

Spider-PM: Uma Ferramenta de Apoio à Modelagem de Processos de Software

Renan Sales Barros¹, Sandro Ronaldo Bezerra Oliveira¹

¹Faculdade de Computação – Instituto de Ciências Exatas e Naturais (ICEN)
Universidade Federal do Pará (UFPA), Rua Augusto Corrêa, 01, Belém-PA – Brasil

renansalesbarros@gmail.com, srbo@ufpa.br

Abstract. *The Spider-PM is a free software tool that aims to provide a mechanism for software processes modeling with a focus on practices and characteristics found in the software industry. In this paper, the main features and characteristics of Spider-PM are presented.*

Resumo. *A Spider-PM é uma ferramenta de software livre que objetiva prover um mecanismo para modelagem de processos de software com foco nas práticas e características encontradas na indústria de software. Neste artigo, as principais funcionalidades e características da Spider-PM são apresentadas.*

1. Introdução

A Engenharia de Software é uma área de conhecimento que compõe a Ciência da Computação e lida com os elementos envolvidos no processo de desenvolvimento de um software. Esse processo também pode ser definido como o mecanismo pelo qual os requisitos dos usuários são transformados em uma solução baseada em um software. Assim, as atividades, procedimentos e recursos envolvidos durante a construção dessa solução fazem parte do processo de Engenharia de Software ou processo de software [Oliveira, 2006].

Esses processos são definidos por organizações inseridas em ambientes de negócios que, com frequência, sofrem modificações. Essas modificações exigem que as organizações alterem seus processos de software [Softex, 2009]. Dessa forma, essas modificações obrigam que as organizações sejam capazes de melhorar continuamente os seus processos de software caso elas desejem produzir softwares com qualidade e de maneira competitiva.

Um processo de software pode ser representado de diferentes formas. Uma dessas formas é através de um modelo. Um modelo é uma representação diagramática dos principais aspectos de um processo de software onde são listados os elementos desse processo e as relações que existem entre esses elementos. Um dos objetivos desse tipo de representação é facilitar o aperfeiçoamento contínuo do processo de software [Kellner e Hansen, 1988].

Existem diversas ferramentas que apóiam a modelagem de processos de software com base em uma das linguagens de modelagem existentes. A maioria dessas linguagens e ferramentas é concebida a partir de estudos acadêmicos que desconsideram aspectos práticos envolvidos na modelagem de um processo de software. Em um levantamento e análise das linguagens de modelagem mais representativas encontradas na literatura especializada [Oliveira, 2006], observou-se que nenhuma dessas linguagens foi elaborada com foco em facilitar ou em tornar mais prática a modelagem de

processos de software justificado pela concepção das mesmas a partir de projetos acadêmicos não levando em consideração boas práticas obtidas com base na experiência em desenvolvimento de projetos pela indústria de software. Diante desse cenário, a Spider-PM foi concebida objetivando incorporar as principais práticas adotadas pelas organizações para modelagem de processos e simultaneamente permitir a construção de modelos de processo consistentes com quaisquer processos adotados pela indústria.

Mais detalhes sobre a Spider-PM serão apresentados na Seção 2 desse artigo. Nessa seção, serão mostradas as principais características e funcionalidades da Spider-PM além de uma descrição da sua arquitetura. Uma comparação envolvendo a Spider-PM e um conjunto de outras soluções para modelagem de processos será apresentada na Seção 3. Na seção 4, os resultados e trabalhos futuros envolvendo a Spider-PM serão relatados, e, na seção 5, serão apresentadas as considerações finais do trabalho realizado.

2. A Ferramenta

A Spider-PM foi idealizada a partir das pesquisas do projeto SPIDER da Universidade Federal do Pará [Oliveira, 2009]. Esse projeto visa o levantamento, customização e desenvolvimento de ferramentas livres para apoiar a implantação do MPS.BR – Melhoria de Processo de Software Brasileiro [Softex, 2009].

O MPS.BR é um programa que visa definir e aprimorar um modelo de melhoria e avaliação de processos de software mais adequado às características das organizações nacionais. Esse modelo apresenta sete níveis de maturidade para um processo de software. Os níveis de maturidade são estágios de melhoria e definem graus de evolução em um processo de software. Cada um desses níveis indica um grau de maturidade para o processo da organização. Em cada um desses níveis do MPS.BR podem existir conjuntos de atividades relacionadas entre si. No contexto do MPS.BR esses conjuntos de atividades são chamados de processos. Cada um desses processos apresenta resultados esperados. Um resultado esperado consiste de um item que permite observar se uma meta do processo foi alcançada ou não.

