

Integrando Elementos de Múltiplos Modelos com um Metamodelo de Perfis de Capacidade de Processo

Edgar L. Banhesse^{1,2}, Clênio F. Salviano¹, Mario Jino²

¹Centro de Tecnologia da Informação Renato Archer (CTI)
Rod. Dom Pedro I (SP - 65), Km 143,6
Amarais - Campinas - SP - Brasil

²Universidade Estadual de Campinas (UNICAMP)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Cidade Universitária Zeferino Vaz - Campinas - SP - Brasil

edgar.banhesse@gmail.com, clenio.salviano@cti.gov.br,
jino@dca.fee.unicamp.br

Resumo. *MPS tem sido utilizada com sucesso nas organizações intensivas em software geralmente com base em um único modelo de capacidade ou de maturidade da capacidade, tal como o CMMI-DEV ou MR MPS.BR. Porém tem crescido a necessidade de utilização de múltiplos modelos de referência para melhoria de processo. As soluções atuais envolvem geralmente a criação prévia de um novo modelo que integra um determinado conjunto de modelos. Este artigo apresenta um Metamodelo de Perfis de Capacidade de Processo, que utiliza os conceitos da MDE para a integração dinâmica de elementos de múltiplos modelos. Apresenta também a racionalidade, especificação, estratégia de implementação e um exemplo de uso do metamodelo.*

Abstract. *SPI has been used successfully in software intensive organizations usually based on a single capability model or process capability maturity model, such as CMMI-DEV or MR MPS.BR. However, it has increased the need for utilization of multiple reference models for process improvement. The current solutions usually embrace the previous creation of a new model that integrates a set of specific models. This paper presents a Process Capability Profiles Metamodel that uses the concepts of MDE for dynamic integration of elements from reference models. It also presents the rationality, specification, strategy of implementation and an example of utilization of the metamodel.*

1. Introdução

Melhoria de Processo de Software (MPS) é uma abordagem sistemática para aumentar, por meio do estabelecimento e melhoria contínua dos processos, a eficiência e a eficácia de uma organização intensiva em desenvolvimento de software para melhorar os produtos de software (Unterkalmsteiner et al. 2011:1) (SEI 2010). A atual MPS tem sido dirigida nas organizações intensivas em software por ciclos de melhoria baseados num determinado Perfil de Capacidade de Processo. Este Perfil é de um único modelo de maturidade da capacidade, como, por exemplo, um nível de maturidade do modelo CMMI-DEV v1.3 - *Capability Maturity Model Integration for Development* (SEI 2010), ou do modelo MR MPS.BR v2011 - Modelo de Referência para Melhoria

de Processo do Software Brasileiro (SOFTEX 2011). O nível de Maturidade 2 do CMMI-DEV, por exemplo, é composto por 7 áreas de processo (Gerência de Requisitos, Planejamento de Projeto, Monitoramento e Controle de Projeto, Gerência de Configuração, Garantia da Qualidade de Processo e Produto, Medição e Análise e Gerência de Acordo com Fornecedores) cada uma no nível de capacidade 2. Um Perfil também pode um conjunto de processos em determinados níveis de capacidade de um modelo de capacidade, como, por exemplo, o modelo ISO/IEC 15504-5 v2005 (ISO/IEC 2005), ou o modelo ASPICE PAM v2.5 - *Automotive Software Process Improvement and Capability dEtermination Process Assessment Model* (SPICE 2010).

Como consequência do sucesso da MPS existe um conjunto de forças que demandam uma evolução da atual MPS para tratar novas demandas de melhoria. Entre estas forças existem: a diversidade de estratégia, o relacionamento entre modelo e processo, a generalização do software, a comoditização de processo, os princípios fundamentais dos modelos, a visão da ISO/IEC 15504 e a utilização de múltiplos modelos (Salviano 2009). Em relação aos princípios fundamentais, Card identifica que “apesar de ser uma força no estado da prática do setor de software, a área de melhoria de processo ainda não é um tópico popular de pesquisas mais rigorosas, principalmente nas universidades. Como a área tem sido impulsionada pela prática, os diferentes modelos e abordagens são considerados concorrentes, embora compartilhem os mesmos princípios básicos. A insuficiência de pesquisas nessa área tem limitado a identificação desses princípios básicos, que é papel da ciência” (Card 2004:238).

No intuito de contribuir com a necessidade levantada por Card no que se refere à identificação dos princípios fundamentais dos modelos, e de tratar a força referente ao uso de múltiplos modelos, este projeto de pesquisa apresenta uma proposta da criação de um Metamodelo de Perfis de Capacidade de Processo. Os modelos estão documentados em diferentes arquiteturas, com diferentes objetivos ou mesmo com base em diferentes conceitos para Melhoria de Processo de Software em organizações intensivas em software. Este trabalho é facilitado pelo fato dos modelos compartilharem alguns princípios básicos e possuem aspectos arquitetônicos semelhantes provenientes da ISO/IEC 15504.

O artigo está estruturado com mais sete seções. A seção 2 apresenta o contexto do projeto de pesquisa. A seção 3 apresenta o objetivo do artigo e a metodologia utilizada. A seção 4 apresenta as referências conceituais utilizadas no trabalho. A seção 5 apresenta a especificação das funcionalidades do metamodelo. A seção 6 apresenta um exemplo de uso do metamodelo. Por último, a seção 7 conclui o artigo.

2. Contexto

O Metamodelo de Perfis de Capacidade de Processo é um componente metodológico da metodologia PRO2PI (*Process Modeling Profile to drive Process Improvement* - Melhoria de Processo dirigida por Perfil de Modelagem de Processo). Esta metodologia está sendo desenvolvida pelo Centro de Tecnologia da Informação Renato Archer (CTI) desde 1999. Versões da metodologia foram aplicadas em organizações intensivas de software como parte integrante da metodologia. Na versão atual (4.0). Um dos objetivos da metodologia é orientar a melhoria de processo baseada em perfis de modelagem de processo de múltiplos modelos de referência. Um Perfil de Modelagem de Processo é composto por modelos de três tipos: Modelo de Perfil de

Capacidade de Processo, Modelo de Descrição de Realização de Processo e Modelo Indicador de Desempenho de Processo (Salviano et al. 2010).

Um Modelo de Perfil de Capacidade de Processo é um conjunto de processos ou áreas de processo em determinados níveis de capacidade, como, por exemplo, o processo ENG.1 - *Requirements elicitation* no nível de capacidade 2 - Processo Gerenciado da ISO/IEC 15504-5 v2005 (ISO/IEC 2005) e a área de processo REQM - *Requirements Management* no nível de capacidade 2 - Gerenciado do CMMI-DEV v1.3 (SEI 2010). Um exemplo de Modelo de Descrição de Realização de Processo é a descrição do processo de gerência de projeto de uma determinada organização, que inclui o ciclo de vida, papéis, atores e artefatos. Um exemplo de Modelo Indicador de Desempenho de Processo é a quantidade de não conformidades do processo de gerência de projeto em um determinado mês (Salviano 2011). O PRO2PI apóia a melhoria de processos usando elementos de múltiplos modelos de referência e de outras fontes. Estes elementos são selecionados ou definidos e integrados como um Perfil de Modelagem de Processo.

Para apoiar a integração de elementos de múltiplos modelos de referência, o metamodelo está sendo desenvolvido em três grandes fases. A primeira fase, que já foi realizada, focou na especificação de um conjunto de conceitos básicos unificados dos modelos relevantes de capacidade de processo para a unificação, generalização e modelagem de visões da arquitetura e de elementos de modelos relevantes (Salviano and Figueiredo 2008). Este conjunto foi especificado com um diagrama de classes em UML - *Unified Modeling Language*. O nome *Geraes* foi escolhido devido à semelhança com o desenho da capa do álbum *Geraes* de Milton Nascimento. *Geraes* v3.0 de 2008 é um componente metodológico do PRO2PI v3.0, que contempla os conceitos básicos de perfis de capacidade de processo e de modelos de capacidade de processo. Esta versão foi exercitada para integração de elementos dos modelos CMMI-DEV e do modelo ISO/IEC 15504-5.

O conjunto de conceitos básicos unificados, que são princípios fundamentais para perfis e modelos de capacidade de processo é a base para a criação de um metamodelo para melhoria de processo com múltiplos modelos. O Metamodelo de Perfis de Capacidade de Processo (MM de PCP) faz parte de um novo paradigma para melhoria de processo dirigido por modelagem de processo. Metamodelo nesse trabalho é considerado um modelo de uma linguagem de modelos de capacidade de processo, e tem por objetivo potencializar a utilização mais ampla de múltiplos modelos para a melhoria de processos de software.

O *Geraes* define quatro conceitos básicos baseados em Capacidade de Processo, que é a capacidade que um processo possui em satisfazer a qualidade de produto especificada, a qualidade de serviço, e os objetivos de desempenho de processo (SEI 2010). Os conceitos básicos são: Modelo de Capacidade de Processo, Área de Capacidade de Processo, Nível de Capacidade de Processo e Perfil de Capacidade de Processo. Modelo de Capacidade de Processo é proposto para representar todas as boas práticas de modelos de referência organizadas com o conceito de Capacidade de Processo. Área de Capacidade de Processo é um conjunto de boas práticas relacionadas especificamente a “o que fazer”. Nível de Capacidade de Processo é um conjunto de boas práticas relacionadas genericamente a “quão bem fazer”. Perfil de Capacidade de

Processo é um modelo de um processo sob o aspecto de capacidade de processo, como um conjunto de processos ou áreas de processo em determinados níveis de capacidade.

Na segunda fase, que está em desenvolvimento, estão sendo realizadas a especificação e implementação do metamodelo para gerenciar perfis e modelos de capacidade de processo, com ênfase na arquitetura dos perfis e modelos. Este artigo apresenta a especificação e estratégia de implementação desta fase do desenvolvimento. Com esta versão será possível separar o conteúdo de um perfil ou modelo de sua arquitetura. Desta forma qualquer conteúdo poderá ser automaticamente transformado para qualquer arquitetura. Isto facilita a integração de elementos de múltiplos modelos.

Na terceira fase haverá a evolução do metamodelo para enfatizar o conteúdo dos perfis e modelos. Nesta versão será possível relacionar conteúdos de múltiplos modelos em termos de graus de semelhanças e gerar novos elementos que sejam uma intersecção, união ou diferença de elementos de múltiplos modelos.

3. Objetivo do Artigo e Metodologia de Pesquisa

Este artigo tem dois objetivos principais. O primeiro objetivo está relacionado com a abordagem para o problema e o segundo está relacionado com a estratégia para implementação do metamodelo. A abordagem é apresentada por meio da descrição do contexto realizada na seção anterior e da descrição das referenciais conceituais, que é apresentada na próxima seção. A estratégia para implementação do metamodelo é apresentada por meio de considerações sobre a especificação da implementação e das funcionalidades do software.

Para orientar a condução desta segunda fase do projeto de pesquisa foi adotada uma metodologia baseada em classificações conforme proposto em (Silva and Menezes 2005). Os critérios utilizados para classificar o projeto foram: a natureza da pesquisa, a abordagem do problema, os objetivos, o método científico e os procedimentos técnicos. A Tabela 1 apresenta um sumário da metodologia utilizada no projeto, com os critérios, a classificação do projeto baseado nos critérios e uma justificativa para a classificação.

Critério	Classificação	Justificativa
Natureza	Pesquisa aplicada	Gerar conhecimentos para aplicações práticas e dirigidos à solução de problemas referentes à MPS
Abordagem	Pesquisa Qualitativa	O ambiente fornecido pelo software do MM de PCP é a fonte direta para coleta de dados e o pesquisador é o instrumento-chave
Objetivos	Pesquisa Exploratória	Envolve levantamento bibliográfico e consultas a pessoas especialistas em modelos
Método Científico	Hipotético-Dedutivo	Os conhecimentos disponíveis sobre a MPS a partir de múltiplos modelos são insuficientes e demandam por pesquisa. Para resolver o problema existe a necessidade de se conjecturar na forma de hipóteses
Procedimentos	Ciclos evolutivos (4 ciclos)	1º ciclo - ênfase nos estudos das bases conceituais, 2º ciclo - desenvolvimento do software MM de PCP, 3º ciclo - uso do software e o 4º ciclo - validação e documentação dos resultados obtidos

Tabela 1 - Metodologia do Projeto de Pesquisa do PRO2PI-MMC.

4. Referências Conceituais

Para complementar os resultados da primeira fase e definir a abordagem foram realizadas três atividades que definiram as referências conceituais: identificação de outras abordagens para tratar múltiplos modelos, revisão dos conceitos de MDE e uma revisão sistemática da literatura sobre como implementar MDE.

