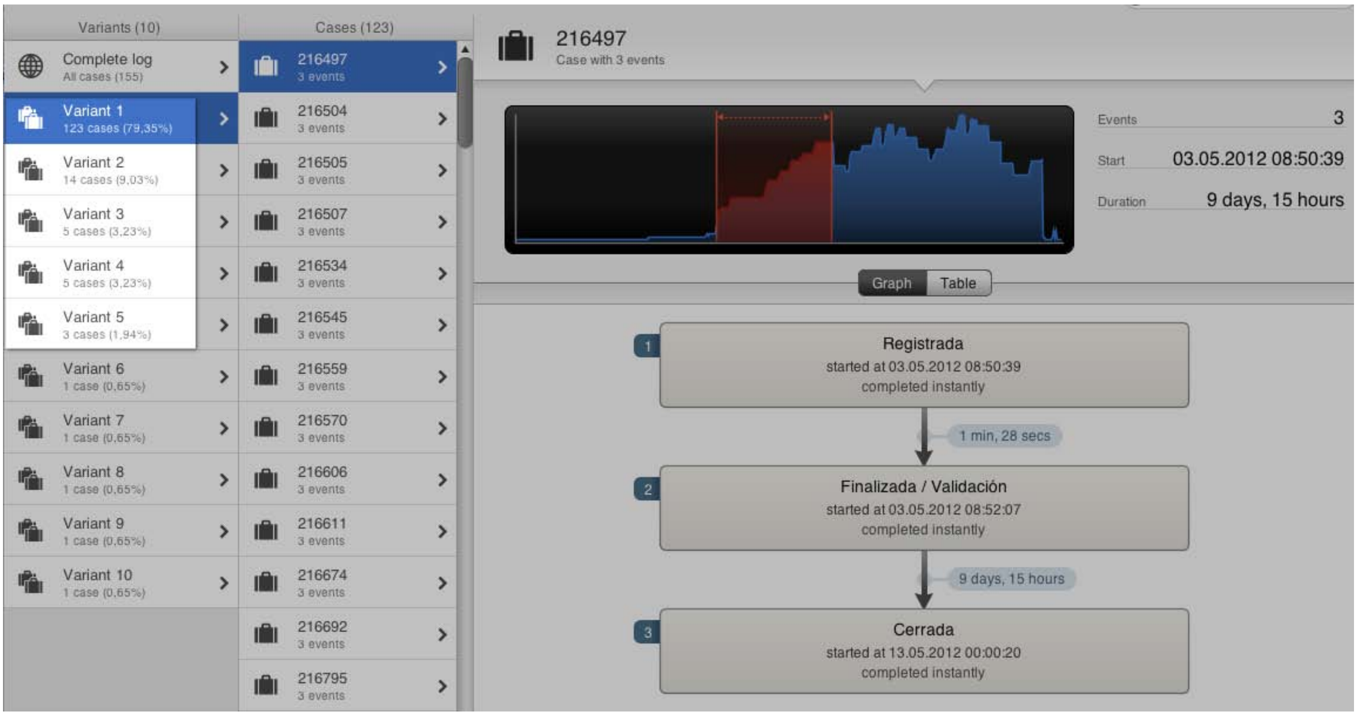


Figure 10. Simplified Márkov Chain.



Variants (10)	
Complete log	All cases (155)
Variant 1	123 cases (79,35%)
Variant 2	14 cases (9,03%)
Variant 3	5 cases (3,23%)
Variant 4	5 cases (3,23%)
Variant 5	3 cases (1,94%)
Variant 6	1 case (0,65%)
Variant 7	1 case (0,65%)
Variant 8	1 case (0,65%)
Variant 9	1 case (0,65%)
Variant 10	1 case (0,65%)

Cases (123)	
216497	3 events
216504	3 events
216505	3 events
216507	3 events
216534	3 events
216545	3 events
216559	3 events
216570	3 events
216606	3 events
216611	3 events
216674	3 events
216692	3 events
216795	3 events

216497
Case with 3 events

Events	3
Start	03.05.2012 08:50:39
Duration	9 days, 15 hours

1 Registrada
started at 03.05.2012 08:50:39
completed instantly

1 min, 28 secs

2 Finalizada / Validación
started at 03.05.2012 08:52:07
completed instantly

9 days, 15 hours

3 Cerrada
started at 13.05.2012 00:00:20
completed instantly

Figure 11. Process Variants.

this is not a problem. Here ITSM process management tools keep logs that can be used for process mining and can define auditable fields to be traced giving us different perspectives or dimensions of analysis.

Therefore, process mining stands out as a powerful and adequate tool to support ITSM practices in its permanent quest for improvement opportunities, both for processes and services of the management system.

References

[1] **Jan van Bon.** *IT Service Management Global Best Practices*, Volume 1. NL, NL: Van Haren Publishing, 2008.

[2] **Rob England.** *Plus! The Standard+Case Approach*. Wellington, NZ: CreateSpace, 2013.

[3] **IEEE Task Force on Process Mining.** *Process Mining Manifesto* (in 12 languages). <http://www.win.tue.nl/ieeetfpm/doku.php?id=shared:process_mining_manifesto>.

[4] **Ian M. Clayton.** *USMBOK - The Guide to the Universal Service Management Body of Knowledge*. CA, US: Service Management 101, 2012.

[5] **Marco Aniceto Vaz, Jano Moreira de Souza, Luciano Terres, Pedro Miguel Esposito.** *A Case Study on Clustering and Mining Business Processes from a University*, 2011.

[6] **Wil M.P. van der Aalst et al.** *Auditing 2.0: Using Process Mining to Support Tomorrow's Auditor*, 2010. <http://bpmcenter.org/wp-content/uploads/reports/2010/BPM-10-07.pdf>.

[7] **Anne Rozinat, W.M.P. van der Aalst.** *Conformance Checking of Processes Based on Monitoring Real Behavior*, 2008. <http://wwwis.win.tue.nl/~wvdaalst/publications/p436.pdf>.

[8] **W.M.P. van der Aalst, H.T. de Beer, B.F. van Dongen.** *Process Mining and Verification of Properties: An Approach based on Temporal Logic*, 2005.

[9] **Linh Thao Ly, Stefanie Rinderle-Ma, David Knuplesch, Peter Dadam.** Monitoring Business Process Compliance Using Compliance Rule Graphs, 2011. <http://dbis.eprints.uni-ulm.de/768/1/paper.pdf>.

Arjel Bautista, Lalit Wangikar, S.M. Kumail Akbar
CKM Advisors, 711 Third Avenue, Suite 1806, New York, NY, USA

<{abautista,lwangikar,sakbar}@ckmadvisors.com>

Process Mining-Driven Optimization of a Consumer Loan Approvals Process

Abstract: *An event log (262,200 events; 13,087 cases) of the loan and overdraft approvals process from a bank in the Netherlands was analyzed using a number of analytical techniques. Through a combination of spreadsheet-based approaches, process mining capabilities and exploratory analytics, we examined the data in great detail and at multiple levels of granularity. We present our findings on how we developed a deep understanding of the process, assessed potential areas of efficiency improvement and identified opportunities to make knowledge-based predictions about the eventual outcome of a loan application. We also discuss unique challenges of working with such data, and opportunities for enhancing the impact of such analyses by incorporating additional data elements.*

Keywords: *Big Data, Business Process Intelligence, Data Analytics, Process Mining.*

Authors

Arjel Bautista is a consultant at CKM Advisors, involved in the development of innovative process re-engineering and analytical research techniques within the firm. In his projects, Arjel has deployed a combination of state-of-the-art data mining tools and traditional strategic analysis to solve a variety of problems relating to business processes. He has also developed strategies for the analysis of unstructured text and other non-traditional data sources. Arjel holds Masters and Doctorate degrees in Chemistry from Yale University, and a Bachelor's degree in Biochemistry from UC San Diego.

Lalit Wangikar is a Partner at CKM Advisors. As a consultant, Lalit has advised clients primarily in the financial services sector, insurance, and payment services industries. His primary area of expertise is use of Big Data and Analytics for driving business impact across all key business areas such as marketing, risk, operations and compliance. He has worked with clients in North America, UK, Singapore and India. Prior to joining CKM Advisors, Lalit ran Decision Analytics practice for EXL Service / Inductis. Prior to that he worked as a consultant with Deloitte Consulting and Mitchell Madison Group where he advised clients in banking and capital markets verticals.

Syed M. Kumail Akbar is a Consultant at CKM Advisors where he is a member of the Analytics Team and assists in data mining, process mapping and predictive analytics. In the past, he has worked on strategy and operations projects in the financial services industry. Before joining CKM, Syed worked as a research assistant in both the Quantitative Analysis Center and the Physics Department at Wesleyan University. He also co-founded Possibilities Pakistan, a Non-Governmental Organization dedicated to providing access to college counseling for high school students in Pakistan. Syed holds a BA in Physics and Mathematics-Economics from Wesleyan University.

1. Introduction

As the role of Big Data gains prevalence in this information-driven era [1][2][3], businesses the world over are constantly searching for ways to take advantage of these potentially valuable resources. The 2012 Business Processing Intelligence Challenge (BPIC, 2012) is an exercise in analyzing one such data set using a combination of commercial, proprietary, and open-source tools, and combining these with creative insights to better understand the role of process mining in the modern workplace.

1.1. Approach and Scope

The situation depicted in BPIC 2012 focuses on the loan and overdraft approvals process of a real-world financial institution in the Netherlands. In our analysis of this information, we sought to understand the underlying business processes in great detail and at multiple levels of granularity. We also sought to identify any opportunities for improving efficiency and effectiveness of the overall process. Specifically, we attempted to investigate the following areas in detail:

- ■ Develop a thorough understanding of the data and the underlying process.
- ■ Understand critical activities and decision points.
- ■ Map the lifecycle of a loan application from start to eventual disposition.
- ■ Identify any resource-level differences in performance and opportunities for process interventions.

As newcomers to process mining, we at CKM Advisors wanted to use this opportunity to put into practice our learning in this discipline. We also attempted to combine process mining tools with traditional analytical methods to build a more complete picture. We are certain that with experience, our approach will become more refined and increasingly driven by methods developed specifically for process mining.

Our attempt was to be as broad as possible in our analysis and delve deep where we could. While we have done detailed analysis in a few areas, we have not covered all possible areas of process mining in our analysis. Any areas that we did not cover (for example, social network analysis) are driven solely by our own comfort and familiarity with the subject matter, and not necessarily a limitation of the data.