A Spider-PM lida com os conhecimentos envolvidos nos seguintes processos do MPS.BR: Definição do Processo Organizacional (DFP) e Avaliação e Melhoria do Processo Organizacional (AMP). Com relação ao processo DFP, a Spider-PM é parcialmente aderente aos resultados esperados DFP1, DFP2, DFP3 e DFP4. Do processo (AMP), a Spider-PM é parcialmente aderente aos resultados esperados AMP1 e AMP2. Apesar dessa relação com o MPS.BR, a Spider-PM é uma ferramenta para modelagem de processos de software de um modo geral, ou seja, tanto processos de software baseados no MPS.BR quanto não baseados podem ser modelados na Spider-PM.

Um dos principais objetivos da modelagem de processos é facilitar o entendimento, comunicação, aperfeiçoamento e gerência do processo de software [Kellner e Hansen, 1988]. Além desse objetivo, a Spider-PM também visa: incorporar as práticas de modelagem adotadas pelas organizações; permitir a produção de modelos compatíveis com quaisquer processos de software do mundo real; e facilitar e agilizar a construção desses modelos.

Para atingir esses objetivos, optou-se por utilizar a linguagem de modelagem SPEM – *Software Process Engineering Meta-Model*. A SPEM é a linguagem padrão da

OMG – *Object Management Group*, para modelagem de processos e foi definida em 2002. A versão 2.0 foi utilizada sendo que essa versão foi lançada em abril de 2008. A OMG é uma organização internacional que define os padrões livres para as aplicações orientadas a objetos. A SPEM define um conjunto de estereótipos específicos para a modelagem de processos que são utilizados com as demais notações e regras determinadas pela UML – *Unified Modeling Language*, ou seja, a SPEM define um perfil da UML [OMG, 2008].

A SPEM foi escolhida principalmente em virtude de ela ser a linguagem padrão da OMG para modelagem de processos de software e por reutilizar as notações UML, as quais são familiares para os profissionais que lidam com o processo de software. No entanto, a partir de observações realizadas durante a implementação de programas de melhoria de processos organizacionais, constatou-se que apenas um subconjunto dos estereótipos e diagramas disponibilizados pela SPEM é utilizado pela maioria das organizações. Dessa forma, optou-se pela definição de um perfil da SPEM denominado SPIDER_ML. Esse perfil da SPEM foi adotado como linguagem de modelagem da Spider-PM. A SPIDER_ML [Barros, 2009] visa: incorporar e formalizar práticas de modelagem de processos adotadas pela indústria de software e para limitar a quantidade de estereótipos e diagramas utilizados na modelagem de processos de modo a torná-la mais simples. A SPIDER_ML é uma linguagem de modelagem de processo de software usando paradigma Híbrido, onde se encontra características de Linguagens Orientadas a Redes, uma vez que ela objetiva expressar concorrência e paralelismo das operações de uma maneira mais natural; e de Linguagens Baseadas em Regras, onde cada processo é representado por um conjunto de objetivos, condições, restrições e efeitos.

A SPIDER_ML foi elaborada para permitir que os objetivos da Spider-PM pudessem ser atingidos. A utilização da Spider-PM e da SPIDER_ML tem como principais vantagens: a simplificação da modelagem de processos, dado o refinamento do conjunto de notações, estereótipos e diagramas que são utilizados pela SPEM; e a modelagem de processos de software aderente às recomendações de programas de melhoria da qualidade do processo de software, como o MPS.BR. Além disso, apesar de não apresentar todos os elementos utilizados pela SPEM, a produção de modelos totalmente compatíveis com a SPEM é possível a partir dos elementos que são disponibilizados pela Spider-PM. Portanto, a Spider-PM também pode ser utilizada como uma ferramenta para modelagem de processos de acordo com as especificações da SPEM.

2.1. Principais Funcionalidades

Assim como a SPEM, a SPIDER_ML e a Spider-PM apresentam uma distinção clara entre os elementos que podem ser reutilizados em diferentes processos de software e os elementos que são únicos em um processo de software. Assim sendo, a definição de um processo de software na Spider-PM ocorre em duas etapas: a **definição do processo padrão**; e sua **instanciação para projetos específicos**. De acordo com essa abordagem, em um primeiro momento é definido o conjunto de elementos básicos e comuns a todos os processos da organização e que são independentes das características do software a ser desenvolvido [Oliveira, 2006] denominado **processo padrão** [ISO, 2008]. Em seguida, **instanciações do processo padrão** para projetos específicos são realizadas considerando aspectos como: o tipo do software; os paradigmas de desenvolvimento;

ciclo de vida do processo; características do projeto e da equipe responsável pelo projeto; e os recursos disponíveis para esse projeto.