4.1. Abordagens para tratar Múltiplos Modelos

O principal desafio adicional de uma Melhoria de Processo de Software com Múltiplos Modelos em relação à atual Melhoria de Processo de Software baseada em um único Modelo, é a utilização integrada de elementos de múltiplos modelos que podem estar documentados em diferentes arquiteturas, com diferentes objetivos ou mesmo com base em diferentes conceitos. Com base na experiência em melhoria de processos foram identificados quatro tipos de abordagens para integrar elementos de múltiplos modelos. Estas abordagens foram denominadas: Integração de Modelos, Modelo Integrador, Mapeamento de Modelos e Framework de Modelos. As mesmas abordagens foram identificadas como iniciativas de melhoria de processo de software e com nomenclaturas diferentes na revisão sistemática da literatura sobre avaliação e medição de melhoria de processo de software em (Unterkalmsteiner et al. 2011).

Na abordagem **Integração de Modelos** a melhoria de processos é baseada em um modelo, que é construído a partir de outros modelos ou normas. Exemplo: o MR MPS.BR - Modelo de Referência para Melhoria de Processo do Software Brasileiro (SOFTEX 2011). A base técnica para a construção e aprimoramento deste modelo de melhoria e avaliação de processo de software é composta pelas normas ISO/IEC 12207 e ISO/IEC 15504-2 e pelo modelo CMMI-DEV.

Na abordagem **Modelo Integrador** um modelo é escolhido para guiar a melhoria de processo na organização, e por meio do uso deste modelo integrador garante-se que no momento apropriado, que outros modelos sejam incluídos no projeto de melhoria de processo da organização, respeitando as arquiteturas e os conteúdos de cada modelo utilizado no programa de melhoria (Mendes 2010).

Na abordagem **Mapeamento de Modelos** o mapeamento de modelos possibilita identificar sobreposições de abrangência, a cobertura, as semelhanças e diferenças existentes entre os modelos e padrões adotados pela organização para melhoria de processos. Essa visão, ou seja, a criação deste perfil facilita a implantação do programa de melhoria de processos na organização e facilita na avaliação da melhoria de processos, porém mapear modelos não é uma tarefa fácil, pois os modelos e padrões possuem arquiteturas, sintaxes e terminologias diferentes, que devem ser respeitadas (Thiry, Zoucas, and Tristão 2010).

Na abordagem **Framework de Modelos** uma arquitetura padrão é utilizada para organizar e combinar os componentes de modelos de um mesmo framework. Cada modelo do framework tem finalidades diferentes, usa a mesma arquitetura e já está preparado para ser utilizado de forma integrada com outros modelos do framework. Um exemplo é o CMMI (SEI 2010) e suas constelações. Constelação é uma coleção de elementos do CMMI, que incluem um modelo, seus materiais de treinamento e documentos relacionados à avaliação para uma área de interesse. Outro exemplo é a ISO/IEC 15504, que introduziu e consolidou conceitos relevantes, incluindo a

arquitetura contínua, o framework de modelos e a generalização de software para sistema (Salviano 2009).

Os modelos de capacidade são constituídos de boas práticas organizadas em suas respectivas arquiteturas, portanto podem ser considerados como repositórios de conhecimento, ou seja, meios utilizados para armazenar o conhecimento. Desta maneira, as abordagens podem ser comparadas em termos das características destas mídias. Armour define cinco características das quais três delas são mais adequadas para comparação das abordagens: **Frequência de Atualização** - quão rápido o conhecimento armazenado na mídia pode ser atualizado; **Intencionalidade** - quanto do armazenamento e da modificação de conhecimento na mídia pode ser feito de maneira deliberada; e **Habilidade de Auto-modificação** - quanto do conhecimento armazenado na mídia pode modificar a si mesmo (Armour 2004). Para realizar a comparação levou-se em consideração o grau de facilidade de cada abordagem (arquitetura e conteúdo) para trabalhar com múltiplos modelos. As abordagens foram pontuadas em uma escala com quatro conceitos: Muito Baixa, Baixa, Média e Alta. O resultado da comparação é apresentado na Tabela 2.

Características	Abordagens para Múltiplos Modelos			
	Integração de Modelos	Modelo Integrador	Mapeamento de Modelos	Framework de Modelos
Frequência de Atualização	Muito Baixa	Alta	Baixa	Muito Baixa
Intencionalidade	Baixa	Média	Média	Alta
Habilidade de Auto-modificação	Baixa	Baixa	Baixa	Baixa

Tabela 2 - Comparação entre as abordagens para Múltiplos Modelos e as características.

4.2. Engenharia Dirigida por Modelos

Segundo Rossini et al, “Engenharia Dirigida por Modelos - *Model-Driven Engineering* - MDE é um ramo da engenharia de software que visa melhorar a produtividade, qualidade e custo-efetividade de software, mudando do paradigma centrado em código para o centrado em modelo. MDE promove modelos e linguagens de modelagem como artefatos principais do processo de desenvolvimento e a transformação de modelos como a técnica principal para gerar (partes de) sistemas de software a partir de modelos. Modelos permitem que os desenvolvedores raciocinem em um nível maior de abstração, enquanto a transformação de modelos alivia os desenvolvedores de tarefas repetitivas e propensa a erros, tais como codificação” (Rossini et al. 2011:66).

A MDE está baseada em três conceitos fundamentais: sistema, modelo e metamodelo, e em duas relações básicas: conformidade e representação. A relação básica de representação surge por meio da relação dos conceitos fundamentais de sistema e modelo, em MDE diz-se que um modelo é a representação de um sistema. Já a relação básica de conformidade surge por meio da relação entre modelo e metamodelo, em MDE diz-se que um modelo está em conformidade com o seu metamodelo. Um modelo está em conformidade com um metamodelo, se e somente se, cada elemento do modelo tem seu metaelemento definido no metamodelo (Bézivin 2006).

Em MDE Sistema ou Sistema em Estudo é o objeto alvo da investigação. Existem três categorias de sistemas importantes neste contexto, que são: sistemas físicos, sistemas digitais e sistemas abstratos. Sistemas físicos são fenômenos ou elementos observáveis que pertencem ao mundo físico. Sistemas digitais são sistemas que residem nas memórias do computador e que são processados pelos computadores. Sistemas abstratos são idéias e conceitos que eventualmente residem na mente humana para serem processados pelos cérebros humanos (Favre 2004).

Modelo é uma abstração de um sistema em estudo para um determinado propósito (Favre 2004). Metamodelo define um acordo consensual sobre a forma como os elementos de um sistema devem ser selecionados para produzir um determinado modelo (Bézivin 2003). O metamodelo representa a relação entre um modelo e sua linguagem de modelos. Um determinado modelo M1 é um metamodelo de outro modelo M2 quando M1 é o modelo de uma linguagem de modelos, da qual M2 é um dos modelos pertencentes a esta linguagem, ou seja, um metamodelo define uma linguagem que é capaz de expressar modelos (Favre 2005).

