A SPEM BASED SOFTWARE PROCESS IMPROVEMENT META-MODEL

Rodrigo Santos de Espindola and Jorge Luis Nicolas Audy
Faculty of Informatics, PUCRS, Av. Ipiranga 6681, Porto Alegre, RS, Brazil

Keywords: Software Process Improvement, Meta-model, Integration.

Abstract: Nowadays the organizations are using Software Process Improvement (SPI) reference models as the starting point to their quality improvement initiatives. There is a consensus that by understanding and improving the software process we could achieve the improvement of the software product as well. Several studies also indicate the concurrent adoption of multiples SPI reference models by the organizations. The need for new approaches to integrate those SPI reference models with each other and with the software process developed aiming compliance with them has increased. This paper propose a SPEM based SPI meta-model as a way to support those kinds of integration.

1 INTRODUCTION

The modern organizations are working in an environment of rising competition. Every day new technologies, markets and competitors emerges. Then, the organizations are continuously looking for solutions to improve their products and processes.

In the case of software engineering industry, several approaches were proposed for the software process improvement (SPI). One of those approaches is the adoption of SPI reference models like ISO 15504, ISO 9003, ISO 12207, CMMI and MR-MPS. Despite the different terms used to refer to those SPI reference models (like standards, quality models, reference models, etc.) they are being called in this paper by the general term of SPI reference models in order to ease the discussion and improve the paper readability. These SPI reference models are important because there is a consensus that by understanding and improving the software process quality we could achieve the improvement of the software product as well (Rocha, 2001), (Sommerville, 2003), (Pressman, 2004) and (Schulmeyer, 2008).

Several criteria can be used in order to choose which SPI reference model is better appropriate to a particular organization and its goals. However, there are no impeditive of using more than one SPI reference model in the same organization or SPI initiative. In fact, there are several papers (Sallé, 2004), (Edgeman, 2005), (Cater-Steel, 2006), (Mingay, 2006), (Espindola, 2009) and (Espindola, 2009b) about the concurrent adoption of multiples SPI reference models by the organizations. Those papers also discuss the issues emerging in this context and the possibilities of integration of the SPI reference models.

This paper presents an approach to integrate different SPI reference models in a way not disruptive with the current process engineering practices. The goal is to create a SPI meta-model that allows representing both the SPI reference models and the process models developed based on those SPI reference models. In order to do that we propose a meta-model based on an extension of the Software Process Engineering Meta-Model version 2.0 (SPEM) (OMG, 2007).

This paper contributes to the research in SPI field proposing a way to raise the formalism used to represent the SPI reference models and the integration among them. Contributes to the industry deploying a meta-model useful for the creation of software process engineering tools that cover the SPI initiatives too.

Study developed by the Research Group of the PDTI 001/2010, financed by Dell Computers of Brazil Ltd. with resources of Law 8.248/91.

Santos de Espindola R. and Luis Nicolas Audy J. (2010).
A SPEM BASED SOFTWARE PROCESS IMPROVEMENT META-MODEL.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 301-306
DOI: 10.5220/0002910503010306

2 RELATED WORKS

There are several SPI reference models adopted by the industry. The following studies are related to the integration of quality models.

In (Pickerill, 2005) a relationship between IDEAL (developed by SEI) and Six Sigma is demonstrated. Using IDEAL as reference model, this works proposes the usage of both Six Sigma implementation methods (DMADV and DMAIC) to develop and implement process with CMMI.

(Siviy and Hallowell, 2005) has complemented this research, evaluating the usage of Six Sigma as a facilitator on CMMI implementation. The conclusions demonstrated that the implantation process and ROI verification have been accelerated.

(Rout, Tuffley and Cahill, 2001) presents a technical report that evaluates the compatibility between CMMI and ISO/IEC 15504-2. As a result, a mapping table is presented and the report states that the ISO/IEC 15504-2 significant elements are addressed by CMMI.

A definition of a meta-model to integrate CMMI and ISO/IEC 15504 is presented in (Lepasaar and Mäkinen, 2002). The meta-model was applied in both models to identify the existing structures. That study distinguishes from other studies by proposing a meta-model to support integration. As stated in (OMG, 2005), (OMG, 2006), and (OMG, 2007), meta-models are utilized to support integration of processes, workflows, tools, database, and middleware's.