2. Materials and Methods

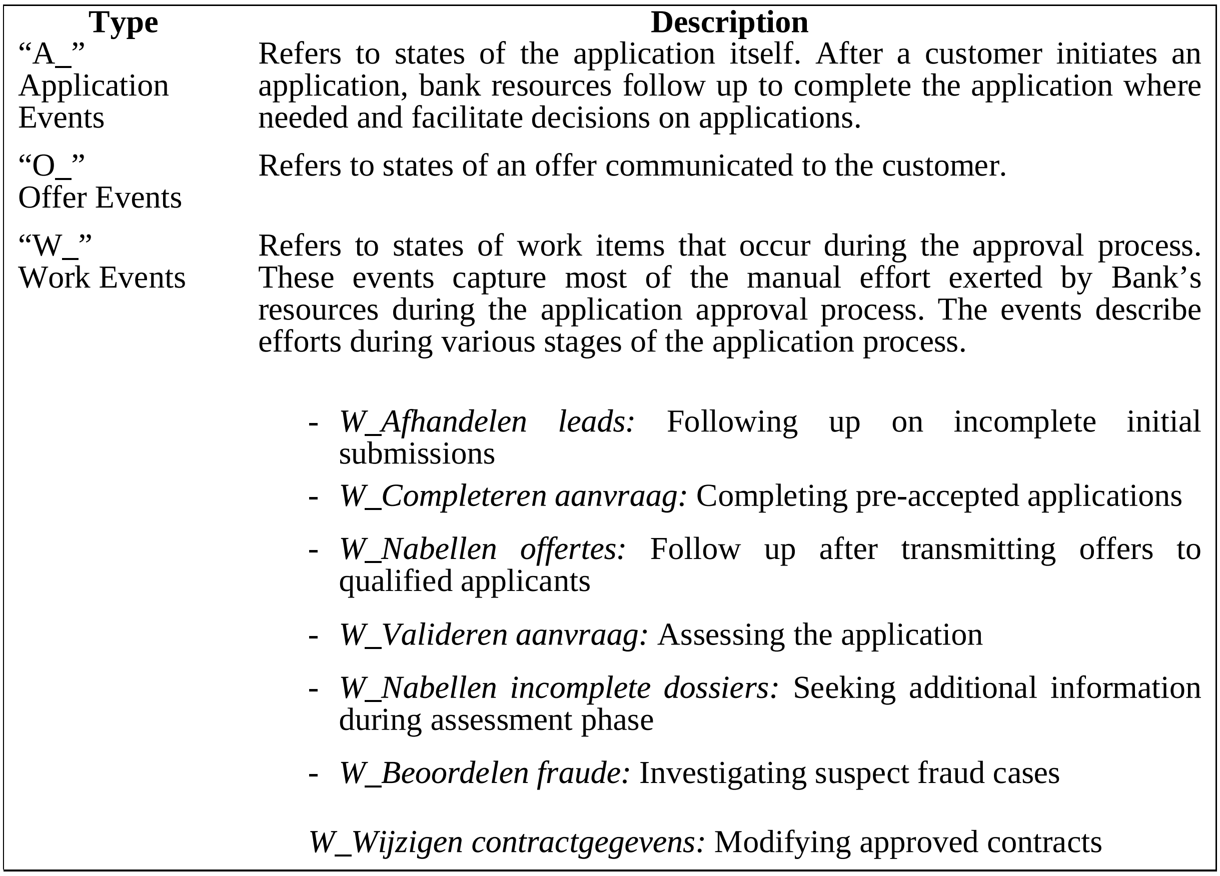
2.1. Understanding the Data

The data captures process events for 13,087 loan / overdraft applications over a six month period, between October 2011 and March 2012. The event log is comprised of a total of 262,200 events within these cases, starting with a customer submitting an application and ending with eventual conclusion of that application into an approval, cancellation or rejection (declined). Each application contains a single attribute, AMOUNT_REQ, which indicates the amount requested by the applicant. For each event, the extract shows the type of event, lifecycle stage (Schedule, Start, Complete), a resource indicator and time of completion.

The events themselves describe steps along the approvals process and are classifiedinto three major types. **Table 1** shows the event types and our understanding of what the events mean.

By itself, the event log is a complicated mass of information from which it is difficult to draw logical conclusions. Therefore, as other researchers have noted [4][5], it is necessary to subject the log to some degree of preprocessing in order to reduce its overall complexity, make visual connections between the steps contained within, and aid in analyzing and optimizing the business concepts at hand.

> “As other researchers have noted, it is necessary to subject the log to some degree of preprocessing in order to reduce its overall complexity”



Type	Description
“A_” Application Events	Refers to states of the application itself. After a customer initiates an application, bank resources follow up to complete the application where needed and facilitate decisions on applications.
“O_” Offer Events	Refers to states of an offer communicated to the customer.
“W_” Work Events	Refers to states of work items that occur during the approval process. These events capture most of the manual effort exerted by Bank’s resources during the application approval process. The events describe efforts during various stages of the application process. - *W_Afhandelen leads:* Following up on incomplete initial submissions - *W_Completeren aanvraag:* Completing pre-accepted applications - *W_Nabellen offertes:* Follow up after transmitting offers to qualified applicants - *W_Valideren aanvraag:* Assessing the application - *W_Nabellen incomplete dossiers:* Seeking additional information during assessment phase - *W_Beoordelen fraude:* Investigating suspect fraud cases *W_Wijzigen contractgegevens:* Modifying approved contracts

Table 1. Event Names and Descriptions.

Although we were provided a rigorously preprocessed event log that could be analyzed in process mining tools quiet readily, we processed the data further to build tailored extracts for various analytical purposes.

2.2. Tools Used for Analysis

■ *Disco*: We procured an evaluation version of Disco 1.0.0 (Fluxicon) and used it in the exportation of data into formats suitable for spreadsheet analysis. Disco was especially helpful in facilitating visualization of typical process flows and exceptions.

■ *Microsoft Excel*: We used Excel 2010 (Microsoft) to foster deeper exploration into the preprocessed data. Excel was especially helpful for performing basic and advanced mathematical functions and data sorting, two capabilities notably absent from the Disco application.

■ *CART*: We used an evaluation version of the CART implementation (Salford Systems) for conducting preliminary segmentation analysis of the loan applications to assess opportunities for prioritizing work effort.

3. Understanding the Process in Detail

3.1. Simplifying the Event Log

Upon obtaining the BPIC 2012 event log, we first attempted to reduce its overall complexity by identifying and removing redundant events. For the purposes of this analysis, an event is considered *redundant* if it occurs concurrently with or subsequently after another event, such that the time between the two events is minimal (a few seconds at most) with respect to the time frame of the case as a whole.

Initial analysis of the raw data in *Disco* revealed a total of 4,366 event order variants among the 13,087 cases represented. We surmised that removal of even one sequence of redundant events could result in a

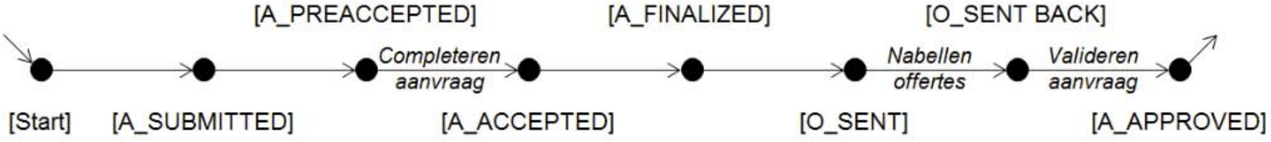
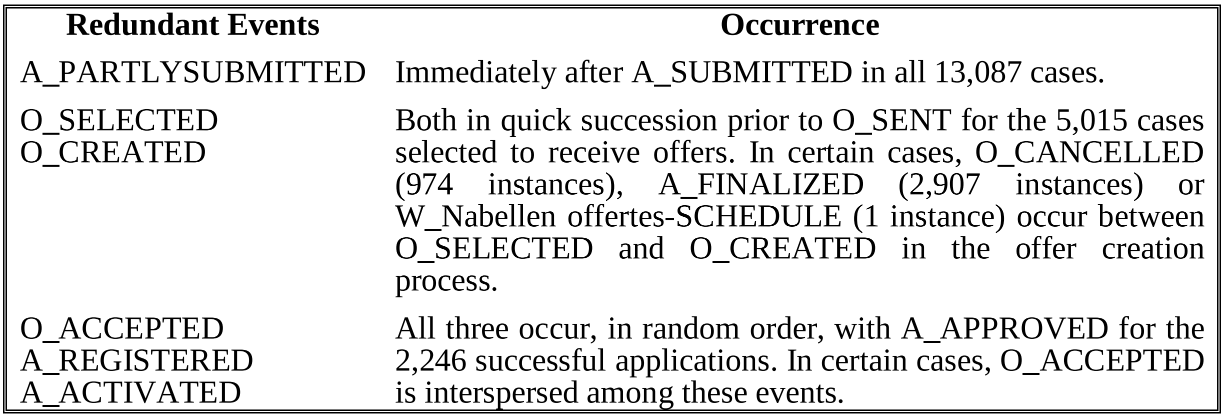


Figure 1. Standardized Case Flow for Approved Applications.



Redundant Events	Occurrence
A_PARTLYSUBMITTED	Immediately after A_SUBMITTED in all 13,087 cases.
O_SELECTED O_CREATED	Both in quick succession prior to O_SENT for the 5,015 cases selected to receive offers. In certain cases, O_CANCELLED (974 instances), A_FINALIZED (2,907 instances) or W_Nabellen offertes-SCHEDULE (1 instance) occur between O_SELECTED and O_CREATED in the offer creation process.
O_ACCEPTED A_REGISTERED A_ACTIVATED	All three occur, in random order, with A_APPROVED for the 2,246 successful applications. In certain cases, O_ACCEPTED is interspersed among these events.

Table 2. Potential Redundancies in the Event Log.

significant reduction in the number of variants. This simplification is compounded further when the number of removed variants is multiplied by others occurring downstream of the initial event.

Additionally, we eliminated two O-type events (O_CANCELLED and O_DECLINED) which occur simultaneously with A_ CANCELLED and A_DECLINED, respectively. W-type events were not considered for removal, as their transition phases are crucial for calculating work time spent per case. With the redundant events removed from the event log, the number of variants was reduced to 3,346 – an improvement from the unfiltered data set of nearly 25%. Such consolidation can aid in simplifying the process data and facilitating quicker analysis. The variant complexity could be further reduced by interviewing process experts at the bank to help consolidate events that occur together and sequencing variations not critical for business analysis.

3.2. Determining Standard Case Flow

We next sought to determine the standard case flow for a successful application, against which all other cases could then be compared. We did this by loading the simplified project into Disco and filtering all cases for the attribute A_APPROVED. We then set both the activities and paths thresholds to the most rigorous level (0%), which resulted in an idealized depiction of the path from submission to approval (see **Figure** 1).

3.3. Understanding Application Outcomes

Before launching into a more detailed review of the data, we found it necessary to define endpoint outcomes for all 13,087 applications. Using the standardized case flow (see **Figure** 1), we determined that all applications are subject to one of four *fates* at each stage of the approvals process:

■ *Advancement to next stage*: The application proceeds to the next stage of the process.

■ *Approved*: Applications that are approved and where the customer has accepted the bank's offer are considered a success and are tagged as Approved, with the end point depicted by the event A_APPROVED.

■ *Cancelled*: The application is cancelled by the bank or at the request of the customer. Cancelled applications have a final endpoint of A_CANCELLED.

■ *Denied*: The applicant, after having been subject to review, is deemed unfit to receive the requested loan or overdraft. Denied applications have a final endpoint of A_DECLINED.

We leveraged Disco's filtering algorithm to define a set of possible endpoint behaviors. 399 cases were classified *unresolved* as they were in progress at the time the data was collected (i.e., did not contain endpoints of A_DECLINED, A_CANCELLED or A_APPROVED).

Figure 2 shows a high-level process flow that marks how the cases are disposed at each of the key process steps. This analysis provides us useful insights on the overall business impact of this process as well as overall case flow through critical process steps.

We observe several baseline performance characteristics from **Figure** 2:

■ ~26% of applications are instantly declined (3,429 out of 13,087); indicating tight screening criteria for moving an application beyond the starting point.

■ ~24% of the remaining (2,290 out of 9,658) are declined after initial lead follow up, indicating a continuous risk selection process at play.

■ 754 of the 3,254 applications that go to validation stage (~23%) are declined, indicating possibilities for tightening upfront scrutiny at application or offer stages.

4. Assessing Process Performance

4.1. Case-Level Analysis

4.1.1. Case Endpoint vs. Overall Duration

In an effort to evaluate how the fate of a particular case changes with overall duration, we prepared a plot of these two variables and overlaid upon it the cumulative amount of work time amassed over the life of these cases. We excluded 3,429 cases that are instantly declined on initial application submission, as no effort is spent on these. We endeavored to visualize the point at which exertion of additional effort yields minimal or no return in the form of completed (closed) applications.