Ao combinar os conceitos fundamentais e as relações básicas da MDE surge um padrão de conformidade. Um exemplo do padrão de conformidade é o relacionamento entre a língua inglesa, um dicionário de inglês e um texto escrito em inglês. O dicionário de inglês é um modelo da língua inglesa. Nesta relação à língua inglesa é o Sistema em Estudo. A língua inglesa pode ser definida em outra relação como uma linguagem de modelos da qual qualquer texto em inglês é um modelo que pertence a esta linguagem. Como consequência, o dicionário de inglês é um metamodelo do texto escrito em inglês. É importante notar que ser um sistema, modelo ou metamodelo não é uma característica intrínseca, mas sim uma função em uma relação.

Uma transformação entre modelos é a geração automática de um modelo destino a partir de um modelo de origem, segundo uma definição de transformação. Transformação é um conjunto de regras de transformação que juntas descrevem como um modelo na linguagem de origem pode ser transformado em um modelo na linguagem de destino. Regra de transformação é uma descrição de como uma ou mais construções em uma linguagem de origem podem ser transformadas em uma ou mais construções em uma linguagem de destino (Kleppe, Warmer, and Bast 2003). Existem quatro dimensões de transformações entre modelos, que são: a) Endógena - quando a transformação entre modelos é expressa na mesma linguagem; b) Exógena - quando a transformação é expressa usando linguagens diferentes; c) Horizontal - quando a transformação possui o mesmo nível de abstração; e d) Vertical - quando a transformação possui diferentes níveis de abstração (Mens, Czarnecki, and Gorp 2005).

4.3. Revisão Sistemática da Literatura

Uma Revisão Sistemática da Literatura (RSL) foi realizada seguindo o processo proposto por Brereton e Kitchenham (Brereton et al. 2006) e (Kitchenham 2007). O intuito da RSL foi o de apoiar e verificar a viabilidade da hipótese proposta para o projeto de pesquisa. Além de entender como um metamodelo pode ser desenvolvido e utilizado. O processo é composto por três fases: planejamento, realização e apresentação dos resultados da revisão sistemática da literatura. Segundo Petticrew e Roberts, uma RSL é mais adequada para o propósito de responder perguntas específicas e testar hipóteses do que uma revisão tradicional (Petticrew and Roberts 2006).

A questão do projeto de pesquisa é “Como potencializar a utilização integrada de elementos de múltiplos modelos, que estão documentados em diferentes arquiteturas, com diferentes objetivos e conceitos para melhoria de processo de software em organizações intensivas em software?”. Para resolver esse problema foi formulada a hipótese de pesquisa “A utilização de um Metamodelo de Perfis de Capacidade de Processo como um modelo de uma linguagem de modelos de capacidade de processo pode potencializar a utilização integrada de elementos de múltiplos modelos para a melhoria de processos de software em organizações intensivas em software, respeitando as características inerentes a cada modelo”. Desta forma, a revisão sistemática da literatura buscou responder uma questão elaborada com base na hipótese do trabalho de pesquisa: “Como desenvolver e utilizar metamodelos como um meio para resolver problemas no paradigma da engenharia dirigida por modelos (MDE)?”.

Como resultado da aplicação do protocolo da RSL nas fontes de pesquisa foi encontrado um total de 1542 artigos; sendo 878 da IEEE Xplore, 328 da ACM Digital Library e 336 da ScienceDirect. Uma verificação inicial foi realizada entre os artigos das três fontes de pesquisa; isso permitiu identificar e eliminar 50 artigos duplicados, o que resultou num total de 1492 artigos únicos. Em uma primeira etapa os títulos, os resumos, as introduções e as conclusões dos artigos foram revisados com base nos critérios primários de seleção e critérios de qualidade contidos no protocolo da revisão sistemática. Essa revisão resultou na seleção de 14 artigos, conforme apresentado na Tabela 3. Em uma segunda etapa, os artigos selecionados foram lidos e classificados de acordo com três grupos de utilização do metamodelo, que são: Criação de Modelos (1), Desenvolvimento (2) e Linguagem de Transformação/Modelagem (3).

Id.	Classificação	Nome do Artigo
1	1	<i>A Definition of "Abstraction Level" for Metamodels</i>
2	1	<i>A MDE Based Approach for Bridging Formal Models</i>
3	1	<i>A Meta-model for Developing Business-model Driven Management Information Systems</i>
4	1	<i>Definitions and approaches to model quality in model-based software development - A review of literature</i>
5	1	<i>Towards a Precise Definition of the OMG/MDA Framework</i>
6	2	<i>An approach for Model-Driven test generation</i>
7	2	<i>A Metamodel-based Approach For Automatic User Interface Generation</i>
8	3	<i>A Flexible Infrastructure for Multilevel Language Engineering</i>
9	3	<i>A Formal Model Driven Approach to Dependable Software Evolution</i>
10	3	<i>Bridging metamodels and ontologies in software engineering</i>
11	3	<i>CacOphoNy: Metamodel-Driven Software Architecture Reconstruction</i>
12	3	<i>Metamodels in Europe: Languages, Tools, and Applications</i>
13	3	<i>Model-based DSL Frameworks</i>
14	3	<i>Model-Driven Development: A Metamodeling Foundation</i>

Tabela 3 - Artigos selecionados no processo da Revisão Sistemática.

A revisão sistemática da literatura permitiu responder a questão de pesquisa do protocolo da revisão sistemática, além de confirmar que a hipótese proposta para o projeto de pesquisa é viável. Isso foi constatado pelo fato dos metamodelos apresentados nos artigos serem desenvolvidos e utilizados como um meio, ou seja, como um instrumento na resolução de problemas no paradigma MDE com diferentes focos de utilização. Existe uma relação muito forte entre os três grupos, porém, apesar desta relação cada artigo tinha um foco diferente na utilização de metamodelo.

O grupo de Criação de Modelo trata do uso de metamodelos para criar modelos que podem ser utilizados em diferentes contextos. O grupo de Desenvolvimento trata do uso de metamodelos para desenvolver software e casos de testes. E por último o grupo de Linguagem de Transformação/Modelagem trata do uso de metamodelos para a criação de linguagens de transformação de modelos e linguagens de modelagem, que são utilizadas para transformar um modelo de origem em um modelo de destino para resolver um problema específico. O trabalho de pesquisa faz parte deste grupo.

5. Especificação do Metamodelo de Perfis de Capacidade de Processo

O componente *Geraes* foi evoluído e passou a ser denominado também como PRO2PI-MMC - *Process Modeling Profile to drive Process Improvement - MetaModel of Process Capability Profile* - Metamodelo de Perfil de Capacidade de Processo. A Figura 1 apresenta a versão 4.0 de 2012 do *Geraes*. O diagrama foi revisado na ferramenta de modelagem ArgoUML.

