Spider-PE: A Set of Support Tools to Software Process Enactment

SPEM Process Enactment in the CMMI-DEV and MR-MPS-SW Context

Carlos Portela, Alexandre Vasconcelos
Informatics Center (CIn)
Federal University of Pernambuco (UFPE)
Recife, Brazil
{csp3, amlv}@cin.ufpe.br

Sandro Oliveira, André Silva, Elder Silva
Postgraduate Program in Computer Science (PPGCC)
Federal University of Pará (UFPA)
Belém, Brazil
srbo@ufpa.br, {aandrecunhas, elderferreirass}@gmail.com

***Abstract*—In order to have a competitive software industry, it is essential to adopt standards and reference models of software processes quality. However, despite the growing adoption of standards and models, the number of organizations that adopt these is a small portion of the total population of software organizations. This paper presents a set of support tools to enactment of modeled processes in the Software Process Engineering Metamodel (SPEM). This set of support tools, called Spider-PE (Process Enactment), aims to assist software organizations in the implementation of the Capability Maturity Model Integration for Development (CMMI-DEV) and Reference Model of Software Process Improvement to Software (MR-MPS-SW) models. We expect Spider-PE to be more easily adopted by software organizations because it is based on models and standards largely accepted. Furthermore, this set of support tools adopts free technologies (non-proprietary) in order to reduce costs.**

Keywords-Software Process Enactment; Quality Models; SPEM; CMMI-DEV; MPS.BR.

I. INTRODUCTION

Achieving competitive advantage, to software companies, involves not only product's quality improvement and parallel services, but also the process production and software distribution [1]. In order to, nationally or internationally, have a competitive software department, it is essential the adoption of patterns, standards and reference models when it comes to processes. In this context, the Capability Maturity Model Integration for Development (CMMI-DEV) [2] stands out internationally speaking, while the Reference Model of SPI to Software (MR-MPS-SW) stands out in the Brazilian scope [1].

Although the adoption of norms and reference models for Software Process Improvement (SPI) is increasing, the amount of organizations that adopt these models is a small portion out of the total population of software organizations [3]. Different studies were conducted in order to understand why organizations do not adopt the standards and models for process improvement, and they indicate to questions regarding high costs, lack of support tools and bureaucracy related to the big amounts of resources demanded by the process execution [3][4]. Other studies were conducted in order to identify the critical success factors in software process improvement initiatives [5][6]. These studies take into account the use of support tools as critical success factor in a software process improvement program.

In this context, the purpose of this paper is to introduce the set of support tools Spider-PE (Process Enactment), which gives support to the flexible and semi-automated process execution, and that is adherent to the quality models CMMI-DEV and MR-MPS-SW. This set of tools is based upon free standards and technologies and is the outcome of the SPIDER Project (Software Process Improvement: DEvelopment and Research).

Besides this introductory part, Section II discusses the software process definition, execution and improvement steps, and it also talks about related work. In Section III, a set of tools Spider-PE and its components are presented. Section IV discusses the analysis of the SPEM 2.0 models' execution and the adherence to the CMMI-DEV and MR-MPS-SW models. Section V presents the results obtained in this research in both academy and industry. Finally, Section VI presents a conclusion.

II. BACKGROUND AND RELATED WORK

In this section, we present the main concepts and definitions of this research; then, we briefly describe the related work.

A. Concepts and Definitions

To represent the elements that integrate the process, or in other words, to build the software process models, it is necessary a language to model them [8]. One of the purposes of this type of representation is to facilitate the software process continuous improvement because it enables the understanding of the process in a visual and representative manner among the elements that compose the process [9].

In this domain, two approaches that have a large acceptation in the software industry regarding the modeling area were identified: Software Process Engineering Metamodel (SPEM) [10] and Business Process Modeling Notation (BPMN) [11]. The comparison between these two

 ISBN: 978-1-61208-367-4

standards and the justifications for the adoption of SPEM in this research are presented by Portela et al. [12].

After modeled, the process enters in the stage of execution, where it will be executed, controlled, validated and improved in short incremental and iterative cycles [13]. According to Reis [14], this execution phase depends on an automated mechanism that comprehends the modeled process and guides the developers during their work, as well as executes automatically some tasks.

In accordance with Oliveira et al. [15], one of the most important evolutions in the field of software quality is in the finding that the quality of the software product process is as important as the quality of the final product. From this affirmation, important software quality evaluation and certification mechanisms emerged based on the maturity and capability of the development organization in the conduction of their processes. Therefore, two improvement models are considered in this paper: the CMMI-DEV [2] and the MR-MPS-SW [1], which has been largely adopted in the Brazilian market.

B. Related Work

The WebAPSEE [16] environment allows the software process management. Based on free software, this environment uses its own visual language to model, the WebAPSEE-PML that is based in the formal specification defined in Reis [14]. França et al. [17] highlights the use of this environment in the adoption of the MPS.BR level G in an organization that develops software. However, there are no explicit evidences that the environment supports some of the results from this level, for example, the vertical traceability.

The Ontology-based software Development Environment (ODE) environment was designed based upon on a specific ontology for Software Quality [18]. The project ODE affirms that ADSs built based on anthologies allows an easier integration with different tools that aids software engineering activities.

The TABA station [19] aims to integrate support tools according to organization specificities, software processes and projects. The tool AvalPro supports the Process and Product Quality Assurance group from an organization. The tool Pilot supports the evaluation of improvement proposes of a process in a systematic, planned and controlled way. Using these tools, the TABA station presents an explicit support to the CMMI levels 2 e 3 e the MR-MPS-SW levels G to C processes areas.

The ImPPros environment [20] supports the implementation of software processes in an organization progressively. This approach takes into account the use of quality models and norms that guide the continuous process improvement and the software process transformation bases on the possible mapping among these models and standards.

Differently from these other environments, the approach presented in this paper introduces a execution formalism based on the standard SPEM 2.0, allowing the process model execution without using intermediary models, as it can be seen in details in Section IV. Furthermore, this approach considers the models CMMI-DEV and MR-MPS-SW through the utilization of good practices related to the institutionalization degree of the process execution in an organization that develops software.

III. SPIDER-PE: SET OF SUPPORTING TOOLS

The set of supporting concept adopted in this paper defines a set of technologies that can be integrated in order to aid in the software process execution. In this context, there are tools, techniques, procedures, processes, roles, methodologies, frameworks, languages, standards, patterns, and so on.

A. xSPIDER_ML: Enactment Language

Although the SPEM [10] is a standard defined by OMG [11], it does not offer native mechanisms to automated software process simulation and execution. Because of such limitation, a language for execution was defined, xSPIDER_ML. The xSPIDER_ML (eXecutable SPIDER_ML) is an extension of the modeling language SPIDER_ML [21], which is defined as a profile of SPEM that aids the process execution flexibly [22].

The xSPIDER_ML's structure was defined based on the structure proposed by xSPEM [13], since the both approaches have as goal turn the SPEM 2.0 into executable language. This structure divided in packages provides ways to define the conceptual structure to organizations, providing the necessary notions to execute their developing processes. Therefore, the xSPIDER_ML structure was divided in five packages: xSPIDERML_Core, ProcessParameters, ProjectVariables, EventTypes and ProcessTrace.

In order to be clear, only a subset of concepts of these packages will be presented, and they were selected in accordance with their relevance to the understanding of the components that composes the xSPIDER_ML. The components are available in the technical specification [23].

The execution operates on the instantiated processes and because of that, the elements of the SPIDER_ML are gathered in the package xSPIDERML_Core. Besides these, the package xSPIDERML_Core reuses the concepts and elements offered by the xSPEM [13] and the SPEM 2.0 [10] in order to provide all the necessary elements to define and organize a software process for later execution. These elements define the basis for all the remaining packages of the xSPIDERML.

In this package, there is the component Activity, a specialization of the WorkBreakdownElement and WorkDefinition, which defines basic work units in a process, as well as in a process per se. The class Process

 ISBN: 978-1-61208-367-4

represents a set of work definitions partially ordered with the intention of achieving the development goals, such as the delivery of a system. These processes are defined as sequences of Phases and Milestones, and they express the life cycle of a product being developed. Also in this package, there is the class TaskUse that represents the instance to a TaskDefinition (a class in the SPEM 2.0 structure and SPIDER_ML). This class must provide information related to the resources that will be involved during the execution of the task that it represents.