3 SPEM BASED SPI META-MODEL

In order to begin the analysis of the SPI reference models concepts and determine which elements should be included in the meta-model, we have chosen the ISO/IEC 15504. The ISO/IEC 15504, as known as Software Process Improvement and Capability dEtermination (SPICE), defines a reference model for software processes and process capabilities that forms the basis for software process assessment (ISO, 1998). Nowadays there are several SPI reference models based on the ISO/IEC 15504 specifications. Thus, dealing with the ISO/IEC 15504 concepts makes the meta-model applicable to all SPI reference models based on ISO/IEC 15504, like CMMI (SEI, 2006) and MR-MPS (Softex, 2009).

Besides, in order to build up the SPI meta-model we have used the approach proposed by (Espindola, 2009b). That method aims to develop a SPI meta-model from the scratch. In this paper we followed a similar method, but started from the SPEM in order to allow a better integration between the SPI reference model representation and the software process representation.

In this section, the SPEM based SPI meta-model is described in three steps. First, we describe the meta-model architecture. Second, we describe the packages of the SPEM extension. And third, the meta-model elements are detailed.

In this paper, due to space reasons, we only present the main contributions, despite the whole meta-model includes more elements.

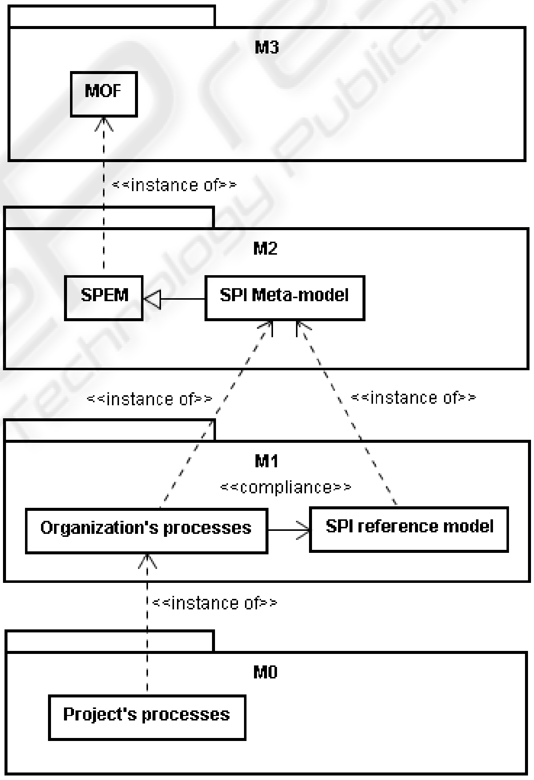

Figure 1: Proposed SPI Meta-model in the OMG modelling architecture.

3.1 SPI Meta-model Architecture

Since the proposed SPI meta-model is a SPEM extension, the OMG modelling architecture [(OMG, 2005), (OMG, 2007) and (OMG, 2007)] is the natural choice for the SPI meta-model architecture description. Figure 1 shows how the SPI meta-model relates to SPEM and the OMG modeling architecture.

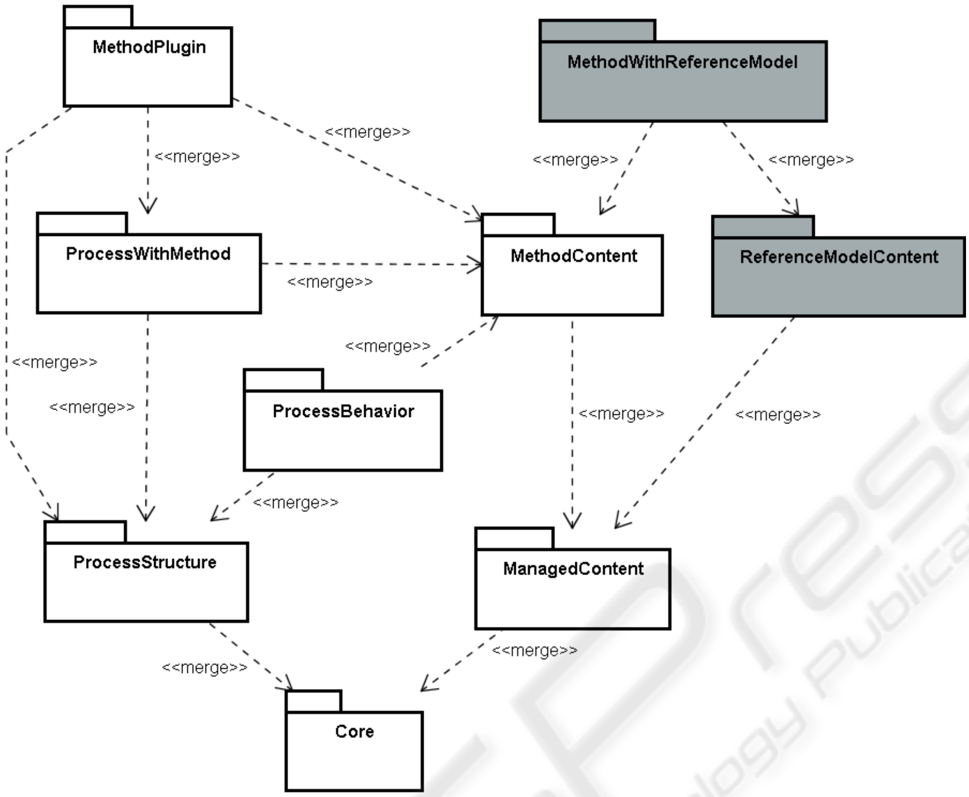

Figure 2: SPI Meta-model package diagram.

As illustrated in figure 1, the process enacted in a specific project is in the object layer. Each project follows his processes, since each project is a new effort, has its own goals and has its own characteristics. But, even being totally different efforts, all the projects follow processes tailored from the same organization’s standard processes. This assumption is true at least in organizations adopting some SPI reference model, which are in the scope of this papers discussion. While the organization’s standard processes define the overall projects behavior, in several aspects it still defines just the processes attributes and not their actual values (e.g.: roles, artifacts and activities). Since the project’s processes define the actual values for those attributes (real people, real docs and real tasks), we can characterize those project’s processes as instances of the organization’s standard processes.

An organization engaged in an SPI program defines its organization’s standard processes following one or more SPI reference models and aims for compliance between the organization’s standard processes and the SPI reference models. Both the organization’s standard processes and the SPI reference models are just models. Them, cannot be actually enacted in a real project because of their lack of actual information about what should be done, who should do the job and what should be the results. In other words, they don’t define actual values but just the attributes. In this case both the organization’s standard processes and the SPI reference models are in the same layer, the model layer.

Finally, the proposed SPI Meta-model belongs to the M2 layer, the meta-modelling layer. Nowadays, the organization’s processes can be defined as instances of the SPEM meta-model. But the SPI reference models can’t be defined in the same way, because SPEM don’t have all the appropriate concepts. From a process engineering point of view, that creates a gap between what we could do in terms of defining a process and defining a SPI reference model. Besides, using different approaches to define processes and to define their reference models also doesn’t help the integration between them. Thus, we propose a SPEM based SPI meta-model. By extending SPEM we aim to create a SPI meta-model that could be used for both the processes representation and the SPI reference models representation, in an integrated way. Since