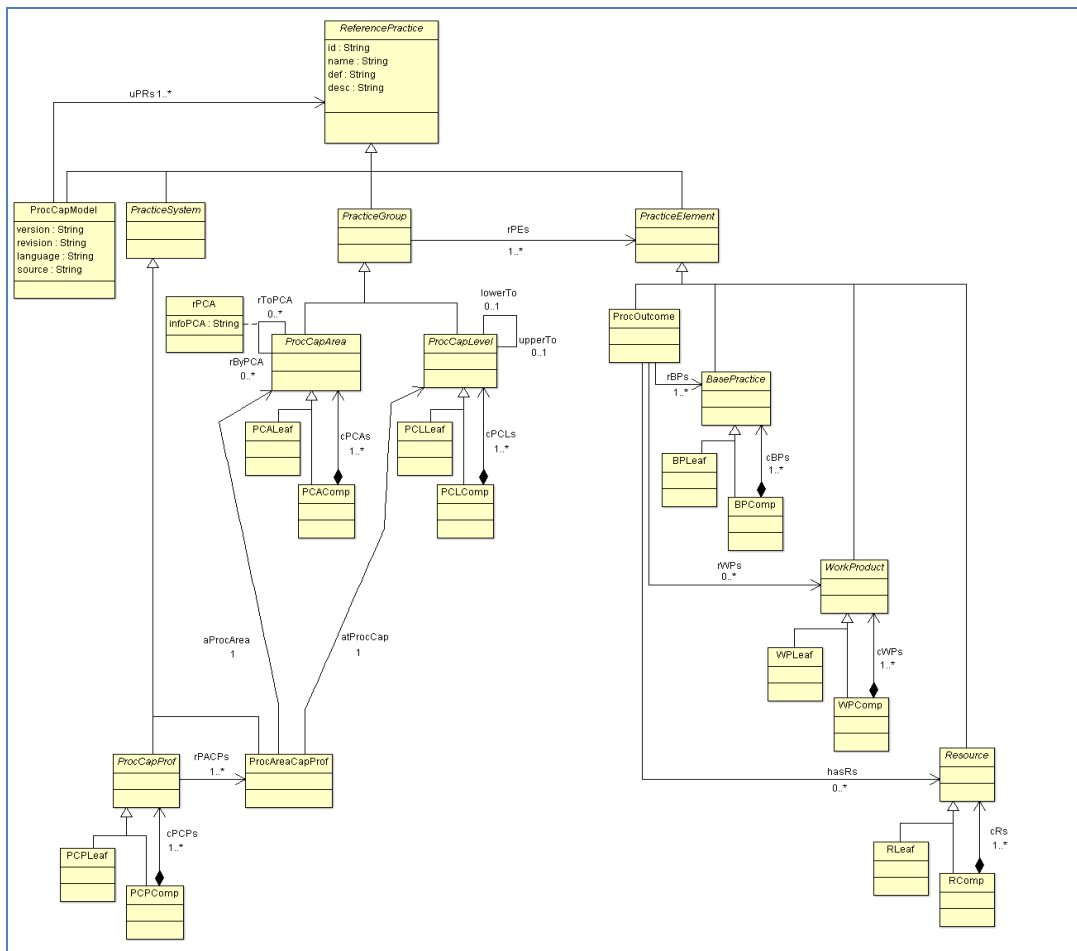


Figura 1 - Componente metodológico PRO2PI-MMC (*Geraes*).

Para validar o Metamodelo PRO2PI-MMC pensou-se em implementar um software, que tem o papel neste trabalho de ser uma prova de conceito, ou seja, ser um modelo prático que demonstra a viabilidade de um conceito teórico estabelecido por uma pesquisa, no caso o PRO2PI-MMC.

No paradigma da MDE o desenvolvimento de software é baseado na construção de modelos e metamodelos em ambientes computacionais e em transformações automáticas entre modelos para a geração de código executável. Porém estes ambientes ainda estão em evolução para uma utilização mais intensa pela indústria de software (Cruz 2010). Não foi encontrado um ambiente de desenvolvimento que suporte completamente de forma amigável todo o paradigma da MDE. Os ambientes de desenvolvimento atuais envolvem duas grandes atividades: a modelagem e a codificação. Como o foco do projeto de pesquisa é desenvolver um Metamodelo de Perfis de Capacidade de Processo e não um metamodelo para desenvolvimento de software, optou-se por desenvolver o software do metamodelo de forma convencional no paradigma orientado a objetos seguindo os conceitos da MDE.

O software está sendo implementado no framework de desenvolvimento CkePHP com o banco de dados relacional MySQL. As transformações de modelos são realizadas por meio da linguagem específica de domínio SQL - *Structured Query Language*. O software foi denominado PRO2PI-MMCA (*MetaModel Process Capability Profile Software Application*) e deve ser considerado como um protótipo. A definição da funcionalidade do software levou em consideração as relações do padrão de conformidade de Favre (Favre 2005) e os testes realizados na fase exploratória do projeto de pesquisa. As funcionalidades especificadas para o software estão relacionadas e descritas a seguir agrupadas em quatro funções, cada uma com uma série de operações. Nesta descrição o termo gerenciar significa um conjunto de cinco operações: cadastrar, consultar, editar, validar e remover. As funções e operações são:

1 **Representar arquiteturas** de modelos de capacidade de processo. São os metamodelos de modelos de capacidade de processo específicos descritos na linguagem do metamodelo de capacidade de processo.

- a. Gerenciar as informações de uma arquitetura de modelos.
- b. Gerenciar os elementos de uma arquitetura de modelos.
- c. Gerenciar os relacionamentos entre os elementos de uma arquitetura de modelos.

2 **Representar modelos** de capacidade de processo. São os modelos de capacidade de processo específicos descritos na linguagem do seu próprio metamodelo.

- a. Gerenciar as informações de um modelo.
- b. Gerenciar os componentes de um modelo.
- c. Gerenciar os relacionamentos entre os componentes de um modelo.
- d. Gerenciar um Perfil de Capacidade de Processo de um modelo.
- e. Gerenciar um Perfil de Capacidade de Processo de múltiplos modelos.

3 **Estabelecer regras de transformação adicionais** entre os elementos de arquiteturas dos modelos de capacidade de processo. São as correlações entre os elementos de arquiteturas de modelos, que possibilitam que uma arquitetura de um modelo de origem possa ser transformada em uma arquitetura de um modelo de destino. As regras são adicionais, pois o metamodelo de capacidade de processo já define algumas correlações entre os elementos de arquiteturas de modelos.