Dessa forma, uma das funcionalidades da Spider-PM consiste na modelagem do processo padrão. Na Figura 1 tem-se a interface que foi desenvolvida para essa funcionalidade. A Figura 1A mostra a definição de uma tarefa do processo padrão e a Figura 1B mostra a decomposição do processo padrão em áreas de conhecimento ou disciplinas.

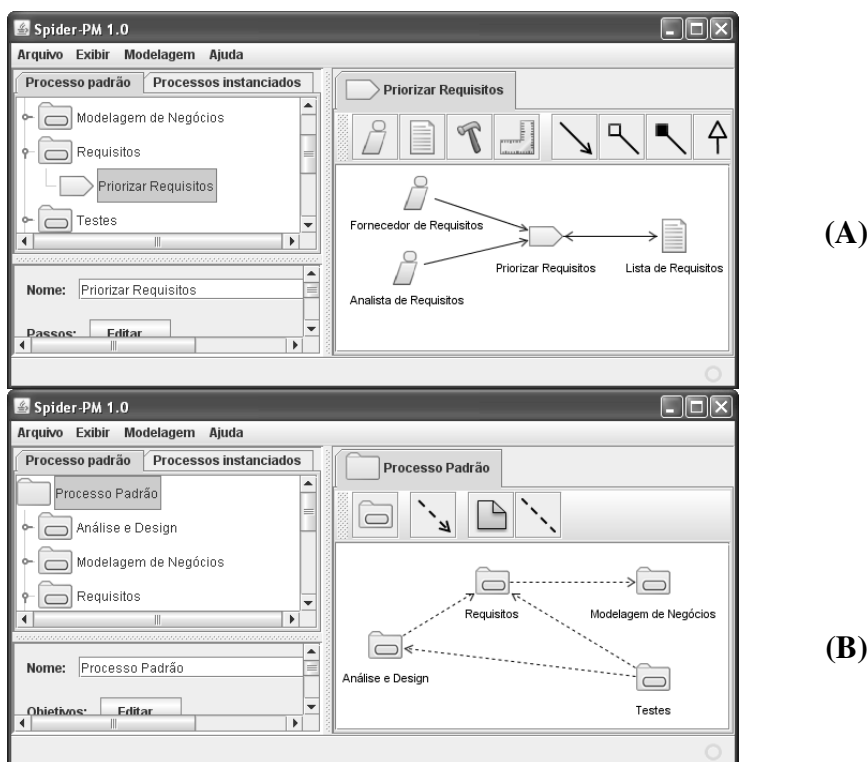


Figura 1. Modelagem do Processo Padrão na Spider-PM.

Além de tarefas e disciplinas, a modelagem do processo padrão também pode apresentar: procedimentos para a realização das tarefas; papéis desempenhados durante a execução das tarefas; produtos de trabalhos consumidos, gerados ou modificados pelos papéis durante a realização de uma tarefa; ferramentas que apoiem o cumprimento das tarefas; e os relacionamentos entre esses elementos.

Assim como ocorre para o processo padrão, outra funcionalidade disponibilizada pela Spider-PM é a modelagem das instanciações do processo padrão. Essa funcionalidade tem como objetivo possibilitar o refinamento do processo padrão. A Spider-PM permite que diversas instanciações possam ser feitas a partir de um único processo padrão. É possível, ainda, que os elementos utilizados nessas instanciações sejam criados automaticamente pela ferramenta a partir dos elementos do processo padrão ou de instanciações do processo padrão.

Além da descrição dos elementos do processo de software através de diagramas UML, a Spider-PM possibilita detalhar cada um dos elementos da modelagem. Esse detalhamento é feito a partir do preenchimento de campos específicos que são apresentados na interface gráfica de acordo com o tipo do elemento do processo que está selecionado na ferramenta, por exemplo, se o elemento selecionado é uma

ferramenta, é possível associá-la a um arquivo executável ou URL, caso seja uma ferramenta *web*, que pode ser iniciada a partir da Spider-PM. No decorrer desse detalhamento e ao longo da elaboração dos diagramas da modelagem, as informações fornecidas pelos usuários são verificadas com o intuito de garantir que não sejam produzidos modelos inconsistentes. Esse processo é feito automaticamente e, quando possível, as inconsistências são corrigidas pela ferramenta sem a intervenção do usuário. Todas essas verificações são realizadas com base nas regras e restrições definidas pela sintaxe da linguagem SPIDER_ML.

A Spider-PM também apresenta uma funcionalidade que permite estender a semântica dos elementos do processo a partir da definição de estereótipos e regras para utilização desses estereótipos. Basicamente, pode-se definir uma lista de estereótipos e associá-la a um determinado tipo de elemento da modelagem. Quando um desses estereótipos for atribuído a um elemento do tipo determinado, os demais estereótipos dessa lista não mais poderão ser atribuídos a esse componