Conceptual Alignment Between SPEM-Based Processes and CMMI

Noureddine Kerzazi
National Higher School for Computer Science and System Analysis
ENSIAS - Rabat, Morocco
Email: {n.kerzazi@um5s.net.ma}

***Abstract*—In the course of process improvement, software organizations attempt to concurrently implement practices that originate from different models, extracting the best from each. Issues of integration and compliance are raised due to missing conceptual view. This paper presents a new method for an explicit alignment between SPEM-based software process modeling and CMMI corpus of best practices. The alignment of process activities with CMMI goes beyond the maturity assessment; it sustains the conceptualization of introducing new software engineering practices at the process-level and open a new topic of combining conceptually parts of different process models. Our method extends the SPEM Meta-model by introducing new interface to CMMI Meta-data. This extension allows visualization of the practices' intentions within the activity breakdown structure. In addition, it offers a conceptual assessment of process maturity level according to CMMI scales. The method has been fully implemented in a process modeling tool and applied in both academic and industrial contexts to justify improvement opportunities through meaningful dashboards.**

***Keywords*—Software Process Modeling, SPEM, SPI, Software Engineering, Meta-model Alignment, Multi-model, CMMI.**

I. INTRODUCTION

Software organizations try to improve their software processes by combining parts of different process models (e.g., CMMI, Six Sigma, Lean), extracting the best from each [23]. However, each model presents its concepts, terminology, and is organized in an appropriate structure to the aims it proposes (e.g., quality, productivity, practices). Finding a suitable strategy for combining different model elements, in the context of multi-model environment, is a challenging task. Indeed, several software companies struggle with the cohabitation of disparate models such as CMMI, Agile models, and quality models. A recent topic discussing the ability to deal effectively with multi-model environment has been raised by the Software Engineering Institute (SEI) [21].

In the context of software process improvement (SPI), there are models that emphasize a view oriented to *"What"* organizations should do (i.e., content-level) and others that emphasize a view oriented to *"How"* organizations achieve goals (i.e., process-level). When organizations have to identify SPI initiatives, they focus on content-level and scant attention to the process-level [4], [3], [24], [6]. This fact leads problems related to the effectiveness of SPI initiatives [10]. For instance, introducing a modern code review practice requires knowing what to review, who should help reviewing and when. Consequently, introducing this practice will affect the entire flow of process activities, roles, and the states cycle of artifacts.

A typical scenario might involve a process engineer in a growing company who has just figured out the relevance of modern code review practice that could increase the software products quality. In contrast to exhaustive compliance pertaining to one process model, adaptability becomes a key element of effective process modeling. For instance, in order to help organizations align and integrate CMMI practices within the existing software activity breakdown structure, it is essential to provide tools for conceptualization, and methods for automatic evaluation. Also, combining different practices coming from different paradigm may raise inconsistencies.

As a fundamental step to achieve a valuable integration of practices coming from different process models, we propose a method for a conceptual integration level, in contrast of a number of works trying to combine/compare different process models at the operational level [17], [18], [16], [15]. This paper presents a strategy to align SPEM-based process activities with corpuses of best engineering practices. Conceptual integration is cost-free and would be useful process analysis. As a proof case of our proposal, we align a SPEM-based process model with CMMI specific practices, a form of integration aiming at: *(i)* operationalizing the CMMI best practices within activity breakdown structure by looking more closely at how CMMI best practices can fit in a structured process model; *(ii)* assessing the capability of process models and providing automatically a dashboard to visualize this integration. The emphasis was not only on assessing the maturity of designed process models, but also on how enhanced practices could be integrated to the existing flow of activities. We implemented the proposal as a new perspective in our tool[1] [12] of software process modeling. We then applied the approach to evaluate an academic case study as well as an adapted version of Scrum [20] used in an industrial context.

The rest of the paper is organized as follows: Section 2 presents theoretical background related to SPEM which is devoted to software process modeling, and the CMMI components. Section 3 discusses related work regarding integration approaches. Section 4 outlines the three pillars steps of our CMMI alignment strategy including the extension of SPEM meta-model, the representation of corpus meta-data and generation of dashboards to visualize indicators of alignment (e.g., continuous process assessment according to CMMI perspective). Section 5 evaluates the viability and effectiveness

[1]DSL4SPM tool for software process modeling can be freely downloaded online at: http://dsl4spm.adilou.com/

of the alignment strategy through two case studies. Section 6 concludes and gives an outlook of future works.

II. Background and Motivations

A. SPEM 2.0 Overview

Unlike ready-to-use process definitions like Scrum, SPEM establishes a common framework for describing software processes. It defines a set of concepts and associated terminology (e.g., Activity, Role, and Work Product). These concepts can be used to build up a customized process structure. Selected sets of these concepts can be used to instantiate a process model for managing and performing the phases of a software system's production.

According to SPEM, a process is a breakdown structure of elements, which embodies an abstract generalization for process elements including roles, activities and artifacts. The latest version of SPEM [22] promotes the reusability of process elements in more than one instance of a process. To do this, SPEM framework has been divided on two main packages: Method Content and Process with Methods. The Method Content package defines the core concepts such as Task, Role and Work Product, while the Process with Methods package allows instantiation of process models as nested activities breakdown structure that embodies essentially the predefined elements in the Method Content. SPEM introduces also the concept of Capability Patterns to enhance reusability by using a subset of activities.

B. CMMI Overview

Capability Maturity Model Integration (CMMI) outlines a corpus of proven practices for development and maintenance activities [22]. It has been broadly used for assessing organizational maturity and process capability [23]. The structure of the CMMI highlights two possible representations: ***Continuous*** and ***Staged***. The continuous representation focuses on capability levels (range from 1 to 5) that are used by organizations which aim to improve their existing processes by emphasizing one area at a time. Each capability level describes generic and specific goals. The implementation of a set of specific practices leads to achieve a specific goal. The staged representation focuses on maturity levels, which range from 2 to 5. The implementation of a set of process areas (PAs) leads to achieve a level of maturity. To facilitate the manipulation of its content, the CMMI model shows another classification according to four categories: *Engineering*, *Support*, *Project Management*, and *Process Management*.

CMMI model provides a guide for choosing process improvement strategies by assessing process capabilities and identifying the critical issues that limit the software quality [7]. The version 1.3 of CMMI for development covers 22 PAs comprising: *i)* a total of 165 specific practices, which are mapped to 48 specific goals and *ii)* a total of 13 generic practices, which are mapped to 3 generic goals.

C. Motivations

Although CMMI maturity indicators could be important for certain software companies that pride themselves to achieve higher maturity levels, orchestration of CMMI practices can be problematic, as stated by Paulk [18]:

> *Standards such as the CMM can help organizations improve their software process, but focusing on achieving a maturity level without addressing the underlying process can cause dysfunctional behavior.*

On the one hand, CMMI provides best practices without guidelines for the structure, and gives indicators on processes capability (focusing on what should be done?). Under this functional point of view, it can be used as a *"Bottom-up"* approach for process tailoring and improvement. On the other hand, SPEM-based software process modeling sustains building activity breakdown's structures (focusing on how?) without providing the best practices. Under this architectural point of view, the framework could be considered as a *"Top-down"* approach for process modeling. Are they simply different viewpoints of the same approach of software process modeling or is there a facet that makes it possible to correlate these different viewpoints? Other relevant issues are raised regarding the compliance across these models such as structure, terminology, level of abstraction, and operational point of view. These discrepancies in concepts' definitions and structures raise a fundamental concern regarding how to integrate CMMI and SPEM-based software process modeling frameworks. Therefore, our contribution is made in this direction by providing a strategy for alignment between SPEM-based process models and CMMI. The conceptualization of the alignment brings responses to the following questions:

- How CMMI and software process models defined using SPEM 2.0 interact with each other?
- How to integrate CMMI content into SPEM-based process model using an articulated and non invasive approach?
- To what extent the SPEM-based process model could be assessed automatically according to CMMI scales?

We begin by answering the first question; the answers for second and third question will follow in section 4.

Table I shows that the SPEM's concepts are sufficient to represent all CMMI's concepts, and accordingly could be used as a starting point for integration. Furthermore, one can see that not all concepts have their equivalent. For example, the concept Role is missed into CMMI. Although CMMI does not specify the nature of the ***Role-Task*** and ***Role-Work*** Product relationships, we use the existing relationships in SPEM namely ***Perform*** and ***Responsibility*** to show how those concepts can be mapped.

III. Related Work

Prior researches have explored strategic alignment in a broad sense [1], [9]. Strategic alignment, also termed *fit*, *bridge*, *linkage* and *integration* [1] relates to techniques of combining different models. For instance, Tallon [25] discusses at process-level the alignment between information technology and business strategy, he points out that the process-level can foster a deeper and more meaningful understanding on how alignment affects organization performance.

Table I: Concepts Mapping.

SPEM concept	CMMI concept
Instance of Kind used to qualify Extensible Elements of SPEM	Process Area
Activity or TaskUse	Specific Practices/Goals
Activity or TaskDefinition	Generic Practices/Goals
Step	Sub Practices/Goals
Work products including all kind of Artifact, Deliverable, and Outcome	Work Products
Role	* Nothing
Guidance	* Nothing

In the following subsections, we discuss the related work organized according to the main approaches of alignment: *Mapping*, *Combining*, and *Formal methods*.

A. Mappings Models

Previous works have proposed methods for mapping different process models with the CMMI. Manzoni and Price [15] describe an assessment of the Rational Unified Process (RUP) based on the CMMI levels 2 and 3 seeking to elaborate proposals for enhancing RUP. They point out missing key practices and propose some activities and artifacts in order to complement RUP (e.g., RUP does not support key process areas (software subcontract management and training). Marcal et al. [16] make the same effort emphasizing agile processes. The authors map CMMI Project Management PA to SCRUM practices and conclude that Scrum does not cover all the specific practices of the Project Management *PA*. Paulk [18] assesses the XP process model and concludes that XP provides a systems perspective on programming, while CMMI provides a systems perspective on organizational process improvement. He claims that organizations should take advantage of the good ideas in both. However, all these attempts of mapping involve the original version of a *ready-to-use* process model, but how about the customized models?

B. Combining models

Siviy and Kirwan [23] discussed a multi-model environment, especially for SPI initiatives, and explore how to capitalize on multiple technologies by extracting the best from each, and manage the complexity and confusion. Motafelija and Stronberg [17] explored the combination of ISO 9001 and CMMI under the topic of process improvement. The authors emphasize the quality of product outcomes as a result of the integration process. Hefner and Sturgeon [8] propose an approach to combine six Sigma and CMMI with the focus to quantify standard deviation between them. Although these studies focus on the utility of such mappings regarding the well known relationship between the development process and the outcome product quality, this kind of mapping has a major drawback that lies in the lack of operational view.