- a. Gerenciar uma regra de transformação entre os elementos de arquiteturas de modelos.

4 **Realizar transformações** de modelos de capacidade de processo e transformações de perfis de capacidade de processo. São as operações realizadas com base no metamodelo de capacidade de processo e com base nas regras de transformações adicionais entre os elementos de arquiteturas dos modelos.

- a. Apresentar uma área de processo/processo de um modelo de origem na arquitetura de um modelo de destino.
- b. Apresentar um perfil de capacidade de processo de um modelo de origem na arquitetura de um modelo de destino.

6. Exemplo de utilização do metamodelo

O exemplo se baseia nas funcionalidades do PRO2PI-MMCA para criar um perfil de capacidade de processo que pode ser utilizado para melhorar o processo de software de uma empresa de software. O cenário é uma empresa de software que gostaria de melhorar o seu processo atual de gerência de requisitos. O processo de gerência de requisitos está em conformidade com as práticas específicas da área de processo REQM - Gerência de Requisitos do modelo CMMI-DEV v1.3 (SEI 2010), porém a empresa gostaria que o seu processo também estivesse em conformidade com as práticas dos processos de gerência de requisitos dos modelos MR MPS.BR v2011 (SOFTEX 2011) e ASPICE PAM v2.5 (SPICE 2010). Novas práticas, não existentes em nenhum destes modelos, mas consideradas importantes para a empresa, podem ser também modeladas como um novo modelo, utilizando alguma arquitetura já existente ou mesmo criando uma nova arquitetura.

Neste caso, como a organização já está acostumada com a arquitetura do CMMI, é desejável que o perfil criado, com áreas de processo e processos de 3 modelos com arquiteturas diferentes, esteja na arquitetura do CMMI. Porém, quando for necessário interagir com especialistas nos outros modelos, é também desejável que o mesmo perfil seja apresentado nas outras arquiteturas. Com a metamodelagem das arquiteturas e dos modelos, estas transformações são automáticas. Para criar o perfil desejado o PRO2PI-MMCA já teria previamente cadastrado a área de processo REQM do CMMI-DEV, o processo GRE do MR MPS.BR e o processo ENG.1 do ASPICE PAM. Nos próximos passos do exemplo alguns elementos da arquitetura e alguns componentes dos modelos de capacidade de processo serão cadastrados. Os atributos **id** - *identification*, **name**, **def** - *definition*, **desc** - *description* e **version** do metamodelo de perfil de capacidade de processo serão utilizados no exemplo, os atributos não preenchidos serão omitidos.

- 1 Representar arquiteturas de modelos de capacidade de processo.
 - a. Cadastrar as informações de uma arquitetura de modelos.

Arquitetura CMMI, id : "CMMI" e name : "Capability Maturity Model Integration".
Arquitetura MPS.BR, id : "MPS.BR" e name : "Melhoria de Processo do Software Brasileiro".
Arquitetura SPICE, id : "SPICE" e name : "Software Process Improvement and (...)".

- b. Cadastrar os elementos das arquiteturas de modelos.

Elementos da arquitetura do CMMI, id : "PA", name : "Process Area"; id : "SG", name : "Specific Goal"; id : "SP", name : "Specific Practice".
Elementos da arquitetura do MPS.BR, id : "Pro", name : "Processo", id : "RE", name : "Resultado Esperado".
Elementos da arquitetura do SPICE, id : "Proc", name : "Process"; id : "PO", name : "Process Outcome"; id : "BP", name : "Base Practice".

- c. Cadastrar os relacionamentos entre os elementos das arquiteturas de modelos.

Relacionamentos da arquitetura do CMMI, uma área de processo é composta por metas específicas e uma meta específica é composta por práticas específicas.

Relacionamentos da arquitetura do MPS.BR, um processo é composto por resultados esperados.

Relacionamentos da arquitetura do SPICE, um processo é composto por resultados de processo e um resultado de processo é composto por práticas base.

- 2 Representar modelos de capacidade de processo.
 - a. Cadastrar as informações dos modelos.

Modelo: “CMMI-DEV” na arquitetura do CMMI, metaelemento: “ProcCapModel”, **id**: “CMMI-DEV”, **name**: “Capability Maturity Model Integration for Development”, **version**: “1.3”.

Modelo: “MR MPS.BR” na arquitetura MPS.BR, metaelemento: “ProcCapModel”, **id**: “MR MPS.BR”, **name**: “Modelo de Referência para Melhoria de Processo do Software Brasileiro”, **version**: “2011”.

Modelo: “ASPICE PAM” na arquitetura SPICE, metaelemento: “ProcCapModel”, **id**: “ASPICE PAM”, **name**: “Automotive Software Process Improvement and Capability dEtermination Process Assessment Model”, **version**: “2.5”.

- b. Cadastrar os componentes dos modelos.

Modelo: “CMMI-DEV”, área de processo: “REQM”, metaelemento: “ProcCapArea”, **id**: “REQM”, **name**: “Requirements Management”, **def**: “The purpose of...”, **desc**: “Requirements management processes manage all requirements received...”; meta específica: “SG 1”, metaelemento: “ProcCapArea”, **id**: “SG 1”, **name**: “Manage Requirements”, **def**: “Requirements are managed...”, **desc**: “The project maintains a current and approved...”; e prática específica: “SP 1.1”, metaelemento: “BasePractice”, **id**: “SP 1.1”, **name**: “Understand Requirements”, **def**: “Develop an understanding with the...”, **desc**: “As the project matures and requirements are...”.

Modelo: “MR MPS.BR”, processo: “GRE”, metaelemento: “ProcCapArea”, **id**: “GRE”, **name**: “Gerência de Requisitos”, **def**: “O propósito do processo Gerência de Requisitos...”; e resultado esperado: “GRE 1”, metaelemento: “BasePractice”, **id**: “GRE 1”, **name**: “O entendimento dos requisitos é obtido junto aos fornecedores de requisitos”.

Modelo: “ASPICE PAM”, processo: “ENG.1”, metaelemento: “ProcCapArea”, **id**: “ENG.1”, **name**: “Requirements elicitation”, **def**: “The purpose of the Requirements elicitation process is...”; resultado de processo: “1)”, metaelemento: “ProcOutcome”, **id**: “1)”, **name**: “continuing communication with the customer is established”; e prática base: “ENG.1.BP1”, metaelemento: “BasePractice”, **id**: “ENG.1.BP1”, **name**: “Obtain customer requirements and requests”.

- c. Cadastrar os relacionamentos entre os componentes de modelos.

Modelo: “CMMI-DEV”, a área de processo REQM é composta pela meta específica SG 1, que por sua vez é composta pela prática específica SP 1.1.