After the structure of the xSPIDER_ML's components be defined, it is necessary to define the rules that will be applied in these elements and their relationships. In this context, rules define ante and post conditions in the same way as an inference engine of an expert system [14]. These rules extend the SPIDER_ML's semantic and consequently the SPEM 2.0's semantic in order to represent the dynamic information inherent to the properties defined before. In order to be clear, it is presented in this section an example of these rules. The complete detailing of these basic rules is available in [23].

According to the xSPIDER_ML's structure, it is possible to identify a common aspect to the TaksUse components. One task can have a status notStarted, started, paused, finished. An abstract observation of the operational semantic of processes in execution related to this property can be accomplished. Considering t as a task to be executed and whose initial status is notStarted, the possible transitional relationships for t are presented in Figure 1.

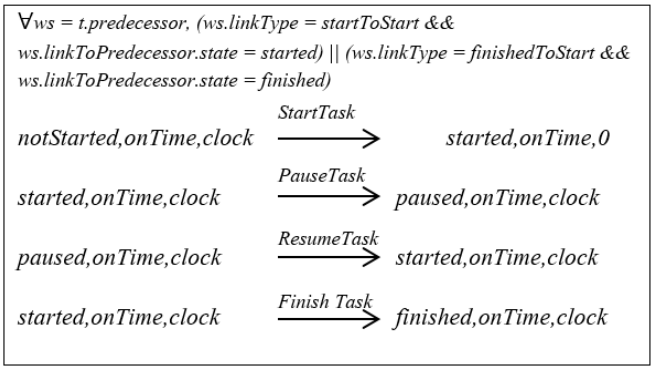

Figure 1. State Transition of a task *t*.

B. SPIDER_PE: Process Enactment Framework

The SPIDER-PE framework was defined in [31], and it aims to support the flexible execution of software processes adherently to the capability levels of the quality models CMMI-DEV and MR-MPS-SW.

It was decided to work with the capability dimension because it relates to the process execution definition directly. Process' capacity is the degree of refinement and institutionalization that the process runs in an organization/organizational unit [1].

The concept of framework adopted in this paper pictures the customization of a process to follow one or more recommendations of the quality models from the perspective of a generic activity flow that are necessary to execute any software process. Figure 2 presents three phases that compose this *framework*. To a complete description of this phases and its components; see [31].

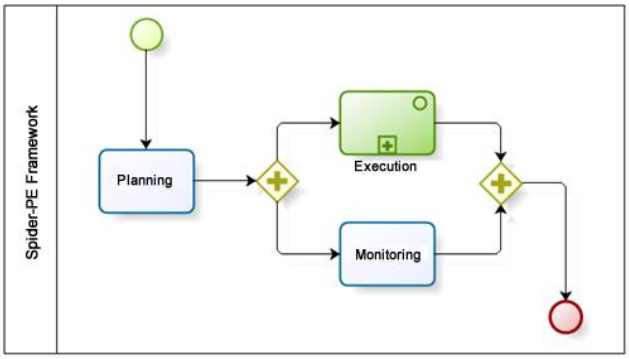

Figure 2. Phases of Enactment Process Framework.

The first phase of the framework is the Process Management Phase that has three steps: Planning, Execution and Monitoring. This phase has as goal the realization of the planning step and the accompaniment of the process execution according with the recommendations of the quality models CMMI-DEV and MR-MPS-SW. In this phase, the Process Policy is defined, the resources and the schedule are estimated, and so forth. Besides that, this phase has a sub phase called Execution of Process Activities, which is defined as the second phase of the framework.

Therefore, the next phase is the Execution of Process Activities, responsible for the actual process execution. In this phase, the team is responsible for creating the work products that are required to conduct the project's activities. To each work product generated, the Configuration Manager must perform the versioning and the control of the access. In the milestones and project's control points, the Quality Assurance team must verify the adherence of the process and work products to the standards and templates.