Figure 3 shows a lifecycle view of all applications, indexed to the time of submission. As shown in the figure, within the first seven days applications continue to move forward or are declined. At Day 7, the number of approved cases begins to rise, suggesting this is the minimal number of days required to fulfill the steps in the standard case flow (see **Figure** 1).

Approvals continue until ~Day 23, at which point >80% of all cases that are eventually approved have been closed and registered. There is a significant jump in the number of cancelled applications at Day 30, as inactive cases receiving no response from the applicant after stalling in the bottleneck stages *Completeren aanvraag* or *Nabellen offertes* are cancelled, likely according to bank policies.

This raises the interesting question of when the bank should stop any proactive efforts to convert an application to a loan, and whether the bank should treat customers differently based on behaviors that indicate likelihood of eventual approval. For example, the bank exerts an additional 380+ person days of effort between Days 23 and 31, only to cancel a majority of pending cases at the conclusion of this period. With additional data on customer profitability or lifetime value and comparative cost of additional effort, one can determine an optimal point

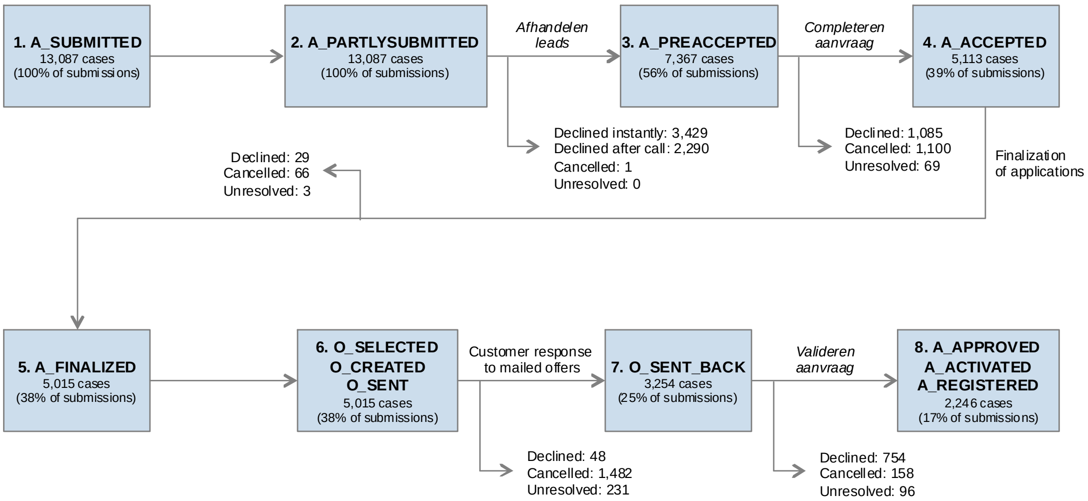


Figure 2. Key Process Steps and Application Volume Flow.

in the process where additional effort on cases that have not reached a certain stage carries no positive value.

4.1.2. Segmenting Cases by Amount Requested

As each case is associated with an amount requested by the applicant, we found it appropriate to arrange them into segments of roughly equal number, sorted by total requested value. We first removed the instantly declined cases by filtering them through Disco, as these are immediately resolved upon submission and do not have any additional effort or steps in the process. The resultant 9,658 cases (which include those in progress) were then split into deciles of 965-966 cases each. Each decile was further segmented by classifying the cases according to eventual outcome, and the ensuing trends were examined for correlation of approval percentage with amounts requested (see Figure 4).

We immediately observed the highest approval percentages in deciles 3 and 6, whose cases contained request ranges of

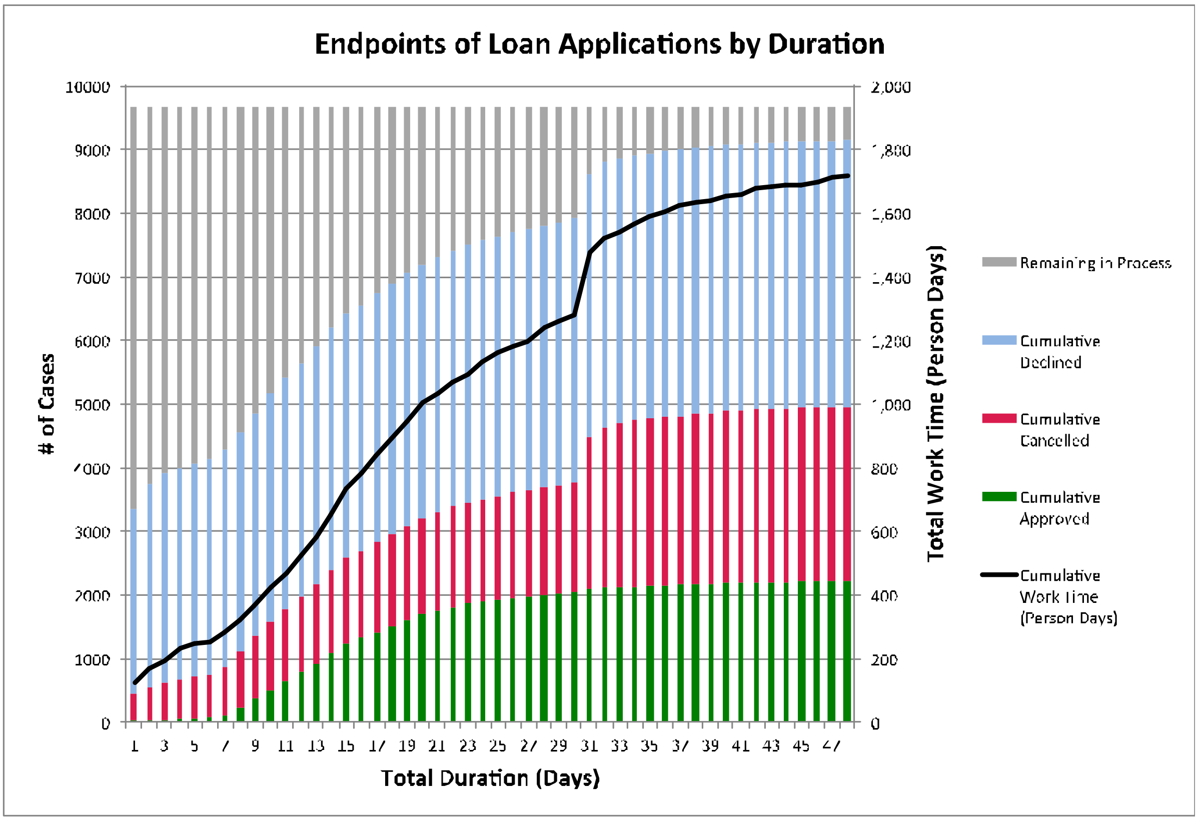


Figure 3. Distribution of Cases by Eventual Outcome and Duration, with Cumulative Work Effort. *Gray:* Remaining In Progress, *Blue:* Cumulative Declined, *Red:* Cumulative Cancelled, *Green:* Cumulative Approved. *Excludes 3,472 Instantly Declined Cases.*

> "These results suggest that an office of specialists performing single activities may be better suited to handle a larger amount of cases than an army of resources charged with a myriad of tasks"

5,000-6,000 and 10,000-14,000, respectively. The exact reason for this pattern is unclear; however, we speculate that typical applicants will often choose a "round" number upon which to base their requests (indeed, this is reflected in the three most frequent request values in the data set: 5,000, 10,000 and 15,000). Perhaps a certain risk threshold change in the bank's approval process causes a step change in approval percentages.

4.2. Event-Level Analysis

4.2.1. Calculating Event Duration

We sought to gain a detailed understanding of the work activities embedded in the approvals process, specifically those that contribute a significant amount of time or resources toward resolution. The format of data made available in this case was not readily amenable to this analysis.

We used Excel to manipulate the event-level data as provided and defined work time (presumably actual effort expended by human resources) for each event as the duration from start to finish (START / COMPLETE transitions, respectively). In contrast, wait time was defined as the latency between event scheduling and commencement (SCHEDULE / START), or the time elapsed between two instances of a single activity type as well as between COMPLETE of one event and START of another:

As shown in **Table 3**, two activities, *Completeren aanvraag* and *Nabellen Offertes*, contribute a significant amount to the total case time represented in the event log. The accumulated wait time attributed to each of these two events can reach as high as 30+ days per case, as the bank presumably makes numerous attempts to reach the applicant until contact is made.

On closer inspection of the data, we realized that the bank attempts to contact the customer multiple times per day until Day 30 in order to complete the application, as well as to follow up on offers that have been extended but not yet replied to.

4.2.2. Initial vs. Follow-Up Activities

The average work time spent performing each event changes whether the bank is conducting it for the first time, or following up on a previous step in a particular case (see **Figure 5**).

Some differences in initial and follow-up instances are minimal (*Valideren aanvraag),* while others are more pronounced (*Beoordelen fraude*). In the case of *Valideren aanvraag*, the bank is likely to be as thorough as possible during the validation process, regardless of how many times it has previously viewed an application. On the other hand, when investigating suspect cases for fraud, the bank may already have come to a preliminary conclusion regarding the application and is merely using the follow-up instance to justify its decision.

Follow-up instances for those events in which the bank must contact the applicant often have smaller average work times than their initial counterparts, as these activities are those most likely to become trapped in repeating loops, perhaps due to non-responsive customers. One can leverage such event data to understand customer behavior

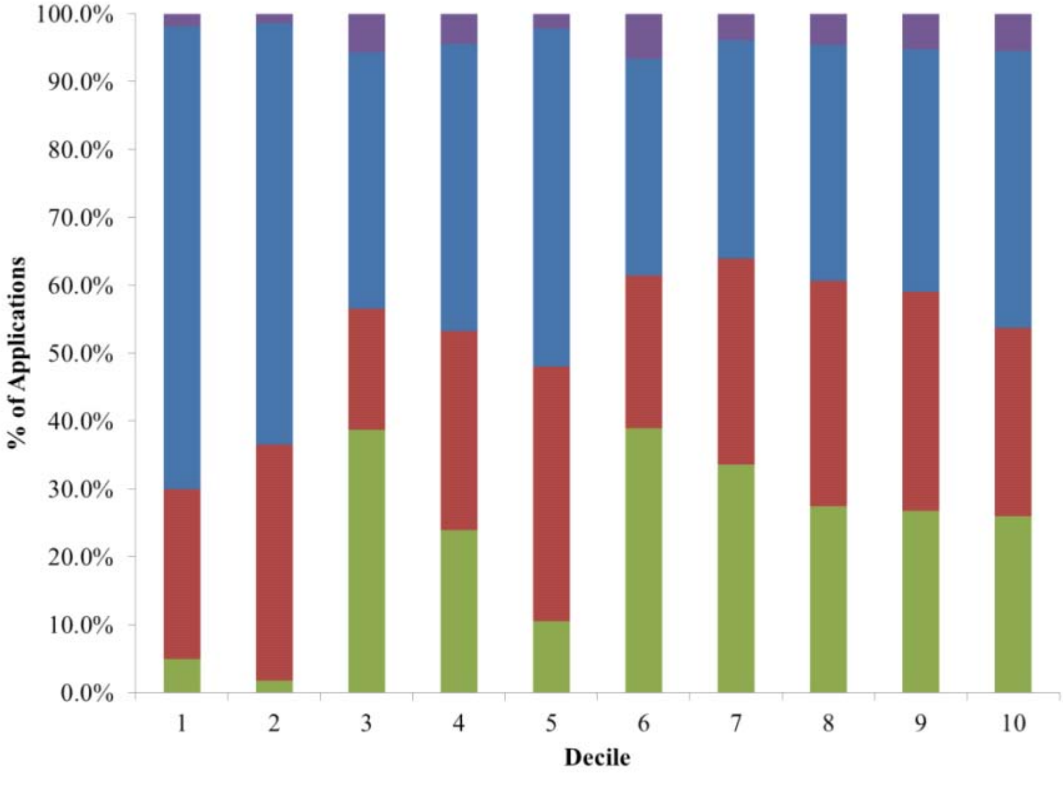


Figure 4. Endpoints of Cases (Left Axis), Segmented by Amounts Requested by the Applicant. *Green:* Approved, *Red:* Cancelled, *Blue:* Declined, *Violet:* In Progress.

> "These results suggest that an office of specialists performing single activities may be better suited to handle a larger amount of cases than an army of resources charged with a myriad of tasks"

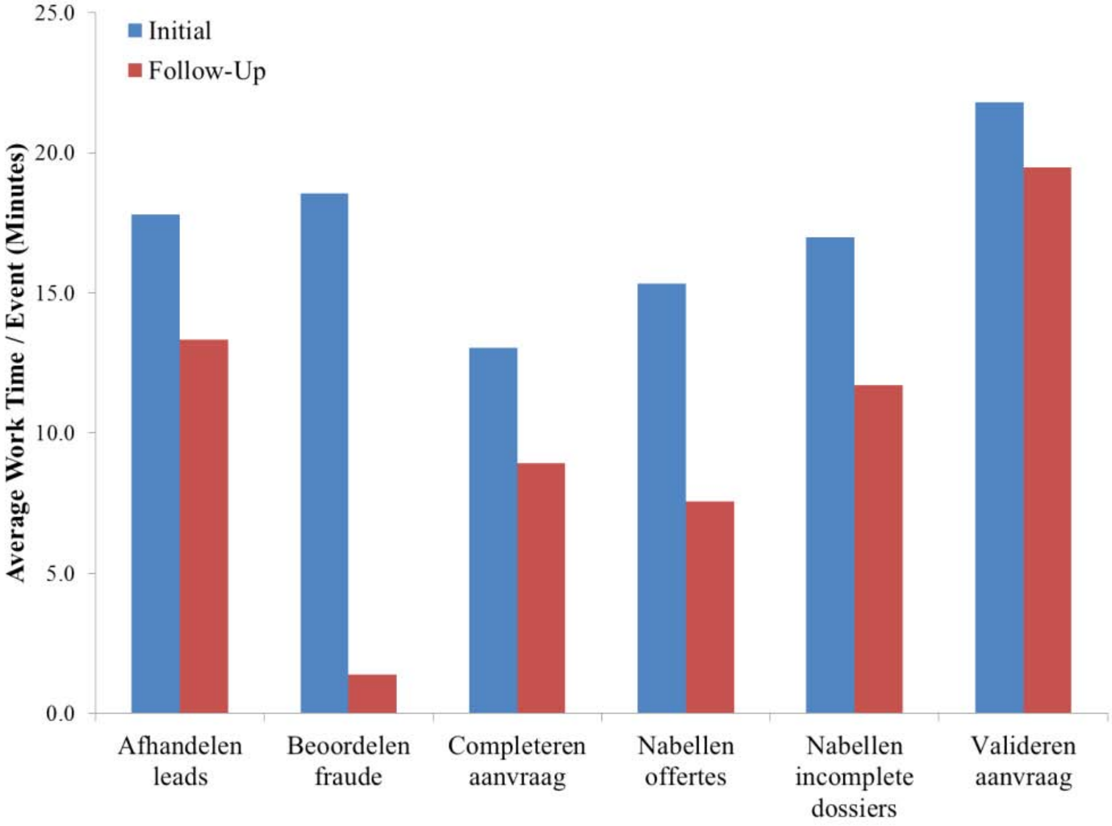


Figure 5. Comparison of Average Work Times, Initial vs. Follow-Up Event Instances.

and assess potential usefulness of such data for work prioritization.

4.3. Resource-Level Analysis

4.3.1. Specialist vs. Generalist-Driven Work Activities

We profiled 48 resources that handled at least 100 total events (excluding resource 112, as this resource does not handle work events outside of scheduling) and computed work volume by number of events handled by each. We observed nine resources that spent >50% of their effort on *Valideren aanvraag*, and a distinct group that mostly performed activities of *Completeren aanvraag, Nabellen offertes* and *Nabellen incomplete dossiers*. It appears validation is performed by a dedicated team of specialists focused on this work type, while customer-facing activities such as *Completeren aanvraag, Nabellen offertes* and *Nabellen incomplete dossiers* might require similar skills that are performed by another specialized group.

We next examined the performance of resources identified as specialists (>50% of work events of one single type) or contributors (25-50%) and compared them with those who played only minor roles in similar activities. To do this, we took the total work time accumulated in an activity by resources belonging to a particular category and calculated averages based on the total number of work events performed in that category. Two activities, *Nabellen offertes* and *Valideren aanvraag*, did not contain specialists and contributors, respectively, and so these categories were omitted from the comparisons for these activities.

As depicted in **Figure 6**, specialists spent less time per event instance than their counterparts, in some cases performing tasks up to 80% more efficiently than minor players. The performance of contributors is far less consistent, however, exhibiting average work times / case that are both higher (*Afhandelen leads, Nabellen offertes*) and lower (*Completeren aanvraag, Nabellen incomplete dossiers*) than those of the minor players. These results suggest that an office of specialists performing single activities may be better suited to handle a larger amount of cases than an army of resources charged with a myriad of tasks.

4.4. Leveraging Behavioral Data for Work Effort Prioritization

One of the objectives of process mining is to identify opportunities for driving process effectiveness; that is, achieving better business outcomes for the same or less effort in a shorter or equal time period. In particular, we sought to use process event data collected on an application to better prioritize work efforts. Specifically, we set out to understand if this could be done on the fifth day since the application was submitted.

To do this, we created an application-level data set for 5,255 cases that lasted >4 days and where the end outcome is known. For these applications, we captured all events from submission until the end of day 4 and used them to calculate the following:

- What stage the application had reached, and if it had been completed.
- How much effort had already gone into the application.
- How many events of each kind had already been logged.
- If the application required lead follow up.

We attempted to find key segments in this

	Afhandelen Leads	*Beoordelen Fraude*	*Completeren aanvraag*	*Nabellen Offertes*	*Nabellen Incomplete Dossiers*	*Valideren Aanvraag*
Work Time:						
Approved	13,659	23	45,909	68,473	89,204	121,099
Cancelled	14,601	2	119,497	94,601	25,633	7,775
Declined	67,560	2,471	63,052	30,870	26,993	29,946
Wait Time:						
Approved	198,916	8,456	1,873,537	34,972,224	5,980,887	10,537,938
Cancelled	300,062	28,763	16,582,465	42,630,195	2,006,774	678,105
Declined	986,421	236,115	3,294,367	13,542,054	1,001,354	3,227,252

Table 4. Potential Time Savings Associated with Conversion of Current Generalists to Single-Activity Specialists. (*) None of the resources performing *Nabellen offertes* were identified as specialists; therefore mean efficiency for area contributors was used instead.

population that were highly likely to be approved and accepted OR highly likely to be cancelled or declined. We did this by subjecting the data to segmentation using the Classification and Regression Tree (CART) technique (see **Figure 7**).

The partial tree above shows two segments with <6% approval rates: Terminal Nodes 1 and 14, consisting of a total of 1,018 cases with only 49 eventual approvals. Node 14, consisting of 818 cases, shows incomplete applications where the bank could not prepare an offer for the customers by the end of Day 4. Such "slow-moving" applications had a <6% chance of being approved, compared to an average of 42% for the entire group of 5,255. Node 1 has applications that are touched by 3 or fewer resources; with 112 being one of them. This might be another indicator for a slow-moving application. Such applications have virtually no likelihood of being approved in the end.

One could repeat this analysis at different stages in the lifecycle of the application to help with effort prioritization. This preliminary analysis indicates significant potential to reduce effort on cases that might not reach the desired end state. Further analysis with customer demographics, application details, and additional information on resources who work on such cases will help refine the findings and suggest specific action steps to improve process effectiveness.

5. Discussion

5.1. Working with Data Challenges

5.1.1. Managing Event Complexity

The optimization of the loan approvals process is an exercise in streamlining each step of the end-to-end operation. One notable point that creates challenges in building a streamlined process view with automated process mining tools is the amount and complexity of data captured. If such data is not used with accompanying business judgment, one can get lost in apparent complexity (>4,000 process variants for a process that has 6-7 key steps). We illustrated this point above in our discussion regarding redundant events. We recommend dealing with such complexities at the time of analysis, using process knowledge and good business judgment, by performing additional data pre-processing steps.

It is also critical to scrutinize event data up front to understand all quirks and to build ways of addressing these. For example, a comparison of the number of START and COMPLETE transitions for W-type events in the data set reveals the existence of 1,037 more COMPLETE transitions than START transitions. As the time stamps for these events are unique with respect to others in the same Case ID, they have the potential to greatly confuse the summation of work and wait times for a particular case and for resources within the institution. We denoted these as systems errors and worked with the first COMPLETE following a START as the "correct" one for a given work event type. In a real project, we would validate our assumption by deeper review of how such instances arise in the system and using that understanding to treat these observations correctly in our analysis.

As described in **Section 3.1**, the event log would also benefit from consolidation of events that happen concurrently, such as those that occur when successful applications are approved (A_APPROVED, A_REGISTERED and A_ACTIVATED). This would not only decrease the overall file size (which becomes important as the volume of data grows), but also reduce the complexity of the initial log.

5.2. Potential Benefits of Resource

5.2.1. Re-Deployment Recasting General ists as Specialists

As mentioned previously, the tasks involved in the loan approvals process are performed by a mixture of specialists and generalists. Through our analysis we concluded that the bank might benefit from specialization of labor, whereby current resources are reassigned to single posts in order to maximize efficiency. In **Table 4**, we show potential gains to be made through such restructuring. If the bank can improve performance of everyone executing a task to the same levels as specialists, we estimate a substantial overall time saving.

We also evaluated the potential savings associated with downsizing the overall pool of resources assigned to these tasks. Using the average amount of work time for resources handling >100 total events (approximately 16,000 minutes; again excluding resource 112), we estimate opportunity to reduce the work effort by 35%:

5.3. The Power of Additional Information Additional

5.3.1. Case-Level Attributes

In its raw form, the BPIC 2012 event log is a gold mine of information that, once decoded, provides a detailed view of a consumer loan approvals process. However, this information would be greatly strengthened by the addition of a few key data points. As each case carries with it a single attribute – the amount requested

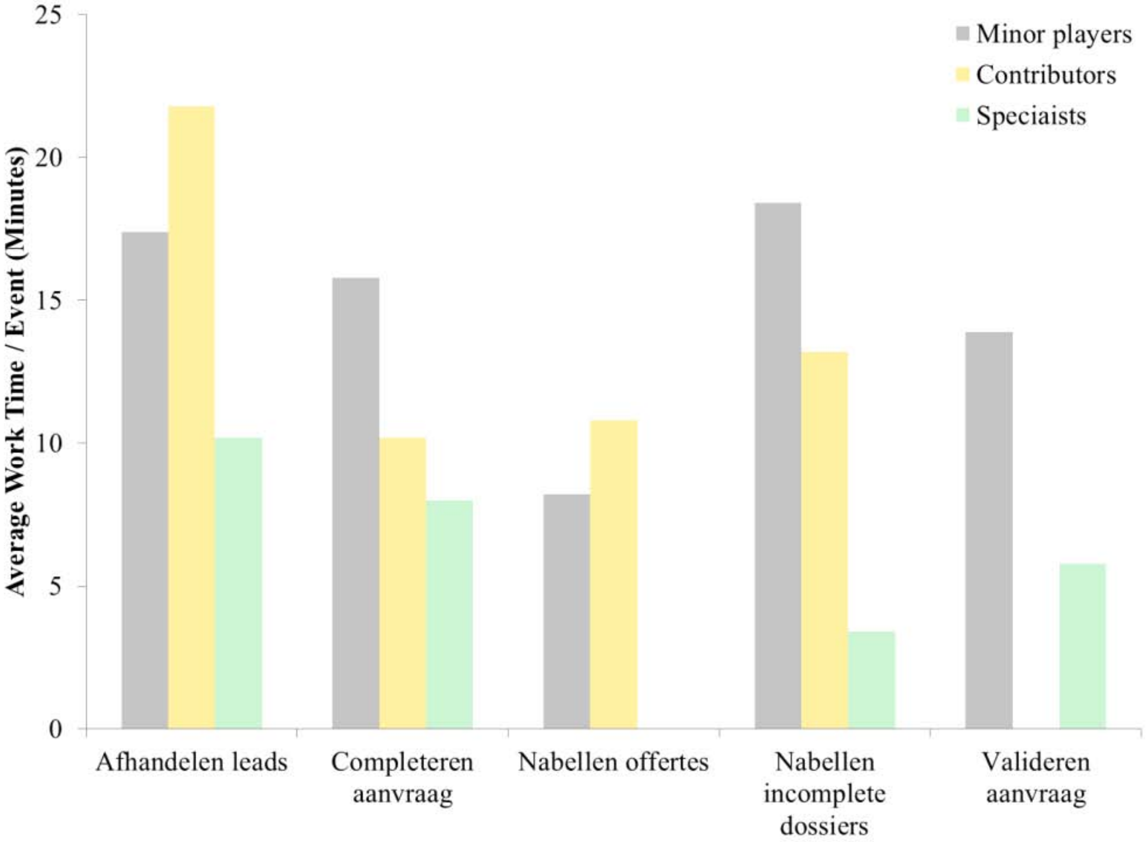


Figure 6. Work Time per Event, Specialists / Contributors vs. Minor Players.

by the applicant – we have no way of knowing why certain cases are approved while others with identical request amounts and paths are rejected. Therefore it would be useful to know customer demographics, current or past relationships with the customers, and additional details about the resources that execute these processes. With this information, we can build specific recommendations for changing the process and more accurately estimate likely benefits of such changes.

5.3.2. Customer Profitability and Operating Costs

A final set of data notably absent from the provided BPIC 2012 log are the overall costs associated with the loan approvals process and value of each loan application to the bank. It would be worthwhile to understand how much it costs to operate each resource, and whether this cost varies based on the activities they perform or the number of events they participate in. This information would also allow us to calculate an average acquisition cost for each applicant, and subsequently understand the minimum threshold below which it does not make economic sense to approve an incoming loan request.

6. Conclusions

Through comprehensive analysis of the BPIC 2012 event log, we converted a fairly complex data set into a clearly interpretable, end-to-end workflow for a loan and overdraft approvals process. We examined the data at multiple levels of granularity, uncovering interesting insights at all levels. Through our work we uncovered potential improvements in a number of areas, including revision of automated processes, restructuring of key resources, and evaluation of current case handling procedures. Indeed, future analysis would be greatly aided by the inclusion of additional data, such as customer information, governing policies, operating costs and relative customer value.

As part of our analysis, we performed a rudimentary predictive exercise whereby we determined the current status of cases at various days in the approvals process and quantified their chances of approval, cancellation, or denial. This allowed us to estimate the fate of a case based on its performance and tailor the overall process to minimize stalling at traditional case bottlenecks. While preliminary in its nature, this opens the door to more elaborate future modeling exercises, perhaps driven by sophisticated computer algorithms.

While we covered several areas in this exercise, there are others where we did not conduct detailed analysis. The bank would find significant additional benefits from exploring such additional areas, for example, social network analysis.

In conclusion, the procedures highlighted by the BPIC 2012 elaborate the role and importance of process mining in the modern workplace. Steps that were previously elucidated only after years of practice and observation can now be examined using a sample set of existing data. As the era of Big Data continues its march toward the business world, we foresee process mining as a central player in the charge toward turning questions into solutions and problems into sustainable profit.

Acknowledgements

We are grateful to the financial institution that made this data available for study. Special thanks to Fluxicon for providing us with an evaluation copy of Disco and an accompanying copy of the BPIC 2012 data set. We also thank Tom Metzger, Nicholas Hartman, Rolf Thrane and Pierre Buhler for helpful discussions and insights. Our thanks also to Salford Systems, who made their software available in a demonstration version.

> “It is also critical to scrutinize event data up front to understand all quirks and to build ways of addressing these”

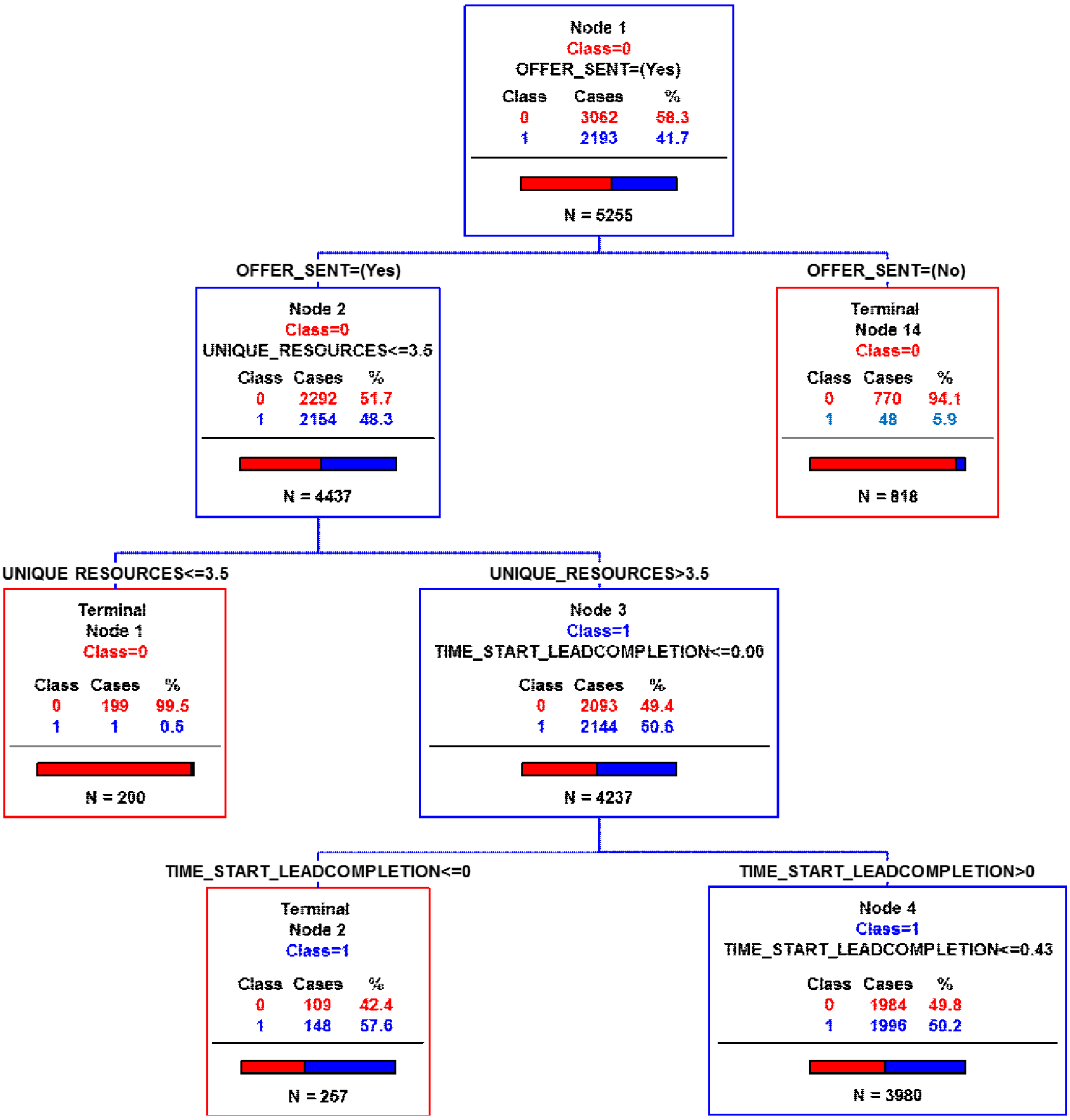


Figure 7. Partial View, CART Segmentation Tree.

References

[1] **W. Van der Aalst, A. Adriansyah, A.K. Alves de Medeiros, F. Arcieri, T. Baier *et al.*** Process Mining Manifesto. *Business Process Management Workshops 2011, Lecture Notes in Business Information Processing, vol. 99*. Springer-Verlag, 2011.

[2] **J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, A. Byers.** Big Data: The Next Frontier for Innovation, Competition, and Productivity. McKinsey Global Institute, 2011. <http://www.mckinsey.com/insights/business_ technology/big_data_the_next_frontier_for_innovation>.

[3] **R. Adduci, D. Blue, G. Chiarello, J. Chickering, D. Mavroyiannis *et al.*** *Big Data: Big Opportunities to Create Business Value*. Technical report, Information Intelligence Group, EMC Corporation, 2011.

[4] **R.P.J.C. Bose, W.M.P. van der Aalst.** Analysis of Patient Treatment Procedures: The BPI Challenge Case Study. *First International Business Process Intelligence Challenge*, 2011. <http://bpmcenter.org/wp-content/uploads/reports/2011/BPM-11-18.pdf>.

[5] **W.M.P. van der Aalst.** *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer, 2011. ISBN-10: 3642193447.

Daniela Luengo, Marcos Sepúlveda
Computer Science Department, School of Engineering, Pontificia Universidad Católica de Chile, Santiago (Chile)

<dlluengo@uc.cl>,<marcos@ing.puc.cl>

Detection of Temporal Changes in Business Processes Using Clustering Techniques

Abstract: *Nowadays, organizations need to be constantly evolving in order to adjust to the needs of their business environment. These changes are reflected in their business processes, for example: due to seasonal changes, a supermarket's demand will vary greatly during different months of the year, which means product supply and/or re-stocking needs will be different during different times of the year. One way to analyze a process in depth and understand how it is really executed in practice over time, is on the basis of an analysis of past event logs stored in information systems, known as process mining. However, currently most of the techniques that exist to analyze and improve processes assume that process logs are in a steady state, in other words, that the processes do not change over time, which in practice is quite unrealistic given the dynamic nature of organizations. This document presents in detail the proposed technique and a set of experiments that reflect how our proposal delivers better results than existing clustering techniques.*

Keywords: *Concept Drift, Clustering, Process Mining, Temporal Dimension.*

Authors

Daniela Luengo was born in Santiago, Chile. She is an Industrial Civil Engineer with a major in Information Technology from Pontificia Universidad Católica de Chile. She also received the academic degree of Master of Science in Engineering from the same university. Additionally she has an academic certificate in Physical Activity, Sport, Health and Education. From 2009 to 2013 she worked at the Information Technology Research Center of the Pontificia Universidad Católica de Chile (CETIUC), as an analyst and consultant in the area of process management excellence, performing process mapping, process improvement and several researches. Her current interests are focused on applying her knowledge in the public sector of her country.

Marcos Sepúlveda was born in Santiago, Chile. He received his Ph.D. on Computer Science from the Pontificia Universidad Católica de Chile in 1995. He made a postdoctoral research in the ETH Zürich, Switzerland. He is an associate professor in the Computer Science Department at the Pontificia Universidad Católica de Chile since 2001. He is also the director of the Information Technology Research Center of his university (CETIUC). His research interests are Process Mining, Business Process Modeling, Business Intelligence, and Information Systems Management.

1. Introduction

In a globalized and hyper-connected world, organizations need to be constantly changing in order to adapt to the needs of their business environment, which implies that their business processes must also be constantly changing.

To illustrate this, consider the case of a toy store whose sales practices may change radically depending on whether they are executed over the Christmas period or during vacations, principally due to the changes in the volume of the demand. For the Christmas season, they might employ a process that prioritizes efficiency and volume (*throughput*), while during the vacation season they might use a process which focuses on the quality of their customer service.

In this example, it is easy to identify the periods in which demand changes, and thus it is possible that the process manager (the person in charge of the process management) will have a clear understanding of the changes that the sales process undergoes over time. However, if an organization has a process that occurs in various independent offices, for example, offices that are located in different geographic locations, the evolution of the changes in the process at each office will not be so evident to the central process manager. Moreover, the changes in each office could be different and could happen at different moments in time.

Understanding the changes that are occurring in the different offices could help to better understand how to improve the overall design of the process. Being able to understand these changes and model the different versions of the process allow the process manager to have more accurate and complete information in order to make coherent decisions which result in better service or efficiency.

To achieve the aforementioned goals, various advances have been made in the discipline of Business Process Management (BPM), a discipline which combines knowledge about information technology and management techniques, which are then applied to business operation processes, with the goal of improving efficiency [1]. Within BPM, process mining has positioned itself as an emerging discipline, providing a set of tools that help to analyze and improve business processes [1], based on analyzing event logs stored by information systems during the execution of a process. However, despite the advances made in this field, there still exists a great challenge, which consists of incorporating the fact that processes change over time, a concept which is known in the literature as *Concept Drift* [2].

Depending on the nature of the change, it is possible to distinguish different types of *Concept Drift*, including: *Sudden Drift* (sudden and significant change to the definition of the process), *Gradual Drift* (gradual change to the definition of the process, allowing for the simultaneous existence of the two definitions), and *Incremental Drift* (the evolution of the process that occurs through small, consecutive changes to the definition of the model).

Despite the existence of all these *Drift* types, the existing techniques of process mining are limited to finding the points in time when the process changes, centering principally on *Sudden Drift* changes. The problem with this limitation is that in practice it is not as frequent for business processes to show a sudden change in definition.

If we apply the existing process mining approaches to processes that have different kinds of changes other than *Sudden Drift*, we could end up with results which make little sense to the business.

In this document we propose a new approach, which allows the discovery of the various versions of a process when there are different kinds of *Drift,* helping to understand how the process behaves over time. To accomplish this task, existing process mining clustering techniques are used, but with time incorporated as an additional factor to the

> "Currently, one of the problems in process mining is that the developed algorithms assume the existence of information relative to a unique version of the process in the event log"

control-flow perspective to generate the different *clusters. Trace Clustering* techniques are used, which unlike other metrics-based techniques that measure the distance between complete sequences having linear complexity, allowing for the delivery of results in a shorter time span [3].

The focus of our work contributes to the process analysis, allowing the process manager to have a more realistic vision of how the process behaves over different periods of time. With this approach it is possible to determine the different versions of the process, the characteristics of each process, and to identify in which moment the changes occur.

This article is organized in the following way. **Section** 2 presents the related work. **Section** 3 describes base and the modified version of the clustering method. **Section** 4 presents experiments and results and finally the conclusions and future work are presented in **section** 5.

2. Related Work

Process mining is a discipline that has attracted major interest recently. This discipline assumes that the historical information about a process stored in information systems can be found in a dataset, known as an event log [4]. This event log contains past information about the activities that were performed in each step of the process, where each row of the log is composed of at least one identifier (id) associated with each individual execution of the process, the name of the activity performed, the timestamp (day and time when the activity occurred), and, optionally, additional information like the person who carried out the activity or other such information. Additionally, in the literature [3] an ordered list of activities invoked by a specific execution of the process is defined as a trace of the execution.

Currently, one of the problems in process mining is that the developed algorithms assume the existence of information relative to a unique version of the process in the event log. However, this is often not the case, which is why applying the process mining algorithms to these logs leads to fairly unrepresentative and/or very complicated results, which contribute little to the job of analyzing and improving the processes.

2.1. Clustering of the Event Log

To resolve the aforementioned problems in process mining, clustering techniques have been proposed for dividing the event log before the process mining techniques are applied [5]. This would consist of dividing the event log into homogenous clusters; in order to later apply the process mining techniques independently to each cluster and thus obtaining more representative models. **Figure** 1 shows the stage of log preprocessing and then uses a discovery technique as an example of a process mining technique.

To produce this clustering, it is necessary to define a way of representing the traces, so that it becomes possible to group them later according to previously determined criteria of similarity. There currently exist various clustering techniques in process mining [6][3]. The majority of them principally consider information about the control-flow of activities. These techniques can be classified into two categories:

1) Techniques that transform the traces into a vector space, in which each trace is converted into a vector. The dividing of the log can be done using a variety of clustering techniques in the vector space, like for example: *Bag of activities, K-gram model* [6], and *Trace clustering* [5]. However, these techniques have the problem of lacking contextual information, which some have attempted to correct with the *Trace clustering based on conserved patterns* technique [3].

2) Techniques that operate with the whole trace. These techniques use metrics of distance like *Levenshtein* and *Generic Edit Distance* [6], together with standard clustering techniques, assigning a cost to the difference between traces.

The existing techniques for both categories, despite improving clustering through the creation of structurally similar trace clusters,

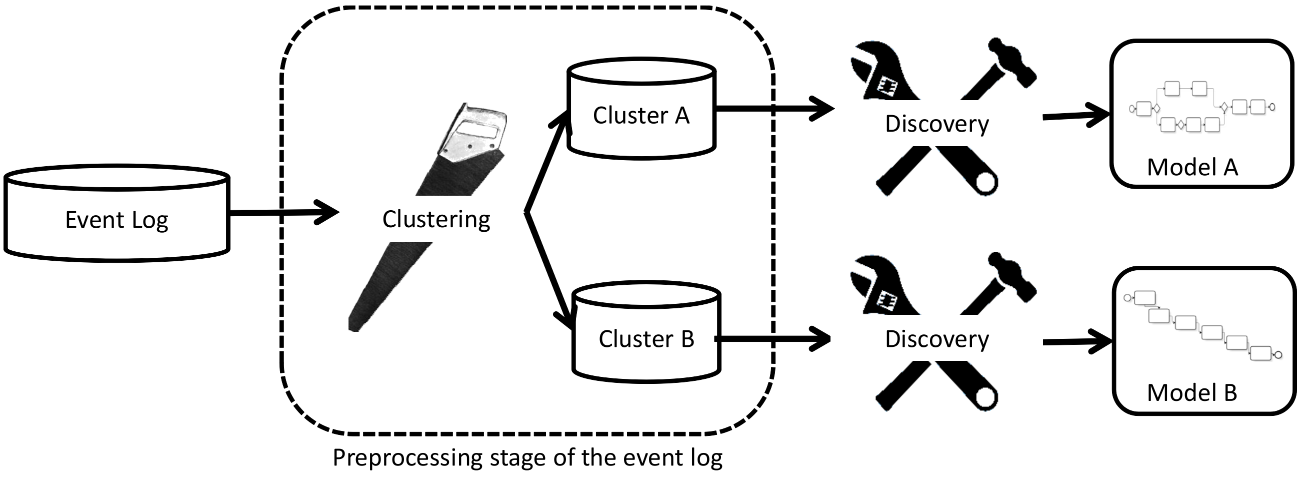


Figure 1. Preprocessing Stage of the Event Log.

> "The study of *Concept Drift* in the area of process mining has centered on process changes in the control-flow perspective, and can be of two kinds, momentary changes or permanent changes, depending on the duration of the change"

do not consider the temporal dimension of the process's execution, nor how the process changes over time.

2.2. The Concept Drift Challenge

In BPM, *Concept Drift* refers to a situation in which a process has experienced changes in its design within an analyzed period (yet the exact moment in which the changes were produced is unknown). These changes can be due to a variety of factors, but are mainly due to the dynamic nature of the processes [7].

The study of *Concept Drift* in the area of process mining has centered on process changes in the control-flow perspective, and can be of two kinds, momentary changes or permanent changes, depending on the duration of the change.

When changes occur over short and infrequent periods, they are considered momentary changes. These changes are also known in processes jargon as process noise or process anomalies.

On the other hand, permanent changes occur over more prolonged periods of time and/or when a considerable amount of instances is affected by the changes, which signals changes in the design process.

Our interest centers on the permanent changes in the control-flow perspective, which can be divided into the following four categories:

- Sudden Drift: This refers to drastic changes, meaning the way in which the process execution changes suddenly from one moment to the next.
- *Recurring Drift*: When a process suffers periodic changes, meaning a way of performing the process is repeated again later.
- *Gradual Drift*: This refers to changes which are not drastic, but rather at a moment when two versions of the process overlap, which corresponds to the transition from one version of the process to another.
- *Incremental Drift:* This is when a process has small incremental changes. These types of changes are more frequent in organizations that adopt agile BPM methodologies.

To solve the *Concept Drift* problem, new approaches have evolved to analyze the dynamic nature of the processes.

Bose [2] proposes methods to manage *Concept Drift* by showing how the process changes are indirectly reflected in the event log and that the detection of these changes is feasible by examining the relationship between activities. Different metrics have been defined to measure the relationship between activities. Based on these metrics, a statistical method was proposed whose basic idea is to consider a successive series of values and investigate if a significant difference between two series exists. If it does, this would correspond to a process change.

Stocker [8] also proposes a method to manage *Concept Drift* which considers the distances between pairs of activities of different traces as a structural feature, in order to generate chronologically subsequent *clusters.*

Bose and Stocker's approaches are limited to determining the moment in time when the process changes, and thus center on sudden changes and leave out other types of changes.

To resolve this, in an earlier article [9] we proposed an approach that makes use of clustering techniques to discover the changes that a process can experience over time, but without limiting ourselves to one particular kind of change. In that approach, the similarity among two traces is defined by the control-flow information and by the moment in which each trace begins to operate.

In this article, we present an extension of the earlier work [9], after incorporating a new form of measuring the distance between two traces.

3. Extending Clustering Techniques to Incorporate the Temporal Variable

As was explained in the last section, the existing approaches for dealing with *Concept Drift* are not sufficiently effective at finding the versions of a process when the process has undergone different types of changes. To solve this problem, we look to the *Trace Clustering* technique proposed by Bose [3] and based on conserved patterns, which allows clustering the event log considering each trace's sequence of activities.

Our work is based on this technique and extends it by incorporating an additional temporal variable to the other control-flow variables used for clustering.

3.1. *Trace Clustering* Based on Conserved Patterns

The basic idea proposed by Bose [3] is to consider subsequences of activities that repeat in multiple traces as feature sets for the implementation of clustering. Unlike the *K-gram* approach that considers subsequences of fixed size, in this approach the subsequences can be of different lengths. When two instances have a significant number of subsequences in common, it is assumed that they have structural similarity and these instances are assigned to the same cluster.

There are six types of subsequences, therefore, we will only give a formal definition of MR, since these are the subsequences that we used to develop our approach, however the work could be extended to use the other subsequences.

- Maximal Repeat (MR): A Maximal Repeat in a sequence T is defined as a subsequence that occurs in a Maximal Pair in T. Intuitively, an MR corresponds to a subsequence of activities that is repeated more than once in the log.

Table 1 shows an example where existing MR in a sequence are determined. This technique constructs a unique sequence starting from the event log, which is obtained by connecting all the traces, but with a delimiter placed among them. Then, the MR definition is applied to this unique sequence. The set of all MR discovered in the sequence with more than one activity, is called a *Feature set.*

Sequence	Maximal Repeat	Feature Set
bbbcd-bbbc-caa	{a, b, c, bb, bbbc}	{bb, bbbc}

Table 1. Example of *Maximal Repeat* and *Feature Set*.

“The basic idea proposed by Bose is to consider subsequences of activities that repeat in multiple traces as feature sets for the implementation of clustering”

Based on the *Feature Set,* a matrix is created that allows the calculation of the distance between the different traces. Each row of the matrix corresponds to a trace and each column corresponds to a feature of the *Feature Set.* The values of the matrix correspond to the number of times that each feature is found in the various traces (see **Table 2**). We will call this matrix the Structural Features Matrix.

This pattern-based clustering approach uses "*Agglomerative Hierarchical Clustering*" as a clustering technique, with the minimum variance criterion [15], and using the Euclidian distance to measure the difference between two traces, defined in as follows:

$$\text{dist}\,(A,B) = \sqrt{\sum_{i=1}^{n}(T_{Ai} - T_{Bi})^2}$$

Where

dist (A, B) = distance between trace A and trace B.
n = number of features in de Feature Set.
T_{Ai} = number of times de feature in the appears in the trace A.

3.2. Clustering Technique to Include the Temporal Variable

To identify the various types of changes that can occur in business processes we must find a way to identify all the versions of a process. If we only look at the structural features (control-flow) then we leave out information regarding temporality. Both temporal and structural features are very important since the structure indicates how similar one instance is to another and the temporality shows how close in time the two instances are. Our approach looks to identify the different forms of implementing the process using both features (structural and temporal) at the same time, as is illustrated in **Figure 2**.

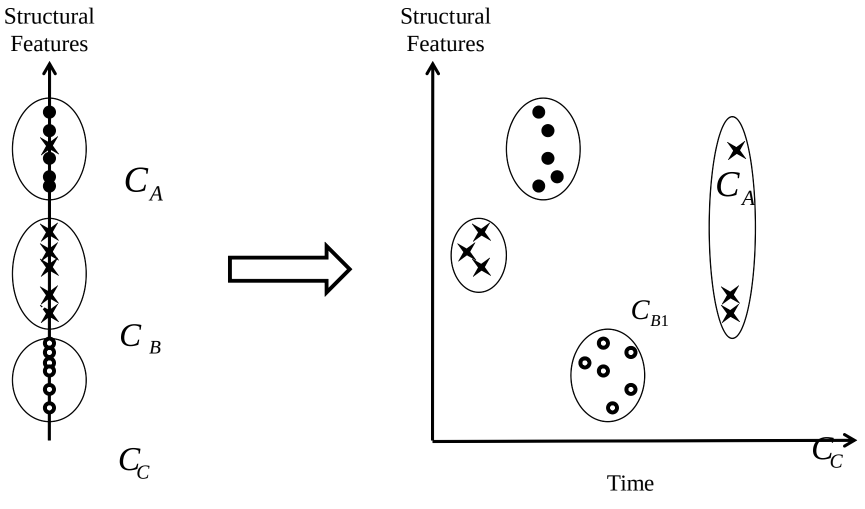


Figure 2. Example of the Relevance of Considering Time in the Analysis.

In order to mitigate the effects of external factors that are difficult to control, we use only the beginning of each process instance as the temporal variable.

For each trace, we store the time that have elapsed since a reference timestamp in the time dimension, for example, the number of days (or hours, minutes or seconds, depending on the process) elapsed since January 1st, 1970, to the timestamp in which the trace's first activity begins (see **Table 3**).

In this new approach, "*Agglomerative Hierarchical Clustering*" with the minimum variance criterion [15] is also used as a clustering technique.

To calculate the distance between two traces we use the Euclidian distance, but modified in order to consider at the same time the structural and temporal features.

First, we define T_{Ji} as the feature *i* of the trace *J*. If the feature *i* cannot be found in the trace *J*, its value will be 0, otherwise its value will be the number of times that the feature *i* appears in the trace *J*.

Trace \ Feature set	bb	bbbc
bbbcd	2	1
bbbc	2	1
caa	0	0

Table 2. Structural Features Matrix.

$T_{J(n+1)}$ corresponds to the temporal feature of the trace *J* and its value is the number of days (or hours, minutes or seconds, depending on the process) that have elapsed since a reference timestamp. The index *(n+1)* is given to indicate that it is to be added to the structural features.

We define *L* as the set of all the *log* traces, and the expression $Max_{J\in L}(T_{Ji})$ represents the largest number of times the feature *i* appears in an event *log* trace. In the same way, $Min_{J\in L}(T_{Ji})$ corresponds to the smallest number of times the feature *i* appears in an event *log* trace.

$Min_{J\in L}(T_{J(n+1)})$ and $Max_{J\in L}(T_{J(n+1)})$ correspond to the earliest and latest time in which an event log trace begin. Also we define $D_E(A,B)$ and $D_T(A,B)$ as the structural and temporal distance between the trace A and the trace B, respectively.

$$D_E(A,B) = \sqrt{\sum_{i=1}^{n}\left(\frac{T_{Ai} - T_{Bi}}{\max\limits_{J\in L}(T_{Ji}) - \min\limits_{J\in L}(T_{Ji})}\right)^2}$$

$$D_T(A,B) = \sqrt{\left(\frac{T_{A(n+1)} - T_{B(n+1)}}{\max\limits_{J\in L}(T_{J(n+1)}) - \min\limits_{J\in L}(T_{J(n+1)})}\right)^2}$$

where:

n = number of features from the Structural Features Set.

Even though both distances, D_E and D_T are normalized, since the domain of D_E is greater than the one of D_T, we define Min_E, Max_E, Min_T and Max_T as:

$$Min_E = \min_{A,B\in L} \sqrt{D_E(A,B)} \quad , \quad A \neq B$$

$$Max_E = \max_{A,B \in L} \sqrt{D_E(A,B)} \ , \quad A \neq B$$

$$Min_T = \min_{A,B \in L} \sqrt{D_T(A,B)} \ , \quad A \neq B$$

$$Max_T = \max_{A,B \in L} \sqrt{D_T(A,B)} \ , \quad A \neq B$$

$Min_E$and Max_E correspond to the minimum and maximum (normalized) distance between all traces, considering only structural features.

Min_T and Max_T correspond to the minimum and maximum (normalized) distance between all traces, considering only temporal features.

The new way of measuring the distance between two traces, $dist(A,B)$, incorporates the parameter μ, which we will call the weight of the temporal dimension, which serves to weigh the structural and temporal features. Additionally, this new way of measuring the distance adjusts D_E and D_T in such a way that the weight of both distances are equivalent.

$$dist(A,B) = (1-\mu)\frac{D_E(A,B) - Min_E}{Max_E - Min_E} + \mu\frac{D_T(A,B) - Min_T}{Max_T - Min_T}$$

The weight of the temporal dimension, μ, can have values between 0 and 1, according to the relevance given to the temporal feature.

4. Evaluation

We analyzed the proposed technique using six event logs obtained from different synthetic processes, which were created with CPN Tools [10][11]. In order to measure their performance we used the *Guide Tree Miner* plug-in [3] available in ProM 6.1 as well as a modified version of this plug-in that incorporates the proposed changes.

The evaluation was carried out using different metrics to measure the new approach's classification effectiveness versus the base approach.

4.1. Experiments and Results

Figure 3 shows the sequence of steps performed in the experiments.

1) To create the synthetic *log*, a simulation was used based on two designed models, M1 and M2, using CPN Tools.
2) The method starts by applying the clustering technique on the synthetic *log* received, which generates a given number of *clusters*. In this case, two *clusters* (C_A *and* C_B). Two clustering techniques are applied:

- Base approach: *Trace clustering* based on conserved patterns.
- **Our approach:** *Extended trace clustering*, where the temporal dimension is incorporated. In this technique, the weight of the temporal dimension,μ, can be given different values, which vary between 0 and 1.

3) For each of the clusters a discovery process is carried out, generating two new models (M_A *and* M_B).

The approach's performance can be measured at two points:
a) Conformance 1: The metrics are measured between any of the original models and one of the generated *clusters*.
b) Conformance 2: Metrics are measured between any of the original models and one of the generated models.

The metrics used to measure Conformance 1 are the following:

- *Accuracy*: Indicates the number of instances correctly classified in each *cluster*, according to what is known about the original events *log*. These values are between 0% and 100%, where 100% indicates that the clustering was exact.
- *Fitness*: Indicates how much of the observed behavior in an event *log*, (for example, *cluster* C_A) is captured by the original process model (for example, model M_2) [12]. These values are between 0 and 1, where 1 means the model is capable of representing all the *log* traces.
- Precision: Quantifies whether the original model allows for behavior completely unrelated to what is seen in the event *log*. These values are between 0 and 1, where 1 means that the model does not allow behavior additional to what the traces indicate [13].

The metrics used to measure Conformance 2 are the following [14]:

- *Behavioral Precision* (B_P)
- *Structural Precision* (S_P)
- *Behavioral Recall* (B_R)
- *Structural Recall* (S_R)

These metrics quantify the precision and generality of one model with respect to another. The values of these four metrics are between 0 and 1, where 1 is the best possible value.

Table 4 summarizes the results of applying the base approach and the new approach (varying the value of the parameter μ), to the different synthetic event *logs* created. This table shows the *accuracy* metric, which indicates the percentage of correctly classified traces.

For each *log*, the highest *accuracy* is reached with our approach, but with different μ values (varying between 0.2 and 0.9). The *accuracy* varies with different μ values because in each *log* the relevance of the temporal dimension versus the structure of the process is not the same.

We use Log (f) to make a more in-depth analysis of the results, measuring all the metrics defined both in Conformance 1 and Conformance 2 (see **Table** 5).

All metrics calculated for Log (f) show good results when the parameter μ has a value of 0.5 or 0.6. When the five metrics are averaged, the highest overall average is obtained with μ equal to 0.5.

5. Conclusions and Future Work

In this article we present the limitations of current clustering techniques for process mining, which center on grouping structurally similar executions of a process. By focusing just on the structure of the executions, the process's evolution over time (*Concept Drift*) is left out. New techniques have been developed to address this, but these also present limitations since they focus on finding the points in which the process changes, which is limited to just one kind of change, *Sudden Drift*.

We present an approach that extends current clustering techniques in order to find the different versions of a process that changes overt time (in multiple ways, i.e., different types of *Concept Drifts*), allowing for a better understanding of the variations that occur in the process and how, in practice, it is truly being performed over time.

Our work focuses on the identification of models associated with each version of the process. The technique we propose is a tool that helps business managers to make decisions. For example, it can help determine if the changes produced in the implementation of the process are really those that are expected, and based on this, take the proper action if they discover abnormal behaviors. Also, by understanding and comparing the different versions of a process, good and bad practices can be identified, which are highly useful when the time comes to improve or standardize the process.

In this document we present a set of metrics to measure the performance of our approach, i.e., whether the approach is able to cluster data in the same way the data was created. Thus, these metrics require *a priori* process information, which is not feasible for real cases.