Modelo: “MR MPS.BR”, o processo GRE é composto pelo resultado esperado GRE 1.

Modelo: “ASPICE PAM”, o processo ENG.1 é composto pelo resultado de processo 1), que por sua vez é composto pela prática base ENG.1.BP1.

- d. Cadastrar um Perfil de Capacidade de Processo de um modelo.

Modelo: “CMMI-DEV”, Perfil de Capacidade de Processo: “ML2”, metaelemento: “ProcCapProf”, **id**: “ML2”, **name**: “Maturity Level 2”, **def**: “O perfil de capacidade de processo nível de maturidade 2 é composto pelas áreas de processos: CM - Configuration Management, MA - Measurement and Analysis, PMC - Project Monitoring and Control, PP - Project Planning, PPQA - Process and Product Quality Assurance, REQM - Requirements Management e SAM - Supplier Agreement Management, cada uma delas no nível de capacidade 2 e é apresentado na arquitetura do seu respectivo modelo”.

Modelo: “MR MPS.BR”, Perfil de Capacidade de Processo: “GRE-NC2”, metaelemento: “ProcCapProf”, **id**: “GRE-NC2”, **name**: “Gerência de Requisitos no Nível de Capacidade 2”, **def**: “O perfil de capacidade de processo Gerência de Requisitos no Nível de Capacidade 2 é composto pelo processo

GRE - Gerência de Requisitos com o nível de capacidade 2 e é apresentado na arquitetura do seu respectivo modelo”.

Modelo: “*ASPICE PAM*”, Perfil de Capacidade de Processo: “*ENG.1-CL2*”, metaelemento: “*ProcCapProf*”, **id**: “*ENG.1-CL2*”, **name**: “*Requirements elicitation no Nível de Capacidade 2*”, **def**: “*O perfil de capacidade de processo Requirements elicitation no Nível de Capacidade 2 é composto pelo processo ENG.1 - Requirements elicitation com o nível de capacidade 2 e é apresentado na arquitetura do seu respectivo modelo*”.

e. Cadastrar um Perfil de Capacidade de Processo de múltiplos modelos.

Modelos: “*CMMI-DEV, MR MPS.BR e ASPICE PAM*”, Perfil de Capacidade de Processo: “*REQM-GRE-ENG.1-CL2*”, metaelemento: “*ProcCapProf*”, **id**: “*REQM-GRE-ENG.1-CL2*”, **name**: “*Integrated Requirements Management no Nível de Capacidade 2*”, **def**: “*O perfil de capacidade de Requirements Management, Gerência de Requisitos e Requirements elicitation no Nível de Capacidade 2 é composto pela área de processo REQM - Requirements Management do CMMI-DEV, pelo processo GRE - Gerência de Requisitos do MR MPS.BR e pelo processo ENG.1 - Requirements elicitation do ASPICE PAM, todos com nível de capacidade 2 e são apresentadas nas arquiteturas dos seus respectivos modelos*”.

3 Estabelecer regras de transformação adicionais entre os elementos de arquitetura de modelos de capacidade de processo.

a. Cadastrar regras de transformação entre os elementos de arquitetura de modelos.

O elemento de arquitetura área de processo do CMMI é correlacionado com o elemento de arquitetura processo do MPS.BR. O elemento área de processo do CMMI também é correlacionado com o elemento processo do ASPICE. O elemento processo do MPS.BR é correlacionado com o elemento processo do ASPICE.

4 Realizar transformações de modelos e perfis de capacidade de processo.

a. Apresentar uma área de processo/processo de um modelo de origem na arquitetura de um modelo de destino.

Modelo: “*MR MPS.BR*”, Transformação de Modelo: “*Processo GRE na arquitetura do modelo CMMI-DEV*”, área de processo: “*GRE*”, metaelemento: “*ProcCapArea*” **{id**: “*GRE*”, **name**: “*Gerência de Requisitos*”, **def**: “*O propósito do processo Gerência de Requisitos...*”}, prática específica: “*GRE I*”, metaelemento: “*BasePractice*” **{id**: “*GRE I*”, **name**: “*O entendimento dos requisitos é obtido junto aos fornecedores de requisitos*”}.

b. Apresentar um perfil de capacidade de processo de um modelo de origem na arquitetura de um modelo de destino.

Modelos: “*CMMI-DEV, MR MPS.BR e ASPICE PAM*”, Transformação de Perfil de Capacidade de Processo: “*Requirements Management, Gerência de Requisitos e Requirements elicitation com Nível de Capacidade 2 na arquitetura MPS.BR*”, metaelemento: “*ProcCapProf*”.

Processo: “*REQM*”, metaelemento: “*ProcCapArea*” **{id**: “*REQM*”, **name**: “*Requirements Management*”, **def**: “*The purpose of...*”, **desc**: “*Requirements management processes manage all requirements received...*”} e resultado esperado: “*SP 1.1*”, metaelemento: “*BasePractice*” **{id**: “*SP 1.1*”, **name**: “*Understand Requirements*”, **def**: “*Develop an understanding with the...*”, **desc**: “*As the project matures and requirements are...*”}.

Processo: “*GRE*”, metaelemento: “*ProcCapArea*” **{id**: “*GRE*”, **name**: “*Gerência de Requisitos*”, **def**: “*O propósito do processo Gerência de Requisitos...*”} e resultado esperado: “*GRE I*”, metaelemento: “*BasePractice*” **{id**: “*GRE I*”, **name**: “*O entendimento dos requisitos é obtido junto aos fornecedores de requisitos*”}.

Processo: “*ENG.1*”, metaelemento: “*ProcCapArea*” **{id**: “*ENG.1*”, **name**: “*Requirements elicitation*”, **def**: “*The purpose of the Requirements elicitation process is...*”} e resultado esperado: “*ENG.1.BP1*”, metaelemento: “*BasePractice*” **{id**: “*ENG.1.BP1*”, **name**: “*Obtain customer requirements and requests*”}.

O Perfil de Capacidade de Processo desejado para a melhoria de processo de software na empresa de software foi gerado no item 4-b do exemplo. Com isto é possível apresentar um Perfil de Capacidade de Processo com elementos de múltiplos modelos em uma única arquitetura.