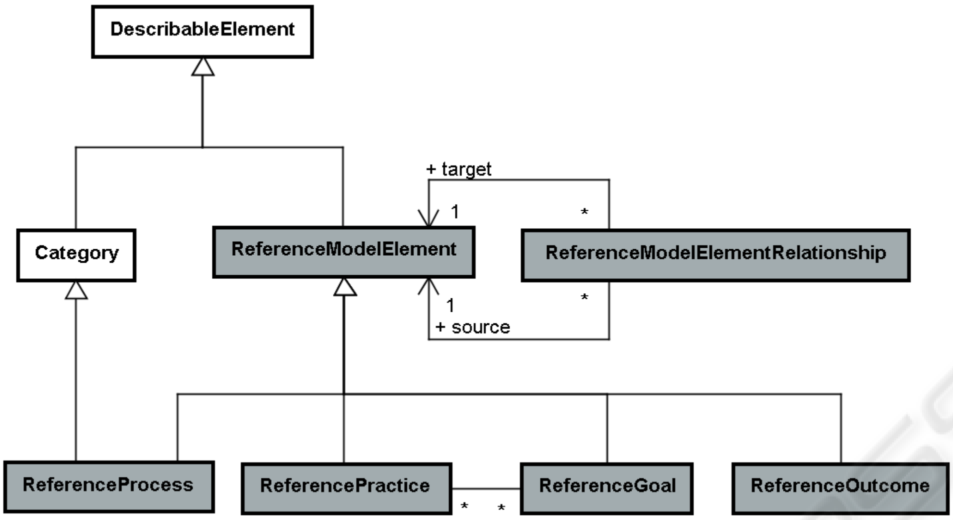

Figure 3: *ReferenceModelContent* class diagram.

this SPI meta-model is a SPEM extension and so inherits all its characteristics, we can define the processes, its reference models and the relationships between them as instances of the same meta-model. Of course the organization's processes defined in this way stay compatible with SPEM, but the new SPI meta-model aggregates the concepts required to deal with SPI reference models that are not present there today.

3.2 SPI Meta-model Packages

The figure 2 shows, in a package level, the extension done. The white packages are the original SPEM 2.0 packages. The gray packages are the packages added in order to deal with SPI reference models concepts and the integration between the original SPEM's process engineering concepts and the SPI reference models concepts. The same coloring schema is used in the other diagrams of this paper section.

The two packages are using the same package merge mechanism used in the SPEM construction. This mechanism allows to gradually build up the meta-model providing optional building blocks. Using this mechanism, concepts defined on a lower layer package, from the package merge perspective, can be extended in higher layer packages with additional properties and relationships to realize more complex modeling requirements (OMG, 2007).

The first package added to the SPI meta-model is the **ReferenceModelContent** package. That package introduces the concepts required to the SPI reference models representation. The package uses the concepts of the *ManagedContent* SPEM's package that provide the concepts required to the textual representation and documentation of any concept defined in the SPEM meta-model. The **ReferenceModelContent** package is detailed described in the section 4.3 of this paper.

The second package added to the SPI meta-model is the **MethodWithReferenceModel** package. That package introduces new concepts and changes other concepts already existing in the SPEM in order to allow the representation of the integration between the concepts of the *MethodContent* SPEM's package and the concepts introduced in the **ReferenceModelContent** package. In other words, this package deals with the associations between the process concepts and the reference models concepts allowing, in this way, the meta-model users to deal with compliance concerns. This package is detailed described in the section 4.4 of this paper.

3.3 ReferenceModelContent Package

The **ReferenceModelContent** package introduces the SPI reference models concepts. The figure 3 shows its elements.

The element used to represent the new concepts is the *DescribableElement*. A *DescribableElement* is an extensible element that represents an abstract generalization for all the SPEM's elements requiring textual documentation. Since the concepts required to represent a SPI reference model also need to be textually documented, then these elements are also extensions of the *DescribableElement*.

Another characteristic shared by these elements is the need for some mechanism that allows representing the equivalence relationship between elements belonging to different SPI reference models but having some degree of equivalence between them. For example, the process area called

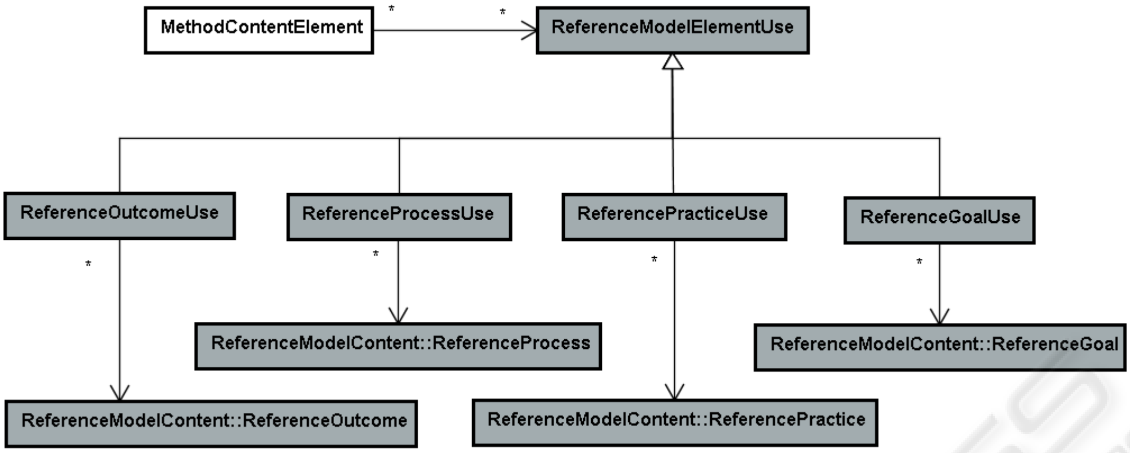
MethodContentElement
*
*
ReferenceModelElementUse
ReferenceOutcomeUse
ReferenceProcessUse
ReferencePracticeUse
ReferenceGoalUse
*
*
*
*
ReferenceModelContent::ReferenceProcess
ReferenceModelContent::ReferenceGoal
ReferenceModelContent::ReferenceOutcome
ReferenceModelContent::ReferencePractice