C. Formal methods of models integration

To support SPI efforts in a multi-model environment, the SEI tried to initiate the PrIME project (Process Improvement in Multi-model Environments) [21]. PrIME will attempt to harmonize the process improvement models, and guide the organization in designing strategies on how to implement SPI initiatives. It also proposes combining CMMI to other models

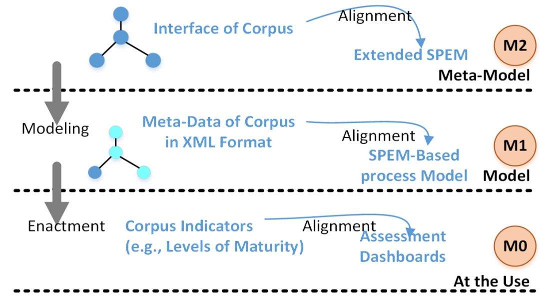


Figure 1: Alignment through three layers

such as Six Sigma, Lean, and agile methods. Furthermore, Malzahn [14] suggests the combination of CMMI to other models across an ontological approach. The central idea consists of capturing the body of knowledge of earch process model into an ontology, and then using a reasoner try to check consistency of the mapping.

Our work differs from previous studies at the abstract level of alignment between SPEM-based process model and corpuses of best practices. For instance, instead of searching the right type of fit between a given process model and CMMI practices, we put emphasis on the conceptual level to extrapolate a SPEM-based process model on a given corpus.

IV. Alignment Overview

In this section, we first present an overview of the alignment approach. we then describe how SPEM metamodel was extended and how CMMI was summarized for the purpose of fluent navigation. Finally, we demonstrate how the alignment works along with an example of a dashboard provided to support the capability measurement.

1) Alignment Overview: In order to satisfy specific needs at different levels, an articulated architecture of alignment is proposed. For simplification purposes, we illustrate the method by aligning CMMI corpus with designed processes models. This choice is motivated by the ability to assess the maturity of our process models, conceptually, according to CMMI levels.

We propose a layered alignment using three layers of process modeling: ***meta-model***, ***model*** and ***at the use***. Each layer corresponds to one level of abstraction emphasizing a particular concern of alignment as illustrated in Figure 1. First, the meta-model layer focuses on extending SPEM to sustain a tight integration with CMMI elements (i.e., concepts mapping). Second, the model layer focuses on the question about how to present the CMMI content in a compact way using a meta-data in order not only to simplify navigation in its content, but also to allow mapping to self-contained units. Third, the *'at the use'* layer focuses on how to present indicators of alignment aiming the assessment of the capacity of process models.

The main characteristic of such an approach of alignment relies on the flexibility, adaptability and openness to integrate other models based on different concepts. These characteristics ensure the continuing evolution over time of both SPEM framework and CMMI model. The major advantage of using a layered alignment resides in the flexibility as well as the facility of adaptation. For example, interoperability in the integration of CMMI content requires the adaptation of independent XML files format and does not affect the meta-model itself. We explain the details for each layer in the next subsections.

A. Extending the SPEM Metamodel

To enable the linkages between each activity (specifically the element task) of process model and CMMI components, SPEM has to be extended. This extension requires not only an extension point, but also a new perspective of modeling. Based on Strategy-oriented perspective of meta-modeling proposed by Rolland [19], we suggest the extension of SPEM to support a new perspective of modeling that complements the Activity-oriented view. According to Rolland, Strategy-oriented view support the investigation of alternative ways of doing a thing and producing a plan for doing it. Figure 2 shows a partial view of our extension of SPEM. In order to support a Strategy-oriented view of process modeling, we enhance the meta-model with two important classes: *Intention* and *Strategy*. these classes are sufficient to represent an intention which link to the goal to achieve, and a strategy class that refers to the way to achieve this goal. By doing so, the class *TaskUse* concretizes an intention, which in turn is linked to the required components of CMMI (i.e., CMMI specific goals) that provide these intentions. The CMMI specific goals, which constitute the prerequisites for CMMI PAs, embody the specific practices required to realize these goals.

Thus, additional semantic information about CMMI concerns has been considered in the design of the proposed SPEM extension to support integration with a corpus of best practices. Once the SPEM has been extended, we then focus on the second layer to represent the CMMI informative components to facilitate the navigation in the CMMI content.

B. Representing Meta-data of CMMI

One of the distinctive features of our method lies in the use of intermediate data (i.e., meta-data) to represent the CMMI model in a compact way. Meta-data refer to structured information that locates, organizes and presents information resources to make it easier to retrieve, use, or manage.

Since these intermediate meta-data are stored in an external XML files, it allows two kinds of adaptation. The first relates to the internationalization, and the second relates to the scope of the content. With the internationalization, it is possible to have multiple sub-files, each of which in a different language (e.g., English, French, etc.). In addition, it is possible to adjust the content of CMMI to represent only the content of a specific level of maturity to fit the needs of organization. For example, a separate XML file of meta-data might be adjusted by selecting a subset of PAs and practices related to the level 2 of maturity, which helps to keep the focus only on goals and specific practices related to this CMMI level.

Figure 3 illustrates the interface of the DSL4SPM tool used to align process activities with CMMI specific practices. A custom form allows linking a given activity to one or more CMMI components that must be embodied by this task. For example, the task named *"MR/PR Analysis"* represents the CMMI intentions REQM1.1, REQM1.2, REQM1.3, and REQM1.4. As it can be seen, the user is supported by a detailed description of each CMMI component.

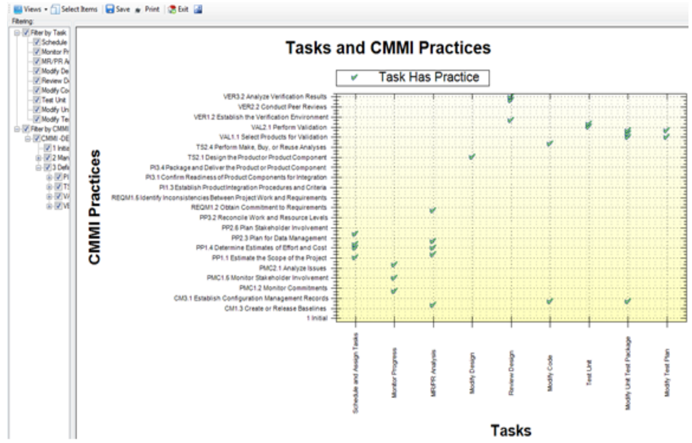


Figure 4: Automatic dashboard as generated by DSL4SPM tool

The CMMI meta-data file is linked to the content file in HTML format, which gives a detailed description of each CMMI component. Moreover, HTML format supports navigation into the content of the CMMI model. Thus, the organization of information in meta-data and content's XML files enables navigation from general to specific.

