# Mining Software Development Process Variations

Renata M. S. Santos
COPPE, Federal University of Rio de Janeiro (UFRJ) / Federal Fluminense Institute (IFFluminense)
Cidade Universitária, Centro de Tecnologia, Rio de Janeiro - Brazil
+55 21 3938-8672
renatames@cos.ufrj.br

Toacy C. Oliveira
COPPE, Federal University of Rio de Janeiro (UFRJ)
Cidade Universitária, Centro de Tecnologia, Rio de Janeiro – Brazil
+55 21 3938-8672
toacy@cos.ufrj.br

Fernando Brito e Abreu
ISTAR / ISCTE-IUL
Av. das Forças Armadas
1649-026 Lisboa, Portugal
+351 21 790 3295
fba@iscte-iul.pt

## ABSTRACT

Process tailoring aims to customize a software process to better suit the specific needs of an organization when executing a software project or due to a social context in which the process is inserted. Tailoring happens, in general, through variations in the process elements, such as activities, artifacts, and control flows. This paper aims to introduce a technique that uses process mining to uncover elements from the software process that are candidates for tailoring. The proposed approach analyzes the execution logs from several process instances that share a common standard process. As a result, execution traces that differ from the standard process flow are identified and assessed to uncover their variable elements. The proposed technique was evaluated with data extracted from a real software development scenario when a large system was under development for a set of Brazilian Federal Institutes of Education, Science and Technology.

## Categories and Subject Descriptors

• Software and its engineering

## General Terms

Algorithms, Management, Documentation.

## Keywords

Variation; Software Process; Process Mining; Process Tailoring.

## 1. INTRODUCTION

Software development organizations continually seek to improve their software development and maintenance processes, since the latter are directly related to the quality of the resulting software products [1]. Processes are important because they orchestrate activities, people, and information [2] in achieving a common goal. To promote the adoption of a process in an organization, it must be well documented. According to [3], in the last decade process modeling has become an important mechanism in understanding the dynamic behavior of software development organizations.

Among the existing process modeling languages, we can stand out two that have raised the attention of both practitioners and researchers for modeling software development processes [20] and are supported by several tools [11]: *SPEM (Software Process Engineering Metamodel Specification)* and *BPMN (Business Process Model and Notation)*.

To describe a software development process, we must expose the logical organization of its technical and managerial activities, involving agents, methods, tools, artifacts, and constraints [4]. In general, software development organizations build a standard software development process model. The latter is a process to be used across the organization as-is or refined according to each software project's needs and execution context [5].

Building processes from scratch involves considerable effort. As a result, developers often tailor existing off-the-shelf or standard processes [6]. Process tailoring is considered such an important mechanism that quality models such as ISO/IEC 15504, ISO/IEC 12007, and CMMI identify this tailoring activity as a requirement for any software development organization [7]. However, defining a standard process model that takes into account the characteristics of the organization and a future tailoring activity for a specific project are not trivial activities. Tailoring a software development process includes deleting, updating, or adding new elements and/or relationships in a software process [7]. Typically, tailoring is executed based on the tacit knowledge project managers and process engineers have about the organizational forces and context, but it does not consider the explicit changes that have occurred during past process executions. Such changes comprise unforeseen or neglected situations typical to software development projects due to several aspects such as the volatility in software requirements or team rotation.

Software processes are heavily based on creative activities [8], whereas changes can be added to process execution to accommodate unforeseen situations such as time or budget constraints. Then, process execution may deviate from the initially defined process model, therefore mischaracterizing it. In this context, the use of process management environments that allow capturing and storing details about process execution, in combination with process mining techniques that allow analyzing such information (semi-) automatically, would bring a new dimension to the discovery and persistence of such changes in the process.