The third and final phase of the framework consists in the Application of the Execution Formalism, where the tool Spider-PE is responsible to apply the execution formalism defined in the technical specification of the xSPIDER_ML [23]. The purpose of this phase is enable the Framework SPIDER-PE to incorporate the rules and characteristics of the xSPIDER_ML that allows the activities to be instantiated by an execution machine and, consequently, meet the concept of flexibility and semi-automated execution specified in this language. Therefore, this phase must occur in parallel to the other phases of the framework, for this formalism must work on the activities of these phases.

 ISBN: 978-1-61208-367-4

C. Spider-PE: Process Enactment Tool

The Spider-PE is a GPL – *General Public License* [24] tool that brings a solution to the semi-automated software process execution [25]. This tool was created as a desktop environment using Java, and it was based on the use of free technologies [26], such as the IDE Eclipse 3.7, the SGBD MySQL 5.5, the objected-relational mapping *framework* Hibernate 4.0, the library for object serialization XML and vice-versa XStream, the library for creation and manipulation of PDF files iText and the library to draw diagrams based on the graph theory JGraph.

Beyond the free technologies used to develop the tool, other free software systems were used to aid the activities in the Spider-PE. The use of these tools allow the reuse of functionalities that meet the recommendations of the quality models CMMI-DEV and MR-MPS-SW, avoiding the development of specific functionalities to meet certain activities of the Framework SPIDER-PE. Hence, the following tools were integrated to the Spider-PE:

- Subversion (SVN), a system that adopts the Apache License of free software, and it is available in [27], to aid the changing management;
- Redmine, a GPL license tool, available in [28], and that allows the recording, monitoring, and accompaniment of the possible solutions to the different problems that can arise during the project execution;
- Spider-APF, that allows estimates to be made using the Function Point Analysis (FPA); Spider-UCP, that is used to measure the quantity of software from the perspective of the Use Case Points (UCP); and the Spider-CoCoMo, that allows time, effort and team quantity estimates to be using the CoCoMo method – Constructive Cost Model;
- Spider-CL, that aids the process and work product evaluation through objective criteria (described in evaluation checklists);
- Spider-MPlan, which supports the measurement process, which enables the definition, collection, analysis and monitoring measures.

The tools developed by SPIDER Project are under the GPL license and are available in [29].

The Spider-PE tool uses the concept of modules to specify a particular set of features, grouped according to the phases of the framework and the actors responsible for its implementation. Thus, this tool has three modules: Management, Process Management and Process Execution.

The first module to be accessed is the Directors module, which is responsible for setting the tools described in the Subsection 3.3.1. This module is also responsible for converting the XML file that contains the process modeling (built in Spider-PM [30] through notations of the SPIDER_ML language) to the relational database. The Administration module also allows the user to define a Process Manager, who will be responsible for managing, planning and monitoring the process.

Once defined, the Process Manager can access the Process Management module. This module is based on the adherence to the good practices in the quality models CMMI-DEV and MR-MPS-SW; therefore, it is related to the activities of planning and monitoring the process. The Process Management module consists of three phases: Planning, Execution and Monitoring. This module starts from planning the process, where the Process Manager has access to many features.

In the Execution Module Process, the human resources allocated to specific tasks may perform semi-automated and flexible process activities, as can be seen in Figure 3.

Figure 3. Enactment of a process task.

This module consists of the application of the execution formalism xSPIDER_ML. This application occurs in parallel to the steps of Process Execution and Monitoring in the Management module and because of this, the formalism runs concurrently with the activities of this module through the execution engine.

For a full description of the modules and each functionality; see the work of Silva et al. [25][26].

IV. SPIDER-PE EVALUATION

In this section, the set of support tools is evaluated from each of its components.

A. SPEM Models Enactment

The execution language xSPIDER_ML presents itself as a viable proposal for execution of process models defined in the standard SPEM 2.0. First, it is necessary to model the process in the Spider-PM [30] tool. This tool allows the modeling of processes using the notations of SPIDER_ML (profile SPEM 2.0). After the modeling stage, the Spider-PM allows saving an XML with the process modeling. The Administration module of the Spider-PE allows the user to export the information from the modeled process, saved in the XML file, to the relational database. By using the JGraph library, it is possible to apply the rules and formalisms implemented in the xSPIDER_ML under the SPEM notation, as shown in Figure 3.

In this image, the user selected the task Instantiated Task 03 to change its state (Started, Paused, or Finished). By selecting one of these states, the Spider-PE will be responsible for applying an internal mechanism of inference rules, and if there are no restrictions, the new information will be recorded on the database.

B. Adherence to CMMI-DEV and MR-MPS-SW

The tool Spider-PE provides support to the

 ISBN: 978-1-61208-367-4

implementation of Level 2 of CMMI-DEV and Level F of MR-MPS-SW, where the process is considered managed. The choice of these specific levels is due to the fact that they are initial levels and, therefore, tend to be more complex to implement [7].

Thus, for each component of the Spider-PE, it were identified Results of Process Attributes (RPA) of the MR-MPS-SW and Generic Practices (GP) of the CMMI-DEV that are supported by these components. To perform this analysis of adherence, first it was defined component levels of support of the Spider-PE related to the recommendations of these quality models:

- **Total:** the components of the Spider-PE fully support the systematization of the recommendations of a particular set of RPA and GP;
- **Partial:** the components of the Spider-PE partially support the systematization of the recommendations of a particular set of RPA and GP. I.E., these components do not meet all the recommendations of these models;
- **Not support:** the Spider-PE components do not support the systematization of a particular set of RPA and GP recommendations.

Figure 4 shows the relationship between the components of the Spider-PE, the MR-MPS-SW's RPA and CMMI-DEV's GP. This relationship was made from the analysis of the required requirements to meet the recommendations of these models by three experts (officially certified) in the implementation and evaluation of these quality models.

	xSPIDER_ML	Framework SPIDER-PE	Ferramenta Spider-PE
RAP 1 e GP 1.1	!	!	!
RAP 2 e GP 2.1; RAP 3 e GP 2.2; RAP 5 e GP 2.3; RAP 6 e GP 2.4; RAP 7 e GP 2.5; RAP 13 e GP 2.6; RAP 8 e GP 2.7; RAP 4 e GP 2.8; RAP 10 e GP 2.9; RAP 9 e GP 2.10; RAP 13 e GP 2.6	✗	✓	✓
RAP 11, RAP 12 e RAP 14	✗	✓	✓

Label: ✓ Total ! Partial ✗ Not Support

Figura 4. Adherence between the Spider-PE Components to CMMI-DEV and MR-MPS-SW.

To view the details of each of the recommendations of the components listed in the first column of table in Figure 4, it is necessary to consult the official guides of the MR-MPS-SW [1] and CMMI-DEV [2] models. A complete analysis of the adherence of Spider-PE, including the results, is available in [25][31].

V. OBTAINED RESULTS

In this section, the results obtained in this research are presented.

A. In Academy

An initial version of the proposal of this work has been published and presented during the WTDQS - Workshop of Theses and Dissertations in Software Quality [32]. This research is characterized as a subproject of the SPIDER Project, and it was accepted in the 2011/2012 cycle of the PBQP-SW (Brazilian Program of Software Quality and Productivity). In 2012, a comparative study of the patterns of SPEM and BPMN modeling and the proposed implementing xSPIDER_ML language was published in the JSEA - Journal of Software Engineering and Applications [12][22]. The Framework SPIDER-PE was the subject of a dissertation defended at the Informatics Center in the Federal University of Pernambuco (CIn/UFPE) [31]. The research related to the Spider-PE tool was also published in the Free Software Workshop [26] and was ranked among the "Best Papers" in this event, and in the VIII Annual Workshop of the MPS [25].

B. In Industry

The technologies presented in this article are used by the authors in consulting projects related to process improvement. First, the xSPIDER_ML, as well as the results of this phase of the research, were used by companies that are SPIDER project partners, such as the *FabSoft* and the *Pronto Digital*, both located in Belém city. Basically, the language aided on the steps of defining and monitoring the projects. On the other side, the activities of the Framework SPIDER-PE are widely adopted in the implementation of the Level 2 of CMMI-DEV and F of MR-MPS-SW in organizations in which the authors provide consulting, located at *Porto Digital* (Recife city) and *Farol Digital* (João Pessoa city). Finally, it is noteworthy that the last feature of the Spider-PE tool was released in November 2013. Nevertheless, requests for using the tool in consulting processes in a partnership agreement were made by the company *SWQuality* (based in Recife city and subsidiaries Maringá and Belém cities).