During early experimentation with four projects, we observed that the participants are able to align the structure of a SPEM-based process model with the CMMI PAs fairly readily. Nevertheless, at a finer-grained level the participants become confused when it comes to align the process activities with the CMMI specific practices. We thus added optional filters (see Figure 3) that limit the space of choice such as choosing a particular level of maturity or the inclusion of generic goals (GG). Furthermore, to allow a good fit to a particular situation, the system provides filters (Figure 3) acting to show only the practices of a given level of maturity or to show/hide the generic practices.

C. Assessment of Process Models Capability

One of the primary goals behind the alignment between process model and CMMI relies on the automatic generation of dashboards to assess process models. The generation of a dashboard consists in gathering CMMI components referenced by each activity and comparing them to the project's needs or to the expectations for a given level of maturity.

Figure 4 shows a prior application of our strategy to evaluate a prescriptive process model namely ISPW-6 [11]. As seen, a high-level overview of the propagation of CMMI practices through the process model tasks. This high-level overview was found to be particularly useful when looking at assessment of process capability. For example, it can be used to identify mismatches in process requirements and expectations or to clarify what practices are implemented in each activity.

Such a dashboard helps to identify the missing practices in the process model that are expected to reach a given level of maturity. Moreover, it is useful for the support of assessment at modeling phase by focusing on a subset of PA to define an improvement path for a particular situation context, which would help emphasis the future improvement effort and resources on promising avenues. The dashboard also provides the two following benefits:

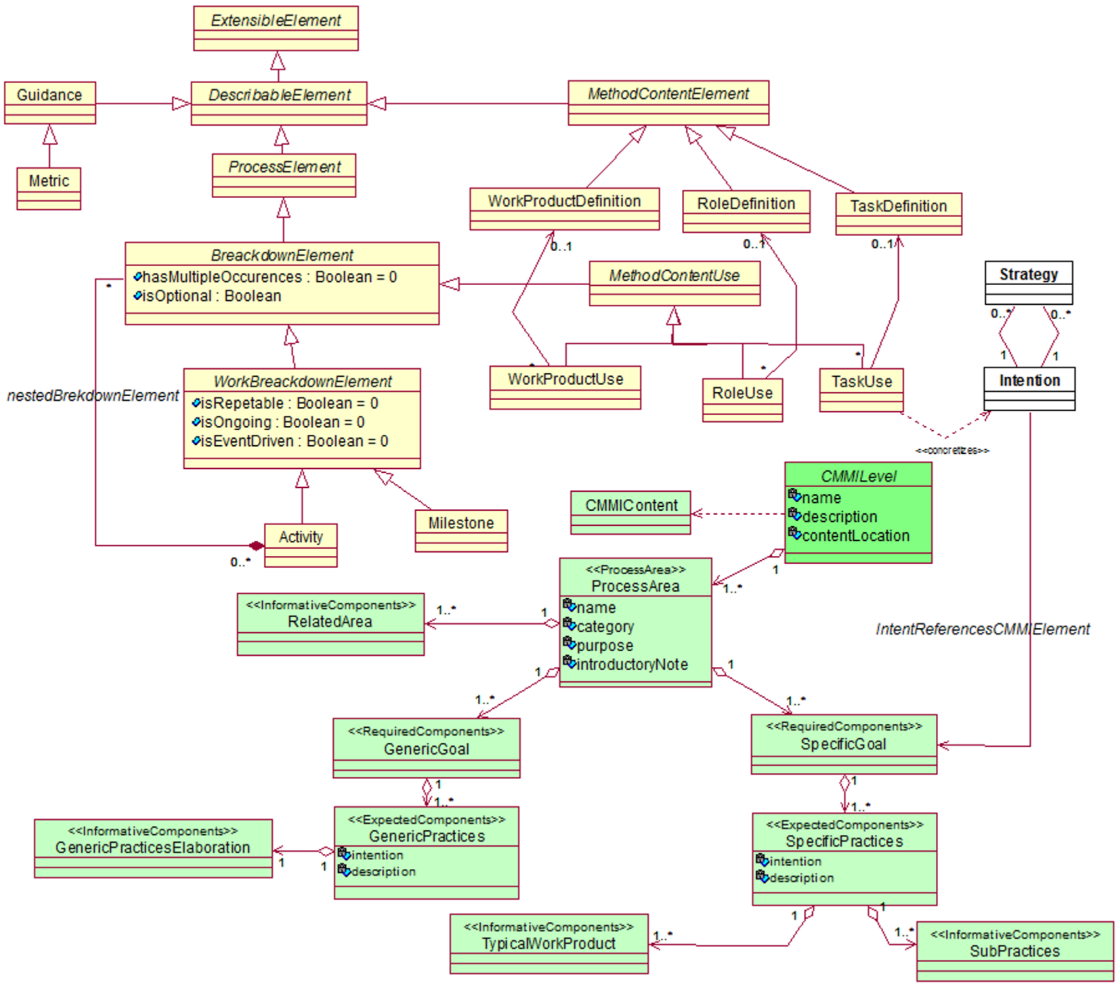


Figure 2: Extension of SPEM Meta-Model to Integrate CMMI Content

Continuous evaluation of software process models. The process assessment is made automatically and at a conceptual level. Such a continuous evaluation mitigates the concerns related to reliability of measures since we are able to repeat the measurements several times which is not the case with a questionnaire for example. El Emam and Madhavji [5] have claimed that a perfect measure of maturity Questionnaires-based can never be really known due to the empirical difficulties to repeat assessments of the same organization. The authors point out the question of how reliable such assessments are, and how reliable the instruments of measurement are. Moreover, CMMI maturity refers to a construct value that is indirectly measured by considering the values of the various items expected. In this work, we state that an automatic assessment could be a reliable measurement instrument, and should be based directly on the conceptual view of process models. In addition, the ability to produce indicators for processes assessment opens up the possibility of exploring new metrics. For example, indicators to measure the degree of alignment with CMMI practices such as distance from a situation qualified as acceptable.

Overview of a new landscape based on the propagation of CMMI practices. The proposed dashboard provides a landscape of alignment. This landscape allows the analysis of the propagation of CMMI practices through the work breakdown structure. Hence, we might consider a view of process models based on the flow of crucial practices and their sequences. For example, we could consider a practice breakdown structure. In addition, generated dashboard supports tradeoffs analysis between different scenarios to adapt a process model to the context of use.

V. Applying Alignment Strategy in Real World Processes

To validate to what extent the alignment strategy gives useful insights regarding process analysis and improvement, we designed two studies. The first one was conducted in an academic context (section A) with the aim to quantitatively evaluate the various factors impacting the effectiveness of our approach by comparing participants' results on mapping one prescriptive process model to CMMI with a reference result obtained from three experts on both SPEM and CMMI. The second case study (section B) evaluate the capability of Scrum with respect to Project Management area of CMMI.

A. Academic Study

For the academic study, we were interested in responding the following questions:

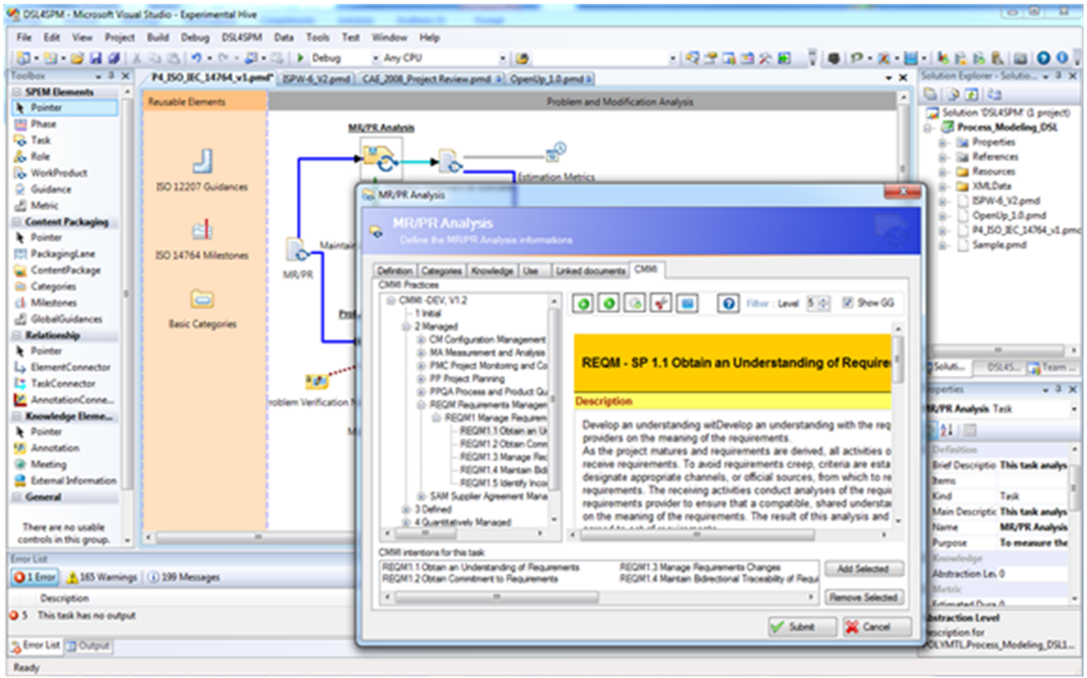


Figure 3: Screen-shot of the specific form to align process activities to CMMI practices

- *RQ1: How do developers interpret the activities of a process model, defined using SPEM 2.0, in terms of CMMI specific practices?*
- *RQ2: What is the overall quality of the mapping results as measured by the number of correct mapping selected for each process activity?*
- *RQ3: What impact would the mapping results on awareness about the process improvement?*

1) Experimental Design: The academic case study involved 34 undergraduate senior students paired in 17 groups. All the participants were enrolled in a mandatory project-oriented course in software engineering and are also enrolled in the Software Engineering Processes course. Students have learned and practiced the core area components of software process modeling and enactment. The teaching format consists of a weekly lecture, in a theater style with exercises (3 Hours), along with a bi-weekly laboratory session (lasting 3 Hours) usually focusing on specific concepts. The course covers the acquisition of the following skills:

1) S1: Students must understand the importance of process models;
2) S2: Students must understand the integration of process models into an organizational environment;
3) S3: Students must be able to analyze realistic process models in different size and complexity related to different size of organizations and projects;
4) S4: Students must be able to compare different process modeling approaches;
5) S5: Students must learn the most relevant existing process models (Scrum, OpenUP, RUP) and must be aware of different corpus of practices (such as CMMI, ITIL, Swebok).

Students use an adaptation of the ISPW-6 process model [11] to carry out a real software development project. The adapted process is composed of seven activities: *(1) Schedule and Assign Tasks*, *(2) MR/PR analysis*, *(3) Monitor Progress*, *(4) Modify Design*, *(5) Review Design*, *(6) Modify Code*, and *(7) Modify Test plan*.

In order to simplify the setting of this case study, we use only four CMMI process areas from level 2 and four from level 3 (using the filtering features), which constitute 66 specific practices (out of 165 specific practices). The restriction to these specific practices was selected for practical reasons: the prerequisite level of projects in which students were involved, and pedagogical requirements of the curriculum.

2) Results: The results show that the developers can easily align process activities with CMMI PAs. While at a fine-grained level, we observed some confusion that occurs when mapping process activities with certain CMMI specific practices. For instance, a single CMMI PA can be operationalized into multiple process activities; a single activity can respond to the needs of multiple PAs; one activity's intention can hide that of another.

Some CMMI practices have to be operationalized in a particular order (e.g., context dependency). For example, we observed a strong dependency between performing Verification/Validation and performing Analysis of the results. We must therefore ensure that both Verification/Validation and analysis activities are adequately expressed and ordered. Finally, most of the participants reported that the strategy of mapping allows a careful examination of process activities, which can help to guide the development of effective process models.

B. SCRUM CASE STUDY

To gather preliminary evidence that our approach produces similarly high-quality results when applied in industrial contexts, we performed a second case study based on action research method [2]. The study takes place within a North American company specialized in online transactions banking. The development teams use the well-known Scrum as the primary process model, while the maintenance teams use KANBAN for their daily bug fixing tasks. Both the author and the company were engaged with the intention of improving the development process in place. This case study was selected to complement the first one by involving professionals. The main goal of this study was to identify opportunities for improving the current implemented Scrum practices using CMMI V1.3 as a reference.

1) Study Context: The case study application reported here was performed within the context of a software consultancy firm based in North America (referred to as company P). The company P uses heavily the Scrum process model [20] and has a wide experience with Test Driven Development practices (TDD). Since Scrum mainly covers project management areas, the assessment focuses on PAs: "*Product Integration*" and "*Project Management*". Because the company P adopts agile environment for herself and her clients, she has established a constraint related to time of intervention and the use of non-invasive instruments. Time constraint dedicated to process improvement initiatives must be as shirt as possible, helping this way teams to focus on their business goals instead of wasting time on process analysis. Non-invasive constraint means using conceptual analysis instead interviews or questionnaires. The indicators of measurement developed in the context of this study would also be used as part of the process analysis and improvement services that the company provides.

2) Objectives: This analytic study aims at assessing the maturity of an agile process model. We chose Scrum because it has attracted a significant attention amongst practitioners. Our intent is not only to assess missing practices for compliance with CMMI goals and practices, but also to bring evidence of the overall impression regarding cultural differences due to the prescriptive nature of the CMMI [26]. The question being investigated was about coexistence of the CMMI and agile communities, and the answer was at a conceptual level.

The second experiment complements the first one in an industrial context. The purpose of the study was to evaluate our tool for providing a general measure of an agile organization maturity, and to use the generated scorecard for analyzing the capability of the actual practices according to CMMI model.

3) Alignment Analysis: Since Scrum emphasizes managing software projects, we focus only on CMMI PAs under areas Project Management plus Product Integration PA. The first covers the project management activities related to planning, monitoring, and controlling the project [22]. The second area covers practices related to the best possible integration sequence from the integration of product components to the delivery of product to the customer. These categories cover:

- Level 2: Project Planning (PP), Project Monitoring and Control (PMC), and Supplier Agreement Management (SAM).
- Level 3: Integrated Project Management +IPPD (IPM+IPPD); Risk Management (RSKM) plus Product Integration (PI).
- Level 4: Quantitative Project Management (QPM).

4) Results: Figure 5 shows the results of alignment. For example, Scrum covers 64% of CMMI practices enclosed in project planning PA, it covers 56% from the practices recommended by CMMI for product integration (PI). From the result of alignment, we can conclude that Scrum does not cover all the expected practices for the project management PA nor for the product integration PA.

Figure 5a summarizes the detail related to the alignment of each activity within Scrum. For each activity, we report the number of practices identified as mapped to one of process areas. The last line presents the average of coverage for each CMMI process area. For example, the activity called "*Sprint Planning Meeting*" is mapped to 1 CMMI specific practices (among 14) related to "*Project Planning*" PA. Also, as shown the CMMI PA called "*Product Integration*" is covered at 56% by only one activity called "*Build Increment*", which embodies 5 references to specific practices.

Apart from the cultural differences due to the prescriptive nature of CMMI, our analysis of alignment reveals that there is a significant cultural difference among Scrum and CMMI. Scrum is aligned according to one dimension (i.e., Project Management area), but it brings other dimensions that are not covered by CMMI such as practices to share tacit knowledge, short iterations and customer on the site for a continuous validation. However, this issue makes the integration a promising target for the efficiency of the Scrum process model.

Scrum is aligned according to one dimension oriented toward CMMI Project Management area. However, it brings other dimensions that are not covered by CMMI such as practices to share tacit knowledge, short iterations and customer on the site for a continuous validation. This issue makes the complementarity a promising target for the efficiency of the Scrum process model. We believe that establishing an explicit alignment between software processes and CMMI specific practices might have the following consequences:

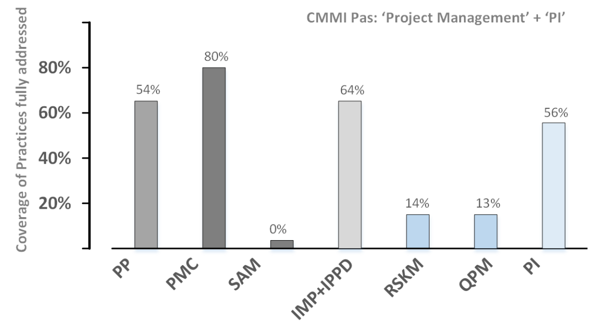
The rationale behind process activity would be better understood- The addition of CMMI intentions when modeling or maintaining a process model would complete the description of activities. Such an enriched description would lead to a better understanding of the rationale behind each activity according to CMMI concerns, and could, we believe, improve project performance in terms of the achievement of the strategic goals defined by the organization.

