

# Process Mining of Enterprise Applications based on OutSystems Agile Platform

Alexandre L. Gouveia

IST – Instituto Superior Técnico  
Avenida Prof. Dr. Cavaco Silva 2744-016 Porto Salvo, Portugal  
E-mail: alexandre.gouveia@ist.utl.pt

**Abstract.** Process Mining allows the analysis of business processes based on event logs. When a certain application is running, data referring to its events are recorded to these event logs. Using the details of the recorded events, such as their identification, their status, the date and time they were executed and the connection they have to other events, the business processes embedded in this application can be discovered. By an analysis of a considerable amount of events through process mining algorithms, it is possible to reveal behavioral patterns as well as the process model that drives the application execution. Through the study and comparison of these patterns and models with the ones that modeled the application, an overall improvement to the application may be achieved through successive refinements. The purpose of this paper is to illustrate how process mining can be applied in a specific application development platform, namely the Agile Platform of OutSystems.

**Keywords:** Process Mining, OutSystems Platform, Agile Development, Event Log, Process Models

## 1 Introduction

The term of Process Mining was introduced by [6] for the method of extracting the description of a structured process from a set of events, where each event refers to a task (i.e., a real execution). In their work they referred to process discovery concepts [12] and the idea of applying this method on the context of workflow management [8].

This paper presents a study on how to retrieve usage information from applications developed in the OutSystems Agile Platform [23]. More specifically, the approach taken was based on several Process Mining concepts and techniques [29,1,3,6,5] in order to analyse sets of events that were triggered by users interacting with those applications and, ultimately, to allow improvements of the business models that define them. These sets of events are often stored together within specific logs. New and powerful process mining techniques are being incorporated by software vendors in their products, but few of the more advanced process mining techniques have been tested on real-life processes [4]. Therefore it is of interest to verify if the platform records information about those processes.

One of the great challenges of Process Mining lies in accessing event logs recorded by any enterprise application in order to retrieve and store the relevant information. The

challenge has been acknowledged and partially met by a large framework supporting process mining research<sup>1</sup> – the ProM Framework [13,2].

In this context, the framework was used to analyse the event logs produced by the OutSystems Platform with the following goals:

- Study the possible methods to extract run-time data from applications developed in the OutSystems Platform.
- Develop preprocessing techniques to achieve a standard log format from the extracted data, with all the required fields.
- Identify the process mining techniques relevant for analyzing the extracted logs.
- Implement visualization capabilities for the extracted models.
- Compare the models developed in the OutSystems Platform with the extracted models and related information from event logs analysis.

A custom solution was developed as well, named as LogRetriever, to perform similar kinds of analysis and more, such as performance analysis as well. Altogether, these goals aim to support the development cycle of the applications developed in the OutSystems Platform. By applying process mining techniques it is possible to study the run-time behavior of these applications in new ways which, in turn, will foster the continuous development of these applications.

## 2 OutSystems Platform

OutSystems is a registered trademark and company distributing a platform based on agile development methodologies<sup>2</sup>. With the recent technology breakthroughs on information systems, new tools have been developed to offer methods of agile development [27,7]. These allow a great flexibility when developing applications, since they can be easily altered when necessary.

The Agile Platform is currently used by software teams to rapidly develop and manage flexible web applications. On the 5.0 version of the platform [24], the possibility of business processes development using agile methodologies was fully integrated. The platform is currently used for business applications and web sites by companies across industries including Telecom, Banking and Insurance, Pharma, Utilities, Consumer Goods and Government.

The platform comprises the main components briefly described below (overview in figure 1).

The **Integration Studio** is a desktop environment where developers create custom components to integrate external applications and databases. Its wizards help in the integration of databases, library APIs and third party applications such as SAP.