Figure 4: *MethodWithReferenceModel* class diagram.

Configuration Management is present in several reference models, like CMMI, MR-MPS and ISO 12207. Each one of this reference models have then a different instance of this specific *ReferenceProcess* meta-class, because this process area has not exactly the same content in each reference model, having particular characteristics in each one. But, besides the different characteristics, all this different instances represent the same or at least an equivalent concept in the modeling layer. In other words they are all Configuration Management process even having differences among their different reference models. Besides, this equivalence relationship can occurs between instances of different meta-classes. There is no impeditive of equivalence between a practice in one reference model and a process in another one. Therefore, we need some way in the meta-modelling layer to express the equivalence between instances of concepts that have any degree of equivalence between them in the modeling layer. In order to do that, two elements were introduced in this meta-model: the *ReferenceModelElement* and the *ReferenceModelElementRelationShip*.

The *ReferenceModelElement* is an extensible element that represents an abstract generalization for all the elements used to represent SPI reference models. In this way, all its specializations are capable of having equivalence relationships with other elements. The specializations are the following elements:

- *ReferenceProcess*: represents a process area used to group practices and goals;
- *ReferencePractice*: represents practices that must be done in the organization in order to achieve the goals of the process areas;
- *ReferenceGoal*: represents the goals and benefits that a process adoption intends to achieve;
- *ReferenceOutcome*: represents the results expected from the adoption of a process area.

3.4 MethodWithReferenceModel Package

The **MethodWithReferenceModel** package introduces the concepts required to the integration between the original SPEM's process engineering concepts and the SPI reference models concepts. The figure 4 shows its elements.

The *MethodContentElement*, from the *MethodContent* SPEM's package, is an element that represents an abstract generalization for all the elements in the *MethodContent* SPEM's package. Those elements are used to represents the content of a process. Being the generalization of all the elements used to represent process contents, the *MethodContentElement* is the ideal element to be changed in order to allow that process elements become capable of reference the SPI reference model's elements with which they aims for compliance. This change is done by adding a relationship between the *MethodContentElement* and a new element called *ReferenceModelElementUse.*

The *ReferenceModelElementUse* is a abstract generalization of all elements which a *MethodContentElement* extensions can use to reference any SPI reference model concept. Its specializations are:

- ReferenceOutcomeUse;
- ReferenceProcessUse;
- ReferencePracticeUse;
- ReferenceGoalUse.

Which one of those specializations aims to reference a specific element from the *ReferenceModelContent* package. A

ReferenceProcessUse is intended to reference *ReferenceProcess* and so long.

This strategy allows, for example, that an instance of a process called “Project Management” can be associated to an instance of a process area called “Project Management” of some SPI reference model through the use of an instance of the meta-class *ReferenceProcessUse*. In this way it would be represented the compliance goal of that process.

4 CONCLUSIONS

The diversity of SPI reference models being used by the organizations creates new challenges related to the concurrent adoption of them and their integration. This paper reports a result of a research aiming the development of approaches and tools to deal with SPI reference models integration.

From a theoretical point of view, this paper contributes with the software engineering and with the process engineering through a proposal of a meta-model to represent SPI reference models allowing their integration. It also contributes to those research fields through rising the formalism used to represent the SPI reference models.