*SAC'15*, April 13-17, 2015, Salamanca, Spain.
Copyright 2015 ACM 978-1-4503-3196-8/15/04…$15.00.
http://dx.doi.org/10.1145/2695664.269604

The goal of process mining is to use event data to extract process-related information, e.g., to automatically discover a process model by observing events recorded by some enterprise system [9]. According to [12] the starting point for process mining is an event log. All process mining techniques assume that it is possible to sequentially record events such that each event refers to an activity and is related to a particular case (i.e., a process instance)[12]. Many process mining algorithms have been implemented in various academic and commercial systems [9]. A notable case is the *ProM Framework*. It is implemented as an extensible open source (Java) workbench[1] with a user-friendly GUI and has a wide variety of plugins that provide many process mining techniques [12]. The input log files for *ProM* are expressed in *XES (eXtensible Event Stream)*, a standard XML-based format for events log, that facilitates the exchange of events logs between tools and application domains [16]. Process mining techniques (algorithms) are able to extract knowledge from event logs [9, 10].

In order to help project managers and process engineers to perform process tailoring, this paper proposes a technique to uncover the elements found in a software process model that are candidate for adaptation. Candidate process elements are discovered by applying process mining techniques to a project repository that contains execution traces for several process instances. Our goal here is to pinpoint such candidate process elements, rather than automatically performing tailoring. We believe that the decision for adaptation should lie on project managers and process engineers, since they must consider other contextual characteristics that are not captured in process repositories.

The proposed technique was validated by using data from a real scenario in which a team of developers was in charge of developing an academic management system for several Brazilian Federal Institutes of Education, Science and Technology. The team worked under a well-defined development process for two years, and all tasks were recorded in a task management system for later inspection and administrative purposes.

This paper is organized as follows. Section II describes the process variations identification technique. Section III includes the validation of that technique. Section IV describes related work, and, finally, in Section V, some conclusions and future work are outlined.

# 2. VARIATIONS IDENTIFICATION TECHNIQUE

This work proposes a technique to uncover process elements that are candidates to tailoring. The technique is based on the combination of process execution information and mining. Figure 1 shows a representation in BPMN for the process of the variations identification technique, called *VarIdentify*. The process is composed by four activities and their related artifacts: (i) *Get Standard Process Model*; (ii) *Get Event Log*; (iii) *Apply Process Mining Techniques* and; (iv) *Perform Analysis of Conformance*.

The activities *Get Standard Process Model* and *Get Event Log* start the process and occur in parallel. The activity *Get Standard Process Model* gets the standard organizational process model. In general, this process model is provided Project managers and process engineers. This process model should be represented in BPMN due to the aforementioned rationale. In parallel the event logs should be obtained by the activity *Get Event Log*. In order to better control their projects, software development organizations usually automate their development processes through a monitoring system. This system is capable of registering valuable information about process execution, such as timestamps, task identification, stakeholders involved in the execution of an activity, and so on, in the form of execution logs, that is, event logs[22].

To apply the process mining techniques, the event logs must contain at least the following information: identification of process execution (process instance), task title, and start and end timestamps. Before executing the activity *Apply Process Mining Techniques*, the event logs should be preprocessed, i.e., cleaned, integrated, and transformed to the XES format[2]. After preparing the data, process mining techniques are then applied for extracting the executed process model. The executed process model that is obtained is then composed of activities and control flows that depict the process based on real data.

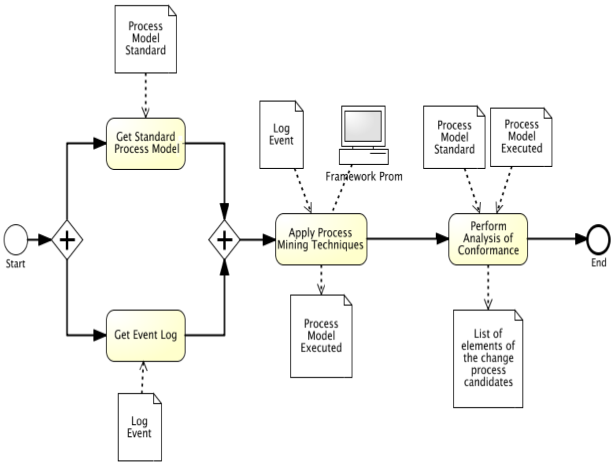

**Figure 1: Variations Identification (*VarIdentify*) Overview**

The activity *Perform Conformance Analysis* ends the process. This activity manually performs the conformance analysis between the standard process model and the executed process model, in order to identify the variable process elements, that is, the Process Variation Points. The standard process model obtained from activity *Get Standard Process Model* and the executed process model obtained from activity *Apply Process Mining Techniques* are analyzed to identify the variations. These variations are activities added and removed from the standard process model. Added activities comprise unforeseen executed activities discovered from executed process model. Removed activities are planned activities that were not executed.

To support conformance analysis we use *BPMNt*, a conservative extension to BPMN for tailoring representation. This extension uses the same terminology found in SPEM: *extension,*

[1] Available at http://www.promtools.org/

[2] For simplicity sake, these activities are not shown in Figure 1.

*suppression, localContribution, and localReplacement* [14]. The conformance analysis generates a list of candidate "tailorable" process elements, which may be represented in the standard process model. The list is composed of activities that have been added, that is, those that were not foreseen in the standard process model and that were performed during the execution process, and the activities that have been removed, which were foreseen in the standard process model and were not performed while executing the process. Project managers and process engineers should then decide on the inclusion of the identified variation points in the standard process model.

Summing up, our proposal analyzes the execution logs from several process instances that share a common standard process. As a result, execution traces that differ from the standard process flow are identified and assessed to uncover their variable elements. Section 3 presents a proof of concept that aims to validate the feasibility of the proposed technique.

## 3. VALIDATION

With the objective of evaluating the effectiveness of our technique, we performed a proof of concept based on real data from the development process of an integrated academic management system, called *SIGA-EPCT* [17]. The development of this system began in 2008 and was developed collaboratively by developers and researchers from the Federal Institutes of Education, Science and Technology involved in the project, through research centers geographically distributed throughout Brazil. The project had a defined software process specifically designed to meet its needs, which comprised the following phases: *Planning*, *Requirements*, *Specification and Design*, *Implementation*, *Test*, and *Deployment*. However, in order to validate our work, in this section we will only consider the Specification and Design phase, since it is the core process phase. This phase is composed of the following activities: *Describe Use Case*, *Review Use Case Description*, *Specify Report*, *Design Screen, Elaborate Class Diagram*, *Update General Class Diagram*, *Review Use Case Specification*, *Define Test Cases*, *Elaborate Physical Model*, and *Update Database*.

We recall that our proposal starts with the activity *Get Standard Process Model*, where we aim to obtain the standard process model. In parallel, the execution event logs were obtained by the activity *Get Event Log*. We extracted from the *Redmine* task manager system, 151event logs, corresponding to thirteen process instances of the *Specification and Design* phase. The information selected were use case (*process instance*), task title, start and end dates, and responsible coordination, which were transformed to a suitable format to the ProM framework, the .xes format.

Following up on evaluating our technique, we progressed to the *Apply Process Mining Techniques* activity. In spite of the project having a well-defined process, we realized that the execution logs did not follow it closely. When a process is unstructured, the traditional mining algorithms tend to overgeneralize the generated model, allowing an excessive amount of behaviors [23]. In order to solve this problem, some clustering techniques were applied. Clustering is the method used for partitioning records from a database into clusters (subsets) in order to allow that data from a cluster may use a set of common properties, which distinguish them from data on other clusters [15]. To do so, we used a process mining clustering technique, available in *ProM's DWS* plugin, which splits a log on sets of instances (clusters) that have similar flows, dividing a complex problem into several simpler ones. The *ProM* framework provides a summary after executing the DWS Plugin, which allowed us to each. Each cluster was modeled in BPMN because the next activity, *Perform Conformance Analysis* uses the *BPMNt* extension to present results. The cluster 0 represents the executed process model regarding to 10 process executions.

The conformance generated a list of candidate process elements to be changed, which is composed of the changes occurred along each process execution. To scaffold the presentation of the list of candidate process elements to be changed, we use *the localContribution* and *Supression* mechanisms from the *BPMNt* extension. The list of candidate process elements of the cluster 0, is shown in Table 1. Each candidate process element to be changed is associated with the mechanisms from *BPMNt* performed. We recall that is up to the project manager or the process engineer to decide whether the standard process model will be tailored, or reused with the identified changes.

**Table 1: Results of Conformance Analysis**

| Executed Process Model | Mechanism (BPMNt) | Activity |
|---|---|---|
| *Cluster 0* | *LocalContribution* | Design and Validate User Interfaces |
| | *Suppression* | Specify Report<br>Design Screen |

Some threats to the validity of this feasibility study are: i) the manual generation of the list of candidate elements to be changed, ii) the selection of process instances, since we only analyzed a sample of them, and iii) the manual modeling in BPMN of the standard and executed process model. Since this evaluation is just a proof of concept, we require further validation efforts in the future.

## 4. RELATED WORK

The process tailoring strategy proposed by [6] aims to support project managers to determine how to adapt a software process to face specific project challenges. Another model-driven engineering (MDE) approach uses abstract models that are systematically transformed into more concrete models, and eventually into source code [18]. In this context, [19] proposed the use of MDE to perform tailoring on organizational processes by means of model transformations. The authors [21] define "method engineering" as an engineering discipline to design, construct and adapt methods, techniques and tools for the development of information systems. In doing so, [13] proposes the Situational Method Engineering (SME), which refers to a potential solution for the problem of the selection of the most appropriate development method for a project and proposes the construction of a specific development method or process for each project according to its characteristics, from fragments stored in a repository, called method base.

All the aforementioned strategies to process tailoring do not explicitly take into account the changes that occur during past process executions. Such changes comprise unforeseen or neglected situations that are typical in software development projects, which was the motivation for our research.

# 5. CONCLUSION

This paper presented a technique to uncover the elements from the software process model, which are used to perform tailoring, from the executed process, in order to provide project managers and process engineers with a mechanism to aid in process tailoring. Our technique took into account the changes that occurred during past process executions. Such changes comprise unforeseen or neglected situations that are typical in software development projects due to the volatility in software requirements and rotation in the team itself. Our main goal was to highlight the candidate process elements to be changed, through process mining, since it is the job of project managers and process engineers to produce the final decision on the inclusion of the identified variations.

We evaluated our technique with real data extracted from the development project of an academic management system, which possesses a well-defined development process, where each task was recorded in a web-based task management system. From the logs, after applying a clustering method, it was possible to get the executed process, and through conformance analysis between the standard process model and executed process models, it was possible to identify the variations of the process, which may be represented in the standard process model and serve as a future basis for process tailoring.

As a future work, we intend to automate the *Perform Analysis of Conformance* step of our approach.

# 6. ACKNOWLEDGMENTS

This work was partially supported by the Federal Fluminense Institute of Education, Science and Technology (IFFluminense) and Brazilian agencies CAPES and CNPq.

# 7. REFERENCES

[1] Fuggetta, A. Software process: a roadmap. In Proceedings of the Conference on the Future of Software Engineering. ACM, 2000, 25-34.

[2] Pfleeger, S.L. & JM Atlee. Software Engineering: Theory and Practice 4th Ed. Prentice Hall, 2010.

[3] Campos, A. L., & de Oliveira, T. C. Modeling Work Processes and Software Development-Notation and Tool. In ICEIS, pp. 337-343, 2011.

[4] Pressman, R.S. & Maxim, B.R. Software Engineering, A Practitioner's Approach, 8th Ed. MCGrawHill, 2014.

[5] Rozinat, A., & Van Der Aalst, W. M. Conformance checking of processes based on monitoring real behavior. Information Systems, 33(1), 2008, 64-95.

[6] Xu, P., & Ramesh, B. Using process tailoring to manage software development challenges. IT Professional, 10(4), 2008, 39-45.

[7] Pereira, E. B., & Bastos, R. M. Uma proposta para adaptação de processos de desenvolvimento de software baseados no Rational Unified Process (Doctoral dissertation). Pontifícia Universidade Católica Do Rio Grande Do Sul, PUCRS, 2005.

[8] Bendraou, R., Sadovykh, A., Gervais, M. P., & Blanc, X. Software Process Modeling and Execution: The UML4SPM to WS-BPEL Approach. In Software Engineering and Advanced Applications, 33rd EUROMICRO Conference IEEE.2007, 314-321.

[9] IEEE Task Force on Process Mining. Process Mining Manifesto. In BPM Workshops, volume 99 of Lecture Notes in Business Information Processing. Springer-Verlag, Berlin, 2011, 169-194.

[10] Yue, D., Wu, X., Wang, H., & Bai, J. A review of process mining algorithms. In Business Management and Electronic Information (BMEI), 2011 International Conference on, 2011. (Vol. 5, pp. 181-185). IEEE.

[11] Gartner Group. Magic Quadrant for Business Process Analysis Tools. [S.l.]. (ID: G00161090), 2008.

[12] Van Der Aalst, W. M. P. Process mining: making knowledge discovery process centric. ACM SIGKDD Explorations Newsletter, 13(2), 2012, 45-49.

[13] Brasil, M. A., Pereira, G. V., & Fontoura, L. M. Software process tailoring using Situational Method Engineering based on criteria of quality improvement. In Informatica (CLEI), XXXVIII Conferencia Latinoamericana. IEEE, 2012, 1-8.

[14] Pillat, R.M., Oliveira, T.C. Alencar, P.S., & Cowan, D.D. BPMNt: A BPMN Extension for specifying software process tailoring, Information and Software Technology, 2015, 57, 95-115

[15] Goldschmidt, R., & Passos, E. Data Mining: Um Guia Prático-Conceitos, Técnicas, Ferramentas, Orientações e Aplicações. Editora Campus, Rio de Janeiro, 1, 2005

[16] Verbeek, H. M. W., Buijs, J. C., Van Dongen, B. F., & Van der Aalst, W. M. XES, xESame, and proM 6. In Information Systems Evolution. Springer Berlin Heidelberg, 2011, 60-75.

[17] Fabro Neto, A., Turchetti, R. C., Trois, C. & Priensnitz Filho, W. Avaliação Qualitativa e Quantitativa entre Ferramentas para Detecção de Vulnerabilidades em Código Fonte para Aplicações Web. In: 7ª Conferência Ibérica de Sistemas e Tecnologias de Informação, Madrid, 2012.

[18] Schmidt, D. C. Guest editor's introduction: Model-driven engineering. Computer, 39(2), 2006, 0025-31.

[19] Hurtado Alegría, J. A., Bastarrica, M. C., Quispe, A., & Ochoa, S. F. An MDE approach to software process tailoring. In Proceedings of the 2011 International Conference on Software and Systems Process. ACM, 2011, 43-52.

[20] Portela, C., Vasconcelos, A., Silva, A., Sinimbú, A., Silva, E., Ronny, M., ... & Oliveira, S. A Comparative Analysis between BPMN and SPEM Modeling Standards in the Software Processes Context. Journal of Software Engineering and Applications, 5, 2012, 330-339.

[21] Brinkkemper, S. Method engineering: engineering of information systems development methods and tools. Information and software technology, 38(4), 1996, 275-280.

[22] Lemos, A. M., Sabino, C. C., Lima, R. M. F., & Oliveira, C. A.Conformance Checking of Software Development Processes Through Process Mining. In SEKE, 2011, 654-659.

[23] Veiga, G. M., & Ferreira, D. R. Understanding spaghetti models with sequence clustering for ProM. In Business Process Management Workshops. Springer Berlin Heidelberg, 2010, 92-103.