VI. CONCLUSION

The purpose of the Spider-PE is to support software development organizations, so they can run their processes flexibly and in a semi-automated way according to the notations of SPEM and the recommendations of the quality models CMMI - DEV and MR- MPS -SW. Therefore, the purpose of the tool is to facilitate the adoption of these standards and quality models for software development.

The set of tools also aims to help the software industry to achieve more satisfactory levels of discipline from the combination of patterns, models, procedures, tools, techniques and methods that help in implementing the process in an automated way that provides information about the progress of the project.

 ISBN: 978-1-61208-367-4

A strong point of this proposal is that the tool is totally free and allows the academic community and/or the industry to contribute to the evolution and improvement of the tool. However, the components of this tool must be customized according to the profile and characteristics of the organization that will use it. Moreover, this tool must be implemented in the organizational department responsible for the software development, requiring, therefore, a strategic, tactical and operational effort of the senior management so it can be deployed in a proper and satisfactory manner.

REFERENCES

[1] Association for Promotion of Brazilian Software Excellence. *SOFTEX: General Guide for Software Process Improvement*. Available from: http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_Geral_Software_2012.pdf 2014.08.02

[2] Software Engineering Institute. *SEI: CMMI for Development*. Available from: http://www.sei.cmu.edu/reports/10tr033.pdf 2014.08.02

[3] M. Staples, et al. "An exploratory study of why organizations do not adopt CMMI", Journal of Systems and Software, vol. 80, pp. 883-895, Jun. 2007, doi:10.1016/j.jss.2006.09.008.

[4] G. Leal, et al. "Empirical study about the evaluation of the implantation of MPS.Br in enterprises of Paraná", Proc. Conferencia Latinoamericana En Informatica (CLEI 12), Oct. 2012, pp. 1-9, doi:10.1109/CLEI.2012.6427201.

[5] C. Almeida, T. Macedo, and A. Albuquerque. "Analysis of the continuity of software processes execution in software organizations assessed in MPS.BR using Grounded Theory", Proc. International Conference on Software Engineering & Knowledge Engineering (SEKE 11), Jul. 2011, pp. 792-797.

[6] M. Montoni and A. R. Rocha, "An Investigation into the Success Critical Factors in Initiatives Software Process Improvement". Proc. Brazilian Symposium on Software Quality (SBQS 11), Jun. 2011, pp. 151-165.

[7] S. Oliveira, et al. "A Proposal for Systemic Solution of a Free Software Tools SUITE to Support to MPS.BR Model Implementation", Magazine of Brazilian Program of Quality and Productivity in Software, 2nd ed., pp. 103-107, 2010.

[8] H. Feiler and S. Humphrey, "Software Process Development and Enactment: Concepts and Definitions", Proc. International Conference on the Software Processes - IEEE Computer Society Press, Feb. 1993, pp. 28-40, doi: 10.1109/SPCON.1993.236824.

[9] M. Kellner and G. Hansen, Software Process Modeling. Pittsburgh, PA: Carnegie Mellon University/Software Engineering Institute, 1988.

[10] Object Management Group. *OMG: Software & Systems Process Engineering Meta-Model Specification*. Available from: http://www.omg.org/spec/SPEM/2.0/PDF 2014.08.02

[11] Object Management Group. *OMG: Business Process Model and Notation (BPMN)*. Available from: http://www.omg.org/spec/BPMN/2.0/PDF 2014.08.02

[12] C. Portela, et al. "A Comparative Analysis between BPMN and SPEM Modeling Standards in the Software Processes Context", Journal of Software Engineering and Applications, vol. 5, pp. 330-339, May 2012, doi:10.4236/jsea.2012.55039.

[13] R. Bendraou, B. Combemale, X. Cregut, and M.-P. Gervais, "Definition of an Executable SPEM 2.0", Proc. Asia-Pacific Software Engineering Conference (APSEC 07), Dec. 2007, pp. 390-397, doi: 10.1109/ASPEC.2007.60.