Each metric measures a different aspect, which when used together, allow for a multi-

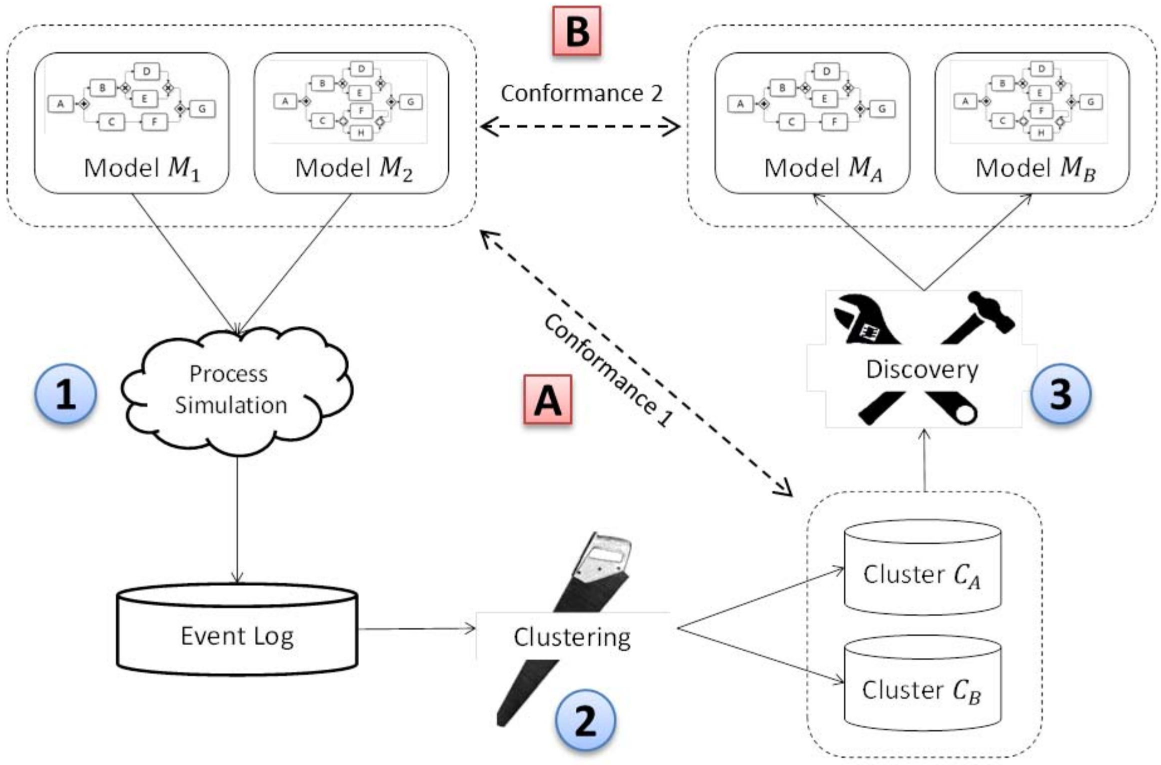
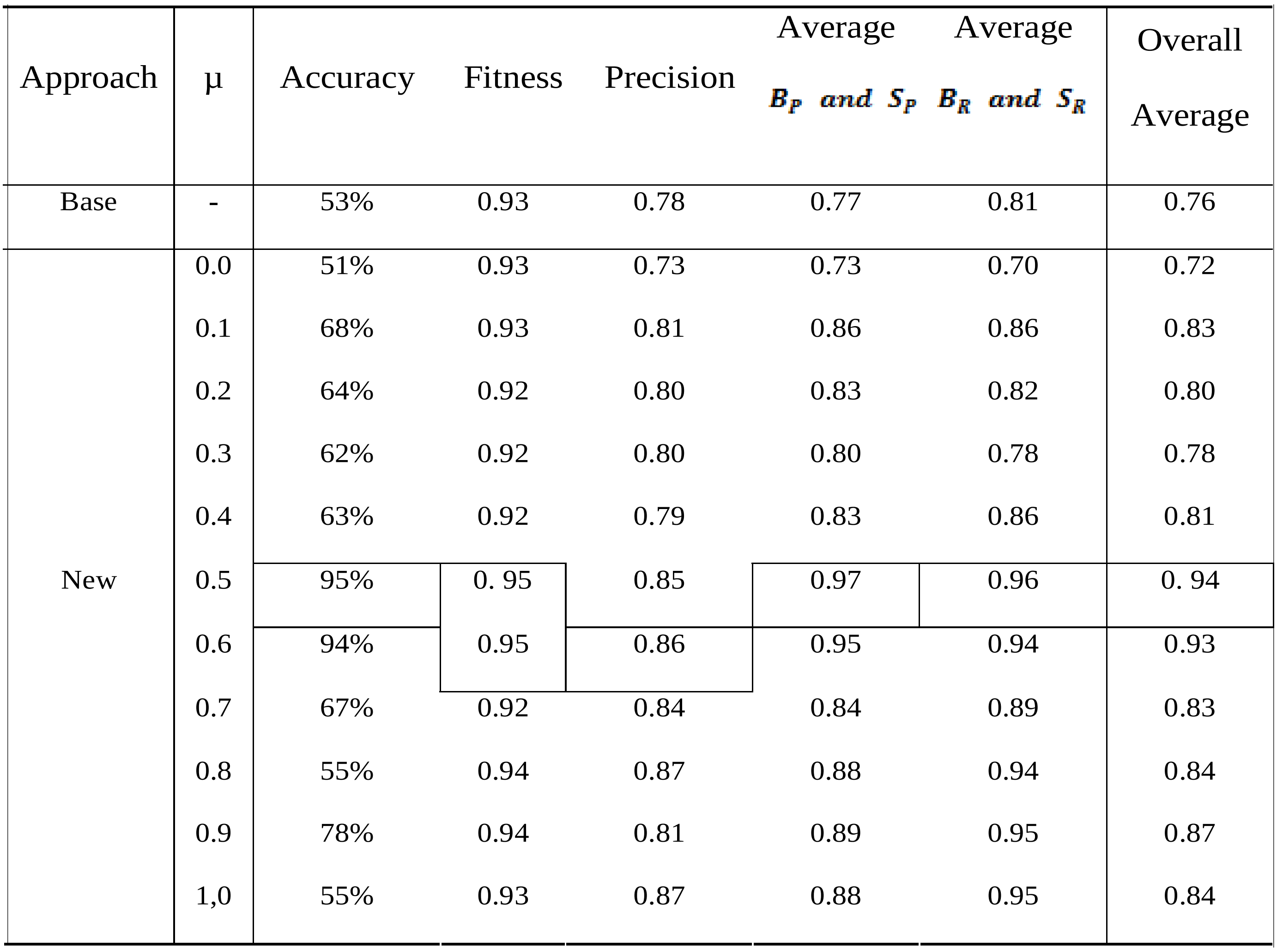


Figure 3. Steps for Performing the Experimental Tests.

Approach	µ	Log (a)	Log (b)	Log (c)	Log (d)	Log (e)	Log (f)
Base	-	57%	38%	55%	86%	99%	53%
New	0.0	59%	52%	49%	82%	59%	51%
	0.1	65%	52%	49%	82%	59%	68%
	0.2	87%	52%	63%	100%	59%	64%
	0.3	87%	52%	63%	100%	59%	62%
	0.4	95%	52%	63%	100%	59%	63%
	0.5	100%	52%	63%	100%	59%	95%
	0.6	100%	52%	73%	100%	100%	94%
	0.7	96%	89%	73%	100%	100%	67%
	0.8	78%	88%	73%	74%	84%	55%
	0.9	79%	77%	77%	68%	77%	78%
	1.0	81%	78%	68%	73%	72%	55%

Table 4. *Accuracy* Metric Calculated for the Six Synthetic Test *Logs*.



Approach	µ	Accuracy	Fitness	Precision	Average B_P *and* S_P	Average B_R *and* S_R	Overall Average
Base	-	53%	0.93	0.78	0.77	0.81	0.76
New	0.0	51%	0.93	0.73	0.73	0.70	0.72
	0.1	68%	0.93	0.81	0.86	0.86	0.83
	0.2	64%	0.92	0.80	0.83	0.82	0.80
	0.3	62%	0.92	0.80	0.80	0.78	0.78
	0.4	63%	0.92	0.79	0.83	0.86	0.81
	0.5	95%	0. 95	0.85	0.97	0.96	0. 94
	0.6	94%	0.95	0.86	0.95	0.94	0.93
	0.7	67%	0.92	0.84	0.84	0.89	0.83
	0.8	55%	0.94	0.87	0.88	0.94	0.84
	0.9	78%	0.94	0.81	0.89	0.95	0.87
	1,0	55%	0.93	0.87	0.88	0.95	0.84

Table 5. Different Metrics for Analyzing Log (f)

faceted vision that makes the analysis more complete.

One key aspect of our clustering approach is the value it is given to the weight of the temporal dimension, parameter μ, which is closely related to the nature of the process. High μ values give a greater importance to time when carrying out the clustering, whereas low μ values give more importance to the structural features of the process.

The experiments results show that the approach proposed in this document has a better performance and that exists at least a value for the parameter μ that gives better results in comparison to only using the structural clustering technique (*Trace clustering* based on patterns). This is achieved because our approach is capable of grouping the *log* traces in such a way so as to identify structural similarity and temporal proximity at the same time.

One of the metrics used is *accuracy*. In some experiments, this metric reached 100%, meaning all traces were classified correctly. When a 100% *accuracy* was not reached, it was because there are processes whose different versions are similar amongst themselves, and therefore there are traces that can correspond to more than one version of the process, which makes the classification not exactly the same as what is expected.

Our future work in this line of investigation is to test the new approach with real processes. We also want to work on developing the existing algorithms so that they are capable of automatically determining the optimal number of *clusters*. In order to do so, it will be necessary to define new metrics that will allow us to calculate the optimal number of clusters without *a priori* information of the process versions.

References

[1] W. van der Aalst. *Process Mining, Discovery, Conformance and Enhancement of Business Processes.* Springer, 2011. ISBN 978-3-642-19345-3.

[2] R.J. Bose, W. van der Aalst, I. Zliobaite, M. Pechenizkiy. Handling Concept Drift in Process Mining. *23rd International Conference on Advanced Information Systems Engineering.* London, 2011.

[3] R. Bose, W. van der Aalst. Trace Clustering Based on Conserved Patterns : Towards Achieving Better Process Models. *Business Process Management Workshops*, pp. 170-181, 2010. Berlin: Springer Heidelberg.

[4] W. van der Aalst, B. van Dongen, J. Herbst, L. Maruster, G. Schimm, A. Weijters. *Data & Knowledge Engineering*, pp. 237-267, 2003.

[5] M. Song, C. Günther, W. van der Aalst. Trace Clustering in Process Mining. *4th Workshop on Business Process Intelligence (BPI 08)*, pp. 109-120. Milano, 2009.

[6] R. Bose, W. van der Aalst. Context Aware Trace Clustering: Towards Improving Process Mining Results. *SIAM*, pp. 401-412, 2009.

[7] W. van der Aalst, A. Adriansyah, A.K. Alves de Medeiros, F. Arcieri, T. Baier, T. Blickle et al. *Process Mining Manifesto*, 2011.

[8] T. Stocker. Time-based Trace Clustering for Evolution-aware Security Audits. *Proceedings of the BPM Workshop on Workflow Security Audit and Certification*, pp. 471-476. Clermont-Ferrand, 2011.

[9] D. Luengo, M. Sepúlveda. Applying Clustering in Process Mining to find different versions of a business process that changes over time. *Lecture Notes in Business Information Processing*, pp. 153-158, 2011.

[10] A.V. Ratzer, L. Wells, H.M. Lassen, M. Laursen, J. Frank, M.S. Stissing et al. CPN Tools for editing, simulating, and analysing coloured Petri nets. *Proceedings of the 24th international conference on Applications and theory of Petri nets* pp. 450-462. Eindhoven: Springer-Verlag, 2003.

[11] A.K. Alves De Medeiros, C. Günther. Process Mining: Using CPN Tools to Create Test Logs for Mining Algorithms. *Proceedings of the Sixth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, pp. 177–190, 2005.

[12] A. Rozinat, W. van der Aalst. Conformance testing: Measuring the fit and appropriateness of event logs and process models. *Business Process Management Workshops*, pp. 163-176, 2006.

[13] J. Muñoz-Gama, J. Carmona. A fresh look at precision in process conformance. *Proceeding BPM'10, Proceedings of the 8th international conference on Business process management*, pp. 211-226, 2010.

[14] A. Rozinat, A.K. Alves De Medeiros, C. Günther, A. Weijters, W. van der Aalst. Towards an Evaluation Framework for Process Mining Algorithms. *Genetics,* 2007.

[15] J. Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, pp. 236-244, 1963.