The **Service Studio** is a desktop environment targeted at business-minded developers to rapidly assemble and change web business applications and business processes using visual models. The web application is completely defined within this studio, where Web 2.0 User Interfaces, Business Logic, Databases, Integration Components,

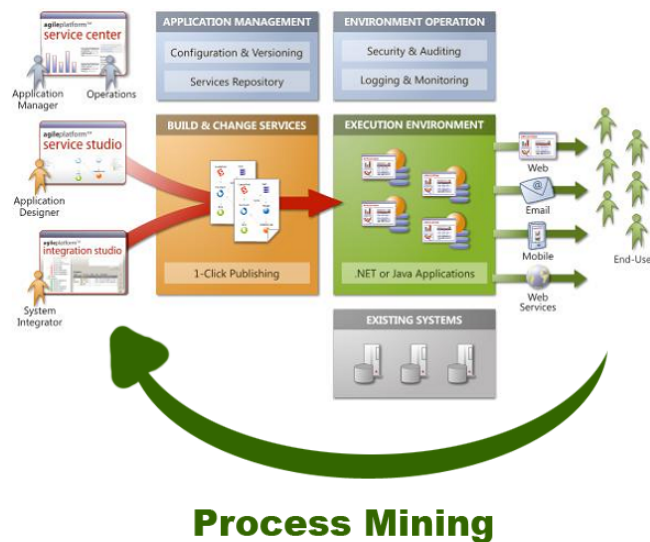
---

<sup>1</sup> More information at [www.processmining.org](http://www.processmining.org)

<sup>2</sup> Agile Platform at [www.outsystems.com/agile-platform](http://www.outsystems.com/agile-platform)

Web Services, Security Rules, Scheduling activities and Business Processes can be modeled. Deployment and publishment is made by 1-Click Publishing, an automated mechanism within the studio, and a drag-and-drop programming style is adopted to allow the creation of applications without hand-written code.

The **Service Center** is a console for the operational management of the platform. It allows monitoring and auditing of the running applications to detect and isolate performance and quality issues, and to manage highly scalable, 24x7 application server farms. Version control and configuration management is available for all web applications, services, integration adapters and other application resources including processes.



**Fig. 1.** Process Mining supports the development of OutSystems Platform's applications[23]

### 3 Process Mining

The aim of Process Mining is to extract knowledge from logs where information systems record events from business contexts of the enterprise and, by comparing the process models that specify those same systems with models that derived from that knowledge, arrive to conclusions of compliance in actions or behaviors in those contexts.

An event log typically contains information about events referring to an activity and a case. The case being handled (also named as process instance) can be a customer order, a job application, an insurance claim, a building permit, etc. The activity, named task, operation, action, or work item is some operation on the case with a timestamp associated, indicating the time of occurrence. Moreover, when people are involved on

these operations, event logs will characteristically contain information of the performer, executing or initiating the event.

Throughout this discovery, management questions can be answered, such as:

- Who in the organization is responsible from certain events occurrence?
- How much time was spent between specific tasks in the process?
- How many people are typically involved in a case? Do they require assistance for completing the case?
- Do the given cases comply with the existing process model?

The process mining techniques described in the following subsections are supported by the ProM Framework [13,2] and were chosen due to their characteristics and relation to the context of event logs of OutSystems' applications. Allowing the interaction between a great number of plug-ins, which are nothing less than implementations of algorithms useful in process mining area, the framework requires a standard structure for log files to use those same plug-ins. An event log under the ProM Framework has an XML structure using a specific schema known as WorkflowLog. To distinguish it from an ordinary XML file, the MXML file format was adopted, having a specific XML schema to be followed.

The relevant event logs provided by the OutSystems Platform are the ones related to the web pages that the users visit when they interact with the Platform's applications. These are screen logs, due to the fact that each web page is represented by a screen transition, which in its turn, is associated to one or several actions. To provide a better understanding on how these logs relate to the ProM Framework and the MXML schema elements, Table 1 illustrates the format of the screen logs.

ProM Framework Fields	Screen Log Fields
Process	Espace_Id (Application's Identifier)
Process Instance	Session_Id (Application's session)
Workflow Model Element	Screen (action referring to a web page)
Timestamp	Instant (time and date of the visit)
Originator	User_Id (user's identifier)

**Table 1.** Fields comparison between OutSystems applications' logs and ProM Framework

### 3.1 Control-Flow Mining Techniques

The goal of workflow mining is to discover the process and collect data at run-time to support workflow design and analysis[3]. Instead of starting with a workflow design, the objective is to start gathering information about the workflow processes as they take place, considering that events are totally ordered by timestamp and each one refers to a task (i.e., a well-defined step in the workflow) and to a case (i.e., a workflow instance).

When a user of an application visits several webpages from it, a workflow is possible to establish by connecting these visits as if they were events themselves. This is

allowed to due to the similarity between pages and tasks. Each page refers to a screen action which is a well-defined step in the workflow (as a task), a session which is a workflow instance (as a case) and a timestamp which allows to order the entire sequence of visits to the pages. Therefore, due to these characteristics, the Alpha ( $\alpha$ ) algorithm [6,30,17,21], the Heuristics Miner [28] and the Genetic algorithms [19,18,20] are examples of control-flow-based algorithms that can be applied to the event logs.

### 3.2 Sequence Clustering

In practical applications with large event logs, applying the above techniques to large event logs may generate very complex models. In order to understand these models it is necessary to apply preprocessing techniques, such as sequence clustering, to study different behaviors separately. With the sequence clustering algorithm, sequences are partitioned into a number of clusters or groups of similar sequences, each with an associated Markov chain [26]. Each chain has a set of states with the respective transition probabilities between those states, where each state has a unique dependence with the previous state (first-order Markov Chain [11]), or has multiple dependencies to a set of previous states (higher-order Markov Chains [16]).

The goal of sequence clustering is to extract the sequences of an event log (relating to specific cases) into groups where each group can be analyzed separately, thus generating simpler process models (Ferreira [15]). This is particularly useful in the event logs of the OutSystems Platform in order to detect usage patterns within an application, or even within several similar applications that share the same webpages. Therefore within this context, this technique allows to mine human activities, as presented in the case study of Ferreira [14]. By recording all the activities taking place in a daily work of a team (e.g., interacting with an OutSystems' application) it is possible to sort out different user behaviors, thus providing insight into the underlying structure of those behaviors.

### 3.3 Social network miner

Event logs' tasks are commonly associated with their performers, i.e. the people or entities initiating or completing a given task. When this information is combined with the concepts from workflow management, it is possible to discover and analyse social networks [5]. If the events are ordered in time the log allows the inference of causal relations between activities and the corresponding social interaction. The hand-over of work from one performer to the next one can be discovered by using these inferences. The strength of their relation can be measured by the amount of times that work was transferred between them.

Social Network Analysis (SNA) refers to the collection of methods, techniques and tools in sociometry aiming and visualization at the analysis of social networks [9]. There are several metrics presented in Aalst, Reijers and Song [5] in this context based on causality (performers handling cases to one another), joint cases (frequency of two individuals are performing activities for the same case), joint activities (frequency of users performing specific activities), special events, reassignment, scheduling, etc.

These metrics allow to derive sociograms which in turn enable the application of SNA techniques, such as betweenness, closeness, power, etc. Ethical and legal issues play a significant role in the practical application of process mining in general and SNA analysis in particular, since the behavior and privacy of the performers is exposed.

### 3.4 Performance

There is a constant pressure to improve the performance and efficiency of business processes. Fine-grained monitoring facilities are required in order to achieve this, such as Business Activity Monitoring (BAM), Business Operations Management (BOM), and Business Process Intelligence (BPI). However, the functionality offered by tools such as Cognos<sup>3</sup> and Business Objects<sup>4</sup> (Power and Kaparthy [25] discuss more vendors of web-based Decision Support Systems) is limited to simple performance indicators such as flow time and utilization, but most of these systems do not focus on causal and dynamic dependencies in processes and organizations [4].

While organizational monitoring measures the organizational efficiency (e.g. idle times, workload analysis etc.), technical monitoring is used for performance measurement (e.g. system response time, system load etc.). Performance metrics can be extracted from workflow logs by tools such as the PISA tool [22]. If the workflow log contains a timestamp for each event (timed logs), this information can be used to extract information about the performance of the process, such as, bottlenecks in the process, flow times (mean, min and max), probabilities of occurrence of specific paths and, consequently, the performance of employees associated to the process [1].

## 4 Approach

Although based on the ProM Framework, the approach presented in this section is the result of an independent development for achieving additional knowledge through usage of specific techniques to event logs related to the OutSystems' applications, comprising phases for *data source identification* (where the data is stored when a user interacts with the running application), *preprocessing* (to allow correct placement of data for an event to be considered complete), *application of Process Mining techniques* (using ProM or developed algorithms), *observation and analysis* (for achieving a clearer understanding of the produced models), *model and behaviors comparison* (Service Studio model's may be difficult to understand to a non-developer since different pages of the same application may be separated through various Screen Flows, therefore comparing these flows with the actual flow that one or more users have followed allows to comprehend how these Screen Flows are connected).

### 4.1 Source - LogScreen application

After considering several possible sources of run-time data, the Screen Logs were the event logs that contained the relevant information, as depicted in table 1. An access to

<sup>3</sup> [www.cognos.com](http://www.cognos.com)

<sup>4</sup> [www.businessobjects.com](http://www.businessobjects.com)

the database would require special access to the server, therefore the LogScreen application was developed, a simple OutSystems application with a query similar to the one executed by the monitoring tools of the Service Studio. This application produces an HTML, such as the one in figure 2, containing all the fields from table 1.

Tenant_Id	Instant	Duration	Screen	Session_Id	User_Id	Espace_Id
47	2010-06-19 19:49:39	15	Contact_list	sqeir055f0jsij45csxbvyyg	0	52
47	2010-06-19 19:49:46	0	Contact_Show	sqeir055f0jsij45csxbvyyg	0	52
47	2010-06-19 19:49:47	0	Contact_edit	sqeir055f0jsij45csxbvyyg	0	52
47	2010-06-19 19:49:51	15	Contact_list	sqeir055f0jsij45csxbvyyg	0	52
47	2010-06-19 19:49:59	15	Contact_edit	sqeir055f0jsij45csxbvyyg	0	52
47	2010-06-19 19:50:04	0	Contact_list	sqeir055f0jsij45csxbvyyg	0	52

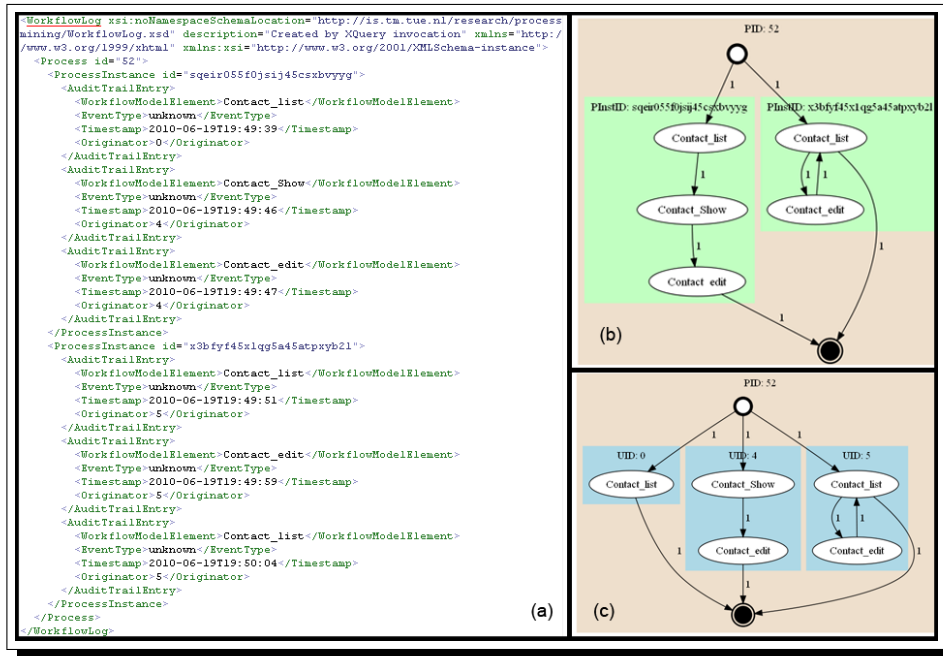
**Fig. 2.** LogScreen Application web page

## 4.2 Preprocessing OutSystems Logs

The main goal is to obtain an MXML document through the HTML data that the LogScreen provides. This document is achieved by applying an XQuery (an XML Query Language [10]) to the HTML document. The XQuery accesses directly to each one of the table's rows and builds an AuditTrailEntry XML element, assigning the page's name, the timestamp and the user's identification to the respective field of the ProM Framework (according to Table 1 on page 4). After acquiring a list of these entries, a new loop is made to verify and place each one of them on the block they belong, thus building the Process Instance and Process XML elements. These blocks when placed inside the WorkflowLog XML element allow the creation of a well-formed MXML document containing all the information of the original event log, as figure 3 depicts (left side).

## 4.3 Process Mining Techniques, Analysis and models comparison

Once the MXML document is acquired, the event logs from the OutSystems' applications can be easily used in ProM. In the context of this work, a tool was developed to fully integrate the preprocessing of Screen Log data and to directly retrieve them from their source of origin (whether in the local disk or in an OutSystems server). The tool produces control-flow graphs, such as the ones presented in figure 3, where flow for different cases (sessions) or originators (users) can be distinguished. Performance analysis were made to these flows to discover the durations of visits to each page, although special considerations needed to be taken into account - transitions between events in the log may have different data that need to be matched in order to conclude that a certain user has left the page. A user may have not left the page if a loop appears in the log,



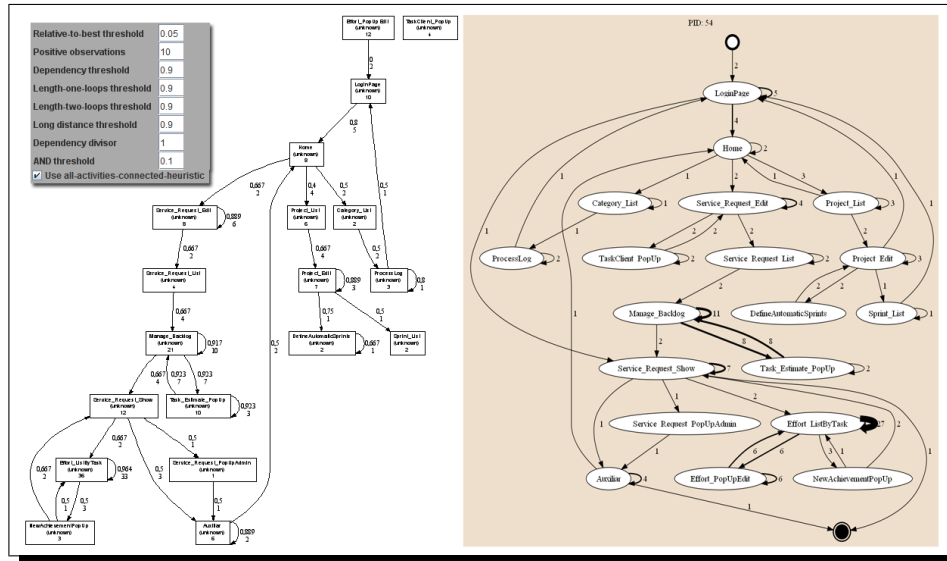
**Fig. 3.** Multiple users (c) and sessions (b) can be distinguished by analysing the same MXML (a)

nor if the end of process or instance has been reached (different interpretations of these facts can lead to different conclusions on user's behaviors).

The ProM plug-ins associated to the process mining techniques described in section 3 were used in the event logs and also, based on these techniques, the tool provides algorithms for another perspective and analysis over the context of the problem. Since each event represents a webpage, loops on a webpage may be associated to a page's refresh or a field's update (e.g. HTML form with a submit button), therefore those loops may be of interest and should not be discarded (figure 4 shows a comparison between the Heuristics Miner's and the tool's analysis). Also, events close to each other in the same case triggered by different users cannot be immediately related to a handover of work in the context of social networks (the cases refer to web-sessions). Particularities such as these ones reveal that both analysis, from ProM or from the developed tool, are of interest to study.

## 5 Conclusions

Applications developed with the OutSystems Platform produce relevant run-time data logs that can be analyzed in order to obtain approximate process models to the ones that were used to develop those same applications. With this work those models were achieved providing additional knowledge to the development process. It is possible to



**Fig. 4.** ProM Heuristics Miner algorithm output (left) and the general screen flow of developed tool (right)

retrieve relevant information of users behaviors and workflows, how they interact with applications developed in the OutSystems Platform and how their behaviors are a basis for improving the original model that defined those same applications.

Further analysis and investigation may lead to different approaches on how to achieve improvements on these applications. For example, different visualization options may be created in order to depict the minimum or maximum flow of pages of one session or of the set of sessions presented in the event log, or to determine which user in the social network takes more time to reach certain pages, or even to find the probability of a user to complete a certain sequence of pages.

## References

1. W. M. P. van der Aalst and B. F. van Dongen. Discovering workflow performance models from timed logs. In Yanbo Han, Stefan Tai, and Dietmar Wikarski, editors, *Engineering and Deployment of Cooperative Information Systems*, volume 2480 of *Lecture Notes in Computer Science*, pages 107–110. Springer Berlin / Heidelberg, 2002.
2. W. M. P. van der Aalst, B. F. van Dongen, Christian W. Günther, R. S. Mans, A. K. A. de Medeiros, A. Rozinat, Vladimir Rubin, Minseok Song, H. M. W. (Eric) Verbeek, and A. J. M. M. Weijters. Prom 4.0: Comprehensive support for real process analysis. In Jetty Kleijn and Alex Yakovlev, editors, *ICATPN*, volume 4546 of *Lecture Notes in Computer Science*, pages 484–494. Springer, 2007.
3. W. M. P. van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. Weijters. Workflow mining: A survey of issues and approaches. *Data & Knowledge Engineering*, 47(2):237–267, November 2003.

4. W. M. P. van der Aalst, H. Reijers, A. Weijters, B. F. van Dongen, A. K. A. de Medeiros, M. Song, and H. Verbeek. Business process mining: An industrial application. *Information Systems*, 32(5):713–732, July 2007.
5. W. M. P. van der Aalst, Hajo A. Reijers, and Minseok Song. Discovering social networks from event logs. *Comput. Supported Coop. Work*, 14(6):549–593, 2005.
6. W. M. P. van der Aalst, A.J.M.M. Weijter, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16:2004, 2003.
7. P. Abrahamsson, J. Warsta, M. T. Siponen, and J. Ronkainen. New directions on agile methods: a comparative analysis. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 244–254, May 2003.
8. Rakesh Agrawal, Dimitrios Gunopulos, and Frank Leymann. Mining process models from workflow logs. In Hans-Jörg Schek, Gustavo Alonso, Felix Saltor, and Isidro Ramos, editors, *Advances in Database Technology - EDBT'98*, volume 1377 of *Lecture Notes in Computer Science*, pages 467–483. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0101003.
9. H. Russell Bernard, Eugene C. Johnsen, Peter D. Killworth, Christopher McCarty, Gene A. Shelley, and Scott Robinson. Comparing four different methods for measuring personal social networks. *Social Networks*, 12(3):179 – 215, 1990.
10. Scott Boag, Donald D. Chamberlin, Mary F. Fernández, Daniela Florescu, Jonathan Robie, and Jérôme Siméon. XQuery 1.0: An XML Query Language. World Wide Web Consortium, Recommendation REC-xquery-20070123, January 2007.
11. Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Model-based clustering and visualization of navigation patterns on a web site. *Data Min. Knowl. Discov.*, 7(4):399–424, 2003.
12. Jonathan E. Cook and Alexander L. Wolf. Discovering models of software processes from event-based data. *ACM Transactions on Software Engineering and Methodology*, 7:215–249, 1998.
13. B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst. The ProM Framework: A New Era in Process Mining Tool Support. In *Lecture Notes in Computer Science: Applications and Theory of Petri Nets 2005: 26th International Conference, ICATPN 2005, Miami, USA, June 20-25, 2005. / Gianfranco Ciardo, Philippe Darondeau (Eds.)*, volume 3536, pages 444–454. Springer Verlag, June 2005.
14. Diogo R. Ferreira. Applied sequence clustering techniques for process mining. In Jorge Cardoso and Wil van der Aalst, editors, *Handbook of Research on Business Process Modeling*, Information Science Reference, pages 492–513. IGI Global, 2009.
15. Diogo R. Ferreira, Marielba Zacarias, Miguel Malheiros, and Pedro Ferreira. Approaching process mining with sequence clustering: Experiments and findings. In Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors, *Business Process Management, 5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007, Proceedings*, volume 4714 of *Lecture Notes in Computer Science*, pages 360–374. Springer, 2007.
16. Mark Girolami and Ata Kabán. Sequential activity profiling: Latent dirichlet allocation of markov chains. *Data Min. Knowl. Discov.*, 10(3):175–196, 2005.
17. A. K. A. de Medeiros, B. F. van Dongen, W. M. P. van der Aalst, and A. J. M. M. Weijters. Process mining: Extending the alpha-algorithm to mine short loops. In *Eindhoven University of Technology, Eindhoven*, 2004.
18. A. K. A. de Medeiros, A. J. Weijters, and W. M. P. Aalst. Genetic process mining: an experimental evaluation. *Data Min. Knowl. Discov.*, 14(2):245–304, 2007.
19. A. K. A. de Medeiros and A. J. M. M. Weijters. Genetic process mining. In *Applications and Theory of Petri Nets 2005, volume 3536 of Lecture Notes in Computer Science*, pages 48–69. Springer-Verlag, 2005.