[14] C. Reis, "A Flexible Approach for Evolutive Software Process Enactment", PhD Thesis in Science Computer. Porto Alegre, RGS: Informatics Institute, Federal University of Rio Grande do Sul, 2003.

[15] S. Oliveira, A. Vasconcelos, and A. C. Rouiller. "A Proposal of Environmental to Software Process Implementation", INFOCOMP - Journal of Computer Science, vol. 4, pp. 71-78, Mar. 2005.

[16] A. Lima, et al. "WebAPSEE: A Free and Flexible Environment for Software Process Management", Proc. Free Software Workshop (WSL 06), Apr. 2006.

[17] B. França, E. Sales, C. Reis, and R. Reis. "Using the WebAPSEE Environment in the MPS.BR Level G implementation in CTIC-UFPA," Proc. Brazilian Symposium on Software Quality (SBQS 09), Jun. 2009, pp. 310-317.

[18] G. Guizzardi, R. Falbo, and R. Guizzardi. "Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The Case of the ODE Software Process Ontology", Proc. Conferencia Iberoamericana de Software Engineering (CIbSE 08), Feb. 2008, pp. 127-140.

[19] G. Travassos, "The Model Integration of TABA Station Tools", PhD Thesis in Science Computer. Rio de Janeiro, RJ: COPPE, Federal University of Rio de Janeiro, 1998.

[20] S. Oliveira, "ProDefiner: A Progressive Approach to the Definition of Software Processes in the Context of a Centered Environment in Process", PhD Thesis in Science Computer. Recife, PE: Informatics Center, Federal University of Pernambuco, 2007.

[21] R. Barros and S. Oliveira, "SPIDER_ML: A Language for Software Process Modeling", Proc. Regional School of Informatics (ERIN 10), Oct. 2010.

[22] C. Portela, et al. "xSPIDER_ML: Proposal of a Software Processes Enactment Language Compliant with SPEM 2.0", Journal of Software Engineering and Applications, vol. 5, pp. 375-384, Jun. 2012, doi:10.4236/jsea.2012.56044.

[23] C. Portela and M. Gomes. *xSPIDER_ML. Technical Specification.* Available from: http://www.spider.ufpa.br/projetos/xspider_ml/xSPIDER_ML.pdf 2014.08.05

[24] GNU Project. *General Public License*. Available from: http://www.gnu.org 2014.08.05

[25] A. Silva, E. Silva, C. Portela, S. Oliveira, and A. Vassconcelos. "Spider-PE: A Support Tool to Implementation of Capacity of the MR-MPS Level F and CMMI-DEV Level 2", Proc. Annual Workshop of the MPS.BR (WAMPS 12), Oct. 2012, pp. 186-194.

[26] A. Silva, E. Silva, C. Portela, S. Oliveira, and A. Vassconcelos. "Spider-PE: A Support Tool to Software Process Enactment adhering to CMMI-DEV and MR-MPS", Proc. Free Software Workshop (WSL 12), Jul. 2012.

[27] Apache Subversion. *SVN Download*. Available from: http://subversion.apache.org/ 2014.08.08

[28] Redmine Project. *Redmine Download*. Available from: http://www.redmine.org/projects/redmine/wiki/Download 2014.08.08

[29] SPIDER Project. *Research Results*. Available from: spider.ufpa.br/index.php?id=resultados 2014.08.08

[30] R. Barros and S. Oliveira, "Spider-PM: A Support Tool for Software Process Modeling", Proc. Annual Meeting of Computing (ENACOMP 10), Oct. 2010.

[31] C. Portela, "Spider-PE: A Support Framework to Flexible Enactment of Software Processes adhering to Quality Models", Master Dissertation in Science Computer. Recife, PE: Informatics Center, Federal University of Pernambuco, 2012.

[32] C. Portela, A. Vasconcelos, and S. Oliveira. "Spider-PE: A Support Set of Tools to Process Enactment adhering to Quality Models", Proc. Thesis and Dissertation Workshop on Software Quality (WTDQS 11), Jun. 2011.

 ISBN: 978-1-61208-367-4