

Doctoral Symposium

PROFES 13

Title

Comprehend and enact the software processes by modeling with BPMN in context of BPM

Your name and contact information

Name: Andre Luis Nogueira Campos

E-mail: andreln campos@gmail.com

E-mail: andrecampos@ufrj.br

My advisor name: Toacy Oliveira

My advisor E-mail: toacy.oliveira@gmail.com

Abstract

Software development organizations need to continuously comprehend and improve their software processes they use. A common manner to foster comprehension is through modeling, a technique that allows the representation of several interrelated concepts. Moreover, models and their associated model instances are subject to automatic reasoning, which can be used to detect inconsistencies. BPM (Business Process Management) is a known discipline in business that uses a common process specification language, BPMN (Business Process and Notation), associated with a set of tools to help process management, execution, analyzing and optimization. As a result using BPMN and BPM in the context of Software Development Process (SDP) can leverage on the BPMN's infrastructure to improve SDP quality and comprehension.

Research area and sub-area of your work

Research area: Software engineering.

Sub-area: Software process.

What is the problem?

Software development organizations need to continuously improve the quality of their products, while facing challenges such as the rapid change in the technologies involved in building and maintaining the software and the increasing complexity of these technologies [1]. To address these challenges organizations are effortlessly looking for ways to improve the software development process and the quality of the software they deploy [2].

A complete comprehension of the practices performed by the software team is necessary for finding process inefficiencies and errors, so it is obvious the importance of software process definition and deployment for any software process initiative. We named "software process initiative" any initiative to describe, deploy, enact, monitor, simulate, analyze, or improve the software process. Actually, studies show that the effort made in implementing process models and standards in the software area, could produce better quality software and productivity increased software development [3], [4] [5] [6] [7] [8].

In fact, several initiatives to organize and better understand software processes were developed. For instance, process models such as the Unified Process, SCRUM and XP, try to specify and organize processes so that developers can use them as guidelines. Many software process initiatives are utilizing SPEM as the modeling language, because Object Management Group (OMG), an organization that gathers industry and the academic community to foster standards such as the UML (Unified Modeling Language), maintain the SPEM 2.0. Despite expectations, SPEM does not provide whole concepts or tools for modeling process behavior precisely. Consequently, SPEM fails to address process execution, optimization and simulation [9], which are important activities used in process analysis and improvement.

Furthermore, the reality in organizations is far from the ideal. In Brazil, for example, a survey was conducted with 223 public organizations. They conducted research and auditing, wanting to identify if organizations had their software development processes defined and executed, among others issues. This survey was conducted in

2007, and only 50% of the respondents had their software processes defined and documented. This survey was repeated in 2010, three years later, with the same public organizations. This time, only 49% of the respondents claimed to have their software processes defined and documented [10].

Another initiative in Brazil, MPS.BR, helps support the improvement of the quality of software. This initiative works with public and private companies across the country. Among the 423 companies audited by the MPS.BR, only 54 reached the level “E” or higher, i.e. have defined and documented its software processes. This represents only 12.59% of the companies audited [11].

It is fundamental to define and document the software process as a prerequisite of any software process initiative. Therefore, for many reasons, it is a difficult task to ensure that software process is well defined and it is running properly. Harder still is to verify that it is running correctly or not, and improve it.

What is the significance of this problem?

According van de Aalst et al [12] [13], the practices shown in Figure 1 are established in Business Process Management area as a lifecycle:

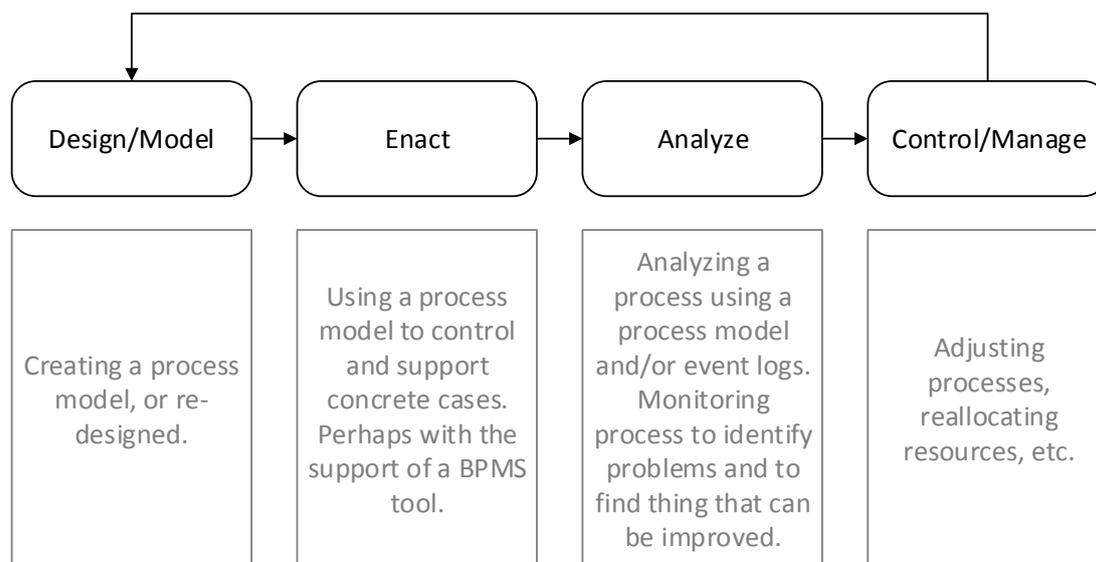


Figure 1. The BPM Lifecycle, adapted from [12] and [13].

As illustrated in Figure 1, this cycle demonstrates a chain of dependencies. There are three steps before achieving process management. The *Design* step mandates a process should be specified, typically using a representation that can be later refined [14] and manipulated during *Enact* and *Analyze*. If a step is not performed, the next step also cannot. Then, if the modeling is not carried out, no other steps also will prevent obtaining the benefits offered by them. As we already mentioned, it is fundamental to define and document the software process as a prerequisite of any software process initiative.

Process improvement initiatives, for example, are related to a software model checking. Regardless of the model used for assessment, the verification is performed by comparing what is practiced with an evaluation criteria. That is why it is so important to document the current practice in a clear and precise manner, because only then it is possible to compare this practice with the evaluation model, such as CMMI and MPS.BR [15]. So, to carry out evaluations of software development practices within an organization, it is crucial that these practices are clearly identified and documented. And more, that these practices are actually performed by the development team.

This means that without clear identification and documentation of software processes of an organization, any initiative to improve the quality of software, and other initiatives, are compromised. In addition, you cannot perform comparative analyses between software development practices of different organizations, thereby preventing the achievement of benchmarks, an important source of information to support the improvement of the quality of the software [16].

Brief literature review, and how the proposed research fits within existing research in the field

Bayona et al [17] claim that organizations are making efforts to implement models and standards for software process improvement, but they still come up against difficulties in their deployment and the processes are not instituted. They did a systematic review of 28 primary studies, to identify factors for process improvement and process deployment initiatives. They identified 16 factors to support the software process initiatives. However, the study does not recommend any practice or framework to solve the problem.

Araujo et al [18] studied the software process definition, or representation, in perspective of the concept, activities and tools suggested by the BPM area. They do that utilizing a case study as a method of research, conducted in an IT department of a Brazilian company. They proposed a framework to elaborate a standard software process. The work was important to building knowledge about the topic, but the conclusions were generic, addressing issues such as the role of the team during the construction of the model, and the breadth of the application of the techniques of BPM for construction of various kinds of processes from standards like CMM and ITIL.

Berrocal et al [19] also applied BPM techniques for software process. This time, they focused on a software that manages tasks of software process. They need to shape the software process in SPEM, translate it to BPMN, and instantiate the process in that software, a BPMS (Business Process Management System). They concluded that it is possible to enact software process with BPM techniques, and that methods and tools facilitating the software project management. Though this study has been very specific, it provided evidence that the techniques of BPM may be useful to define and instantiate software processes.

We, in our turn, believe that software process initiatives can be feasible if supported by clear, understandable and very practical software processes. The software process needs to be usable on a daily basis of organizations, and desirably implemented with the support of tools making it possible to support and monitor their implementation. In addition, it is important to be able to instantiate the process of software in a technological engine that support both the implementation of this process and its monitoring.

We believe the "Software Engineering" field can learn from areas that have achieved effective results in process management. The Business Process Management (BPM) discipline attempts to design, execute, monitor, simulate, analyze and improve processes, and for this objective many tools and techniques were developed throughout the past decades, like patterns interactions identification, enacting processes with BPEL support, processes frameworks building, collaborative processes implementation, quality process improvement, and others [20] [21] [22] [23] [24].

At the core of BPM lays the explicit representation of processes characteristics through the modeling notation, and in BPM world there is a notation in evidence: Business Process Model Notation (BPMN) [25]. Considering that BPMN is a widely used notation and was matured over the years to incorporate several tools and techniques [26], it would be beneficial if the Software Development community could leverage on such maturity to improve quality in software systems. This work presents a qualitative study about the use of the Business Process Modeling and Notation (BPMN) to represent a Software Development Process.

Our proposal is to use the BPMN (Business Process Model and Notation) for modeling of software processes. The goal is to enable the construction of easy-to-understand models and consequently easy to perform by the development team. Moreover, these processes can be run on machines, allowing an objective verification of adherence to the recommended practice.

As a result expected for processes that apply these four aspects illustrated in Figure 1, there is the possibility for increase effectiveness, improve agility and the quality of results [27].

Description of proposed research methodology

Our research is structured in order to evaluate the four main topics mentioned in the BPM Lifecycle [12] [13]. When trying to apply the same cycle within software processes one could argue: 1) How are Software Processes Specified? 2) Is such specification widely used? 3) Can such specification be used as a base to Process Enactment? 4) Can Process Execution create logs that are related to high-level specifications? This research tries to investigate how the adoption of BPM based practices utilizing BPMN could improve software processes management. Our effort is to initially investigate how BPMN can be used to support software process specification and later use BPM tools to foster enactment, as seem in Table 1.

Table 1. Four topics of our research.

Research topic	Research question	How do we intend to find out
Design software process	What is the best notation for modeling software process which can be easy to understand by people and runnable by machine?	<p>St1. Qualitative study about notations, to find out what notation is understandable by people and runnable by machine.</p> <p>St2. Qualitative study about notation selected in St1 to verify if selected notation is suitable for modeling software processes.</p> <p>St3. Qualitative/Quantitative study about impact from BPMN in software process representation, seeking to answer the following questions: will the BPMN notation elements improve the representation of software process? If Yes, is it possible to measure this improvement?</p>
Enact software process	Could the selected notation be appropriate for enact software process?	<p>St4. Qualitative study conducted in simulated environment (<i>in vitro</i>) to see if it is feasible to enact software process.</p> <p>St5. Case study conducted in real environment of industry to check if is possible to enact software process in real life.</p>

Results achieved so far (if any)

We are working on the literature review and constructing various models of the software process. Table 1 shows the main results.

Table 2. Results achieved.

Work done	Status	Result
Positioning of the research from a wide literature review and evaluation of related work.	Completed	-
St1. Qualitative study about notations, to find out what notation is understandable by people and tunable by machine.	Completed	Article published: Modeling Work Processes and Software Development - Notation and Tool. ICEIS (3) 2011 : 337-343.
St2. Qualitative study about notation selected in St1 to verify if selected notation is suitable for modeling software processes.	Completed	Article submitted to PROFES 13.

Future agenda

We still have much work to do in our research. Table 3 presents the main results we want to achieve in 2013 and 2014.

Table 3. Future agenda.

Work to be done	Kind of Work	Status	Estimated time
St3. Qualitative/Quantitative study about impact from BPMN in software process representation, seeking to answer the following questions: will the BPMN notation elements improve the representation of software process? If Yes, is it possible to measure this improvement?	Study	Running	June 2013.
St4. Qualitative study conducted in simulated environment (<i>in vitro</i>) to see if it is feasible to enact software process.	Study	Waiting	November 2013.

St5. Case study conducted in real environment of industry to check if is possible to enact software process in real life.	Study	Waiting	April 2014.
---	-------	---------	-------------

References

- [1] A. M. Magdaleno, C. M. L. Werner and R. M. Araujo, "Reconciling software development models: a quasi-systematic review," *The Journal of Systems and Software*, 2011.
- [2] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw-Hill, 2001.
- [3] K. Buttler, *The economics benefits of software process improvement*, Cross Talk, 1995.
- [4] G. Yamamura, "Software process satisfied employees," *IEEE Software*, pp. 83-85, 09 1999.
- [5] B. Pitterman, "Telcordia technologies: the journey to high maturity," *IEEE Software*, pp. 89-96, 04 2000.
- [6] L. D. Gibson, D. R. Goldenson and K. Kost, "Performance results of CMMI - Based process improvement," CMU/SEI-2006-TR-004, Pittsburgh, PA, 2006.
- [7] N. Mohd, R. Ahmad and N. Hassan, "Resistance factors in the implementation of software process improvement project," *Journal of Computer Science*, pp. 211-219, 2008.
- [8] S. U. Hani, "Impact of process improvement on software development predictions, for measuring software development project's performance benefits," *ACM - Proceedings of the 7th International Conference on Frontiers of Information Technology*, pp. 54:1 - 54:5, 2009.
- [9] R. Bendraou, B. Combemale, X. Crégut and M. P. Gervais, "Definition of an eXecutable SPEM 2.0," in *IEEE Computer Society*, Nagoya, Japan, 2007.
- [10] Tribunal de Contas da União - Governo Federal do Brasil, "Levantamento de governança de TI 2010," TCU, Brasília, 2010.
- [11] MPS, "Softex - Excellence in Software," 11 03 2013. [Online]. Available: http://www.softex.br/mpsbr/_avaliacoes/avaliacoes_mpsbr_total.pdf.
- [12] W. M. P. van der Aalst, A. H. M. ter Hofstede and M. Weske, "Business process management: a survey," *Springer-Verlag Berlin Heidelberg*, pp. p. 1-12, 2003.
- [13] W. M. P. van der Aalst, "Business process management: A comprehensive survey," *ISRN Software Engineering*, p. (in printing), 2013.
- [14] M. C. Branco, J. Troya, K. Czarnecki, J. M. Küster and H. Völzer, "Matching Business Process Workflows across Abstraction Levels," *MoDELS*, pp. 626-641, 2012.
- [15] ISO, "NBR ISO 19.011:2012," ISO, 2002.

- [16] K. E. Emam and A. Birk, "Validating the ISO/IEC 15504 measure of software requirements analysis process capability," *IEEE Transactions on Software Engineering*, pp. v. 25, n. 6, p. 541-567, 2000.
- [17] S. J. Bayona, J. A. Calvo-Manzano and T. S. Feliu, "Critical success factors in software process improvement: a systematic review," *Springer-Verlag Berlin Heidelberg*, pp. 1-12, 2012.
- [18] R. Araujo, C. Cappelli, A. Gomes, M. Pereira, H. S. Iendrike, D. Ielpo and J. A. Tovar, "A definição de processos de software sob o ponto de vista da gestão de processos de negócio," in *VI Simpósio Internacional de Melhoria de Processos de Software*, São Paulo, Brasil, 2004.
- [19] J. Berrocal, J. García-Alonso and J. M. Murillo, "Lean management of software processes and factories using business process modeling techniques," *Springer-Verlag Berlin Heidelberg*, pp. 321-335, 2010.
- [20] K. Schulz and M. Oklowska, "Facilitating cross-organisational workflows with a workflow view approach," *Data and Knowledge Engineering*, vol. 51, no. 1, pp. 109-147, 2004.
- [21] A. Barros, M. Dumas and A. Ter Hofstede, "Service interaction patterns," *Lecture Notes in Computer Science*, vol. 3649, pp. 302-3018, 2005.
- [22] J. Jung, I. Choi and M. Song, "An integration architecture for knowledge management systems and business process management systems," *Computers in Industry*, vol. 58, no. 1, pp. 21-34, 2007.
- [23] P. Trkman, "The critical success factors of business process management," *International Journal of Information Management*, vol. 30, no. 2, pp. 125-134, 2010.
- [24] M. Chen, D. Zhang and L. Zhou, "Empowering collaborative commerce with Web services enabled business process management systems," *Decision Support Systems*, vol. 43, no. 2, pp. 530-546, 2007.
- [25] Object Management Group, "Business Process Model and Notation (BPMN)," 2010.
- [26] M. Chinosi and A. Trombetta, "BPMN: An introduction to the standard," *Computer Standards & Interfaces*, pp. 124-134, 2011.
- [27] T. Panagacos, *The ultimate guide to business process management*, CreateSpace Independent Publishing Platform, 2012.