7. Conclusão

Atualmente o projeto de pesquisa encontra-se na fase de desenvolvimento do software PRO2PI-MMCA, o que possibilitará a geração de novos perfis de capacidade de processo, que poderão ser utilizados como referência em ciclos de melhoria de processo e em avaliações conjuntas. O uso de uma metodologia sistemática ajudou a conduzir o projeto e a atingir os objetivos estabelecidos inicialmente. A abordagem baseada em metamodelo tem se demonstrado vantajosa e dinâmica para múltiplos modelos, devido ao fato de respeitar as características intrínsecas de cada modelo de capacidade de processo durante a geração dos perfis de capacidade de processo. Como próximos passos estão previstas a conclusão da implementação e a utilização do software para representar os principais modelos, gerar perfis de múltiplos modelos e realizar transformações de arquitetura. Em seguida deverá ser realizada a terceira fase que com a evolução do metamodelo para enfatizar o conteúdo dos perfis e modelos.

O “problema” da utilização de múltiplos modelos foi “criado” pelo sucesso da MPS baseado em um modelo previamente definido. As abordagens atuais para múltiplos modelos buscam transformar um conjunto definido de modelos em um novo modelo previamente definido e desta forma tentam “resolver o problema no mesmo nível de abstração” com a atual MPS. O metamodelo apresentado neste artigo é a base para evoluir a MPS e tratar múltiplos modelos em outro nível de abstração, seguindo a orientação de Albert Einstein: “Os problemas significativos que enfrentamos não podem ser resolvidos no mesmo nível de pensamento em que estávamos quando os criamos”.

Agradecimentos

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) por apoiar financeiramente este trabalho de pesquisa com uma bolsa para um dos autores, ao Centro de Tecnologia da Informação Renato Archer (CTI) por oferecer a infraestrutura necessária para a realização do trabalho, e aos revisores anônimos do SBQS que contribuíram para a melhoria deste artigo.

Referências

- Armour, Philip G. 2004. *The Laws of Software Process - A New Model for the Production and Management of Software*. 1st ed. Washington: AUERBACH.
- Bézivin, Jean. 2003. “MDA™: From Hype to Hope, and Reality.” ATLAS Group (INRIA & LINA), University of Nantes, San Francisco.
- Bézivin, Jean. 2006. “Model Driven Engineering: An Emerging Technical Space.” *Generative and Transformational Techniques in Software Engineering* 4143:36-64.
- Brereton, Pearl, Barbara A. Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. 2006. “Lessons from applying the systematic literature review process within the software engineering domain.” *JSS - The Journal of Systems and Software*, July 10, pp. 571-583.
- Card, D. N. 2004. “Research Directions in Software Process Improvement.” *28th IEEE International Computer Software and Applications Conference*, September 27-30, pp. 238-239.

- Cruz, António M. R. d. S. R. d. 2010. "Automatic Generation of User Interfaces from Rigorous Domain and Use Case Models." Tese de Doutorado, Departamento de Engenharia Informática, Faculdade de Engenharia da Universidade do Porto, Porto.
- Favre, Jean-Marie. 2004. "Foundations of Model (Driven) (Reverse) Engineering: Models - Episode I: Stories of The Fidus Papyrus and of The Solarus." Université Joseph Fourier, Grenoble, France.
- Favre, Jean-Marie. 2005. "Foundations of Meta-Pyramids: Languages vs. Metamodels - Episode II: Story of Thotus the Baboon." University of Grenoble, Grenoble, France.
- ISO/IEC. 2005. "ISO/IEC 15504-5: Information Technology - Process Assessment - Part 5 - An exemplar Process Assessment Model." ISO, Geneve. International Organization for Standardization/International Electrotechnical Commission.
- Kitchenham, Barbara A. 2007. "Guidelines for performing Systematic Literature Reviews in Software Engineering." Keele University and University of Durham, Keele and Durham.
- Kleppe, Anneke G., Jos Warmer, and Wim Bast. 2003. *MDA explained: the model driven architecture - practice and promise*. 1st ed. Boston: Pearson Education.
- Mendes, Fabiana F. 2010. "Melhoria de Processos de Tecnologia da Informação Multi-Modelo." Universidade Federal de Goiás - Instituto de Informática, Goiânia.
- Mens, Tom, Krzysztof Czarnecki, and Pieter V. Gorp. 2005. "A Taxonomy of Model Transformations." *Dagstuhl Seminar Proceedings*, pp. 10.
- Petticrew, Mark and Helen Roberts. 2006. *Systematic Reviews in the Social Sciences - A PRACTICAL GUIDE*. Malden: Blackwell Publishing.
- Rossini, Alessandro, Adrian Rutle, Khalid A. Mughal, Yngve Lamo, and Uwe Wolter. 2011. "A Formal Approach to Data Validation Constraints in MDE." *In TTSS 2011: 5th International Workshop on Harnessing Theories for Tool Support in Software*, September, pp. 65-76.
- Salviano, Clenio F. 2009. "A Multi-Model Process Improvement Methodology Driven by Capability Profiles." *COMPSAC - The 33rd Annual IEEE International Computer Software and Applications Conference*, July 20-24, pp. 636-637. In proceedings of COMPSAC 2009.
- Salviano, Clenio F. 2011. "A Modeling View of Process Improvement." *SPICE - Software Process Improvement and Capability dEtermination*, May 31, pp. 1-12.
- Salviano, C. F. and A. M. Figueiredo. 2008. "Unified Basic Concepts for Process Capability Models." *SEKE*, July 1-3, pp. 173-178.
- Salviano, Clenio F., Márcia R. M. Martinez, Edgar L. Banhesse, Angela Enelize, Alessandra Zoucas, and Marcello Thiry. 2010. "A Method for Tridimensional Process Assessment using Modeling Theory." *IEEE - Computer Society - International Conference on the Quality of Information and Communications Technology*, pp. 430-435.
- SEI. 2010. "CMMI for Development, Version 1.3." Carnegie Mellon University.
- Silva, Edna L. d. and Etera M. Menezes. 2005. "Metodologia da Pesquisa e Elaboração de Dissertação." Universidade Federal de Santa Catarina - UFSC, Florianópolis.
- SOFTEX. 2011. "MPS.BR - Melhoria de Processo do Software Brasileiro - Guia Geral." SOFTEX, Campinas. Associação para Promoção da Excelência do Software Brasileiro.
- SPICE. 2010. "Automotive SPICE Process Assessment Model, Version 2.5." The Procurement Forum, SPICE, Brussels.
- Thiry, Marcello, A. Zoucas, and L. Tristão. 2010. "Mapping Process Capability Models to Support Integrated Software Process Assessments." *CLEI ELECT. JOURNAL* 13:Paper 4.
- Unterkalmsteiner, Michael, Tony Gorschek, A. K. M. M. Islam, Chow K. Cheng, Rahadian B. Permadi, and Robert Feldt. 2011. "Evaluation and Measurement of Software Process Improvement - A Systematic Literature Review." *IEEE Transactions On Software Engineering*, pp. 1-29.