20. A. K. A. de Medeiros, A. J. M. M. Weijters, and W. M. P. van der Aalst. Genetic process mining: A basic approach and its challenges. In *Business Process Management Workshops*, pages 203–215, 2005.
21. A. K. A. de de Medeiros, B. F. van Dongen, W. M. P. van der Aalst, and A. J. M. M. Weijters. Process mining for ubiquitous mobile systems: An overview and a concrete algorithm. In *UMICS*, pages 151–165, 2004.
22. Michael zur Muehlen and Michael Rosemann. Workflow-based process monitoring and controlling - technical and organizational issues. *Hawaii International Conference on System Sciences*, 6:6032, 2000.
23. OutSystems. Agile Platform, Technical Overview Datasheet. Published in the OutSystems' website, 2008.
24. OutSystems. Agile Platform, What's New in Version 5.0. Published in the OutSystems' website by Jaime Vasconcelos, February 18, 2009.
25. D. J. Power and S. Kaparathi. Building web-based decision support systems. *Studies in Informatics and Control*, 11:291–302, 2002.
26. Gil Aires da Silva and Diogo R. Ferreira. Applying hidden markov models to process mining. In A. Rocha, F. Restivo, L. P. Reis, and S. Torrao, editors, *Sistemas e Tecnologias de Informação: Actas da 4a. Conferência Ibérica de Sistemas e Tecnologias de Informação*, pages 207–210. AISTI/FEUP/UPF, 2009.
27. W. H. Morkel Theunissen, Derrick G. Kourie, and Bruce W. Watson. Standards and agile software development. In *SAICSIT '03: Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, pages 178–188, , Republic of South Africa, 2003. South African Institute for Computer Scientists and Information Technologists.
28. A. J. M. M. Weijters and A. K. A. de Medeiros. Process mining with the HeuristicsMiner algorithm. BETA Working Paper Series, WP 166, Eindhoven University of Technology, Eindhoven, 2006.
29. T. Weijters and W. M. P. van der Aalst. Process mining: Discovering workflow models from event-based data. In Kröse, B., Rijke, M., Schreiber, G., and Someren, M., editors, *Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001)*, pages 283–290, 2001.
30. Lijie Wen, Jianmin Wang, and Jianguang Sun. Detecting implicit dependencies between tasks from event logs. pages 591–603. 2006.