From a practice point of view, this paper contributes to the software industry through a proposal of a meta-model that could support the development of new process engineering tools capable of support the reuse of process assets developed aiming compliance with multiples SPI reference models concurrently.

As future works, the author intends to develop a process engineering tool based on the meta-model and use that tool to support an experiment to validate the meta-model.

REFERENCES

Cater-Steel, A.; Tan, W. G.; Toleman, M. Challenge of Adopting Multiple Process Improvement Frameworks. *Proceedings of the 14th European Conference on Information Systems*, 2006.

SEI - Software Engineering Institute. “CMMI for Development, Version 1.2”. *Carnegie Mellon University*, 2006.

Edgeman, R.L.; Bigio, D.; Ferleman, T. Six Sigma and Business Excellence: Strategic and Tactical Examination of IT Service Level Management at the Office of the Chief Technology Officer of Washington, DC. *Quality and Reliability Engineering International, vol. 21*, 2005.

Espindola, R. S.; Luciano, E. M. ; Audy, J. L. N. An Overview of the Adoption of IT Governance Models and Software Process Quality Instruments at Brazil - Preliminary Results of a Survey. 2009, Proceedings of the Forty-Second Annual Hawaii International Conference on System Sciences. *Los Alamitos : IEEE Computer Society*, 2009.

Espindola, R. S.; Audy, J. L. N. An Evolutionay Approach for Quality Models Integration. In: ICEIS 2009 - International Conference on Enterprise Information Systems, Milan. *Proceedings of ICEIS*, 2009.

ISO - International Standard Organization. “ISO/IEC TR 15504-2:1998 Information technology — Software process assessment — Part 2: A reference model for processes and process capability”.

Lepasaar, M., Mäkinen, T, 2002. “Integrating Software Process Assessment Models using a Process Meta Model”. IEMC '02 - *IEEE International Engineering Management Conference, Vol 1, pp 224-229*.

Mingay, S.; Bittinger, S. Combine CobiT and ITIL for Powerful Governance. Gartner Inc, 2006.

OMG - Object Management Group. “Meta Object Facility (MOF) Core Specification”. 2006. On-line at http://www.omg.org/mof/ .

OMG - Object Management Group. “Software Process Engineering Metamodel version 2.0”. 2007. On-line at http://www.omg.org/spem/ .

OMG - Object Management Group. “Unified Modeling Language: Superstructure”. 2007. On-line at http://www.omg.org/uml/ .

Pickerill, J., 2005. “Implementing the CMMI in a Six Sigma World”. Software Engineering Institute - Carnegie Mellon University. http://www.sei.cmu.edu/cmmi/adoption/pdf/pickerill.pdf

Pressman, Roger S. “Software Engineering: a practitioner’s approach”. *New York: McGraw Hill, 6th ed., 880p*.

Rocha, Ana R. C.; Maldonado, José; Weber, Kival. Qualidade de Software - Teoria e Prática. *São Paulo, Prentice Hall*, 2001, 303pp.

Rout, T. P., Tuffley, A., Cahill, B, 2001. “Capability Maturity Model Integration Mapping to ISO/IEC TR 15504-2:19998”. Software Quality Institute - Griffith University – Austrália. http://www.sqi.gu.edu.au/cmmi/report/docs/MappingReport.pdf

Sallé, M. IT Service Management and IT Governance: review, comparative analysis and their impact on utility computing: *Hewlett-Packard Company*, 2004.

Siviy, J., M., Hallowell, D., 2005. “Bridging the Gap Between CMMI and Six Sigma Training: An overview and Case Study of Performance-Driven Process analysis”. Software Engineering Institute - Carnegie Mellon University. http://www.sei.cmu.edu/cmmi/adoption/comparisons.html.

Schulmeyer, G. Gordon. "Handbook of Software Quality Assurance". *Norwood, MA: Artech House, INC, 4th ed., 485pp*, 2008.

SOFTEX. “MPS.BR - Melhoria de Processo do Software Brasileiro – Guia Geral”. 2009. On-line at http://www.softex.br/mpsbr/ .

Sommerville, Ian. “Engenharia de Software”. *São Paulo, Addison Wesley*, 2003, 592 pp.