The success rate of SPI initiatives would increase- To be successful, SPI practitioners have to understand how the SPI approach is to be implemented within a specific context of process model [24], [6]. Even when the SPI initiatives are seen through the lens of developers, their purposes in terms of the overall CMMI objectives are not clearly understood. In order to provide a greater degree of confidence in how SPI initiatives are implemented, a tight alignment between the process modeling framework and the CMMI framework should be considered. Moreover, practitioners could use this conceptual view of alignment to identify the weaknesses in

Process Activities	CMMI Process Areas & Practices						
	Project Management						Product Integration
	PP (14)	PMC (10)	SAM (8)	IMP+IPPD (14)	RSKM (7)	QPM (8)	PI (9)
Sprint Planning Meeting	1	1	-	1	-	-	-
Estimating the product backlog	3	-	-	-	-	-	-
Prioritizing the backlog	4	-	-	-			-
Release Planning		2	-	-	-	1	-
The daily Scrum	3	3	-	3	1	-	-
Build Increment	-	1	-	-	-	-	5
Sprint Review Meeting	1	3	-	2	1	1	-
Sprint Retrospective	1	7	-	2	1	1	-
Coverage	**64%**	**80%**	**0%**	**64%**	**14%**	**13%**	**56%**

Values: number of specific practices identified. -: Not Applicable

(a) Quantification of Scrum Alignment



(b) Scrum Coverage for Project Management

Figure 5: Result of Scrum Process Alignment with CMMI

flows of practices, which would help focus future improvement effort and resources on promising avenues.

Alignment at the process-level would be conceptualized- Despite the fact that the content of the CMMI is a de facto standard, the operationalization of this content in a structured process model is not universal. Building an effective structure of software process is a challenging effort for organizations. Therefore, process modeling needs a tight connection to a chosen corpus of "best practices" such as CMMI. This connection should be created through one of the two following approaches [25]: at the process-level, by representing CMMI's intentions within the description of activities, enabling developers (as users of the process model) to understand the rationale beyond the description of activities, their scale and scope, and the expected results; or, at the content-level, by operationalizing the CMMI practices. With one of these approaches in place, we could build a coherent activity breakdown structure from the bottom up. These approaches need to be extended to both the SPM and the CMMI metamodels.

Process maturity would be assessed- The main use of CMMI remains the assessment of process capability [13]. Organizations identify the weaknesses of a process model and make conscious decisions based on assessment scores. However, El Emam and Madhavji [5] have stated that a perfect measure of maturity can never be really known, since it is empirically difficult to repeat assessments of the same organization. Moreover, maturity is considered by CMMI as a construct, which is indirectly measured by considering the values of the various items expected. We claim that an automatic assessment could be a reliable measurement instrument, and would be based on a process model. Consequently, SPM should support, through improved tools, more views regarding the assessment of the maturity and capability of process models. For example, one of the distinctive features of the DSL4SPM tool lies in its use of CMMI indicators to automatically generate a visual display of the most important information needed to achieve a maturity level). This kind of dashboard displays what CMMI practices are covered by what activity. The views could also be used to focus on the flow of CMMI practices throughout the process activity breakdown structure.

VI. Threats to Validity

The external validity of our study was limited by the fact that we mapped only one corpus of practices namely CMMI to our process modeling framework, which means that it might not generalize to other process models. However, the method and the research strategy could be replicated to other corpuses. We argue that Scrum process model is known enough to be of interest. For integration results, three Team leads reviewed our results and concluded that the mapping between their current practices within Scrum and CMMI was mostly complete. Though we used experts to validate the correctness of our results, it is still a human judgment. Finally, even if our approach usually provides correct results for conceptual mapping, we can only make hypotheses about how process managers would use our tool. Mapping other corpus such as SWEBOK is a natural next step in the enhancement of our strategy.

Furthermore, we claimed a multi-model integration, while we use only CMMI corpus. The fact is that other corpus that we have studied exhibited a wide variety of structures, concepts, and terminologies. However, our alignment approach is based on layered architecture to allow more flexibility when using another corpus (structures).

VII. Conclusion and Future Work

This paper presented an approach supported by a tool for alignment of SPEM-based software process models with corpuses of engineering practices such as CMMI. The major premise of our proposal is to harmonize the introduction of new practices within an existing flow of software process activities. The approach extends the SPEM at the meta-model level, presents a compact version of the corpus content using meta-data, and provides dashboards to assess the maturity and capability of process models according to CMMI scales. The approach was fully implemented as a perspective of process modeling tool. We have shown, by means of two analytic studies, that the alignment approach can be successfully used to analyze ISPW-6 and Scrum process models respectively in academic and industrial contexts.

As a proof of concept, we demonstrate how to align an existing process model with CMMI corpus. One major contribution of the proposed method pertains to the ability of continuous assessment of process model, at a conceptual level. We found that a continuous assessment based on a conceptual view allows an effective analysis of the CMMI practices propagation throughout the entire breakdown structure of activities. For instance, the generated dashboard help to focus on potential software process improvement effort and resources.

There are many interesting avenues for future work. For instance, extending the method to demonstrate the alignment with other corpuses such as ITIL, SWEBOOK, and PMBOK guide would be a natural next step. Furthermore, it would also be worthwhile to think about alignment with quality models such as ISO/IEC 9126. In summary, our paper makes contributions to a relatively new line of work on combining process models and alignment between them. We hope that future work will draw on our insights to enhance studies on multi-model environment and make the combination of the well established models more user-friendly. For CMMI alignment perspective, the work continues toward more refined indicators typology in our dashboards. These indicators might help organizations to achieve an expected level of maturity.

ACKNOWLEDGMENT

This work has been partly funded by the BPRDI NSERC program of Canada. It builds fundamentally on the multi perspective framework for software process modeling (DSL4SPM). We also would like to thank the participating staff at payza.com especially the architects for their enthusiasm.

REFERENCES

[1] D. Avison, J. Jones, P. Powell, and D. Wilson. Using and validating the strategic alignment model. *The Journal of Strategic Information Systems*, 13(3):223–246, 2004.

[2] D. E. Avison, F. Lau, M. D. Myers, and P. A. Nielsen. Action research. *Commun. ACM*, 42(1):94–97, Jan. 1999.

[3] N. Baddoo and T. Hall. De-motivators for software process improvement: an analysis of practitioners views. *Journal of Systems and Software*, 66:23–33, 2003.

[4] T. Dyba. An empirical investigation of the key factors for success in software process improvement. *Software Engineering, IEEE Transactions on*, 31(5):410–424, May 2005.

[5] K. El Emam and N. Madhavji. The reliability of measuring organizational maturity. *Process Improvements and Practices*, 1:3–25, 1995.

[6] K. El-Emam, B. Smith, and P. Fusaro. Success factors and barriers for software process improvement: An empirical study. In *Better Software Practice for Business Benefit: Principles and Experience*. IEEE Computer Society Press, Silver Spring, 1999.

[7] D. Goldenson and D. Gibson. Demonstrating the impact and benefits of cmmi: An update and preliminary results. Technical Report CMU/SEI-2003-SR-009 ed., Software Engineering Institute, 2003.

[8] R. Hefner and M. Sturgeon. Optimize your solution: Integrating six sigma and cmm/cmmi-based process improvement. In *Software Technology Conference*, pages 13–22, March 2002.

[9] J. C. Henderson and N. Venkatraman. Strategic alignment: Leveraging information technology for transforming organizations. *IBM Syst. J.*, 32(1):4–16, 1993.

[10] J. H. Iversen and O. K. Ngwenyama. Problems in measuring effectiveness in software process improvement: A longitudinal study of organizational change at danske data. *Int J. Information Management*, 26(1):30–43, 2006.

[11] M. Kellner. Ispw-6 software process example. In *6th International Conference on the Software Process*, pages 176–186, 1991.

[12] N. Kerzazi and P. Robillard. Multi-perspective software process modeling. In *Software Engineering Research, Management and Applications (SERA), 2010 Eighth ACIS International Conference on*, pages 85–92, May 2010.

[13] S. M and M. Niazi. Systematic review of organizational motivations for adopting cmm-based spi. *Information and Software Technology*, 50(1):605–620, 2008.

[14] D. Malzahn. Ontology based multi-model process improvement for business organizations. In *SEPG Europe*, pages 13–22, March 2008.

[15] L. V. Manzoni and R. T. Price. Identifying extensions required by rup (rational unified process) to comply with cmm (capability maturity model) levels 2 and 3. *IEEE Trans. Softw. Eng.*, 29(2):181–192, Feb. 2003.

[16] A. Marcal, B. de Freitas, F. Furtado Soares, and A. Belchior. Mapping cmmi project management process areas to scrum practices. In *Software Engineering Workshop, 2007. SEW 2007. 31st IEEE*, pages 13–22, March 2007.

[17] B. Mutafelija and H. Stromberg. Mappings of iso 9001:2000 and cmmi version 1.2. Technical Report www.sei.cmu.edu/cmmi/casestudies/mappings/pdfs/upload/ISO-9000-2000-mapping.pdf, SEI, white paper Accessed in april 2015, Apr. 2009.

[18] M. Paulk. Extreme programming from a cmm perspective. *IEEE Transactions on Software Engineering*, 18(6):19–26, Nov 2001.

[19] C. Rolland. A comprehensive view of process engineering. *IEEE Transaction on Software Engineering*, 1413(12):1–24, 1998.

[20] K. Schwaber. *Agile Project Management with Scrum*. Microsoft Press, London, 2004.

[21] SEI. The prime project. Technical Report http://www.sei.cmu.edu/process/research/prime-details.cfm, SEI, white paper Accessed in april 2015, Apr. 2008.

[22] SEI. *CMMI for Development, Version 1.3*. Software Engineering Institute, www.sei.cmu.edu/reports/10tr033.pdf, CMU/SEI-2010-TR-033. pp-469, Accessed May 2015, 2010.

[23] J. Siviy, P. Kirwan, J. Morley, , and L. Marino. Maximizing your process improvement roi through harmonization. In *Software Engineering Institute (SEI), Carnegie Mellon University*, 2008.

[24] D. Stelzer and W. Mellis. Success factors of organizational change in software process improvement. In *SOFTWARE PROCESS IMPROVEMENT AND PRACTICE*, pages 227–250, 1999.

[25] P. Tallon. A process-oriented perspective on the alignment of information technology and business strategy. *J. Manage. Inf. Syst.*, 24(3):227–268, Dec. 2007.

[26] R. Turner and A. Jain. Agile meets cmmi: culture clash or common cause? in extreme programming and agile methods - xp/agile universe. In *Second XP Universe and First Agile Universe Conference*, pages 153–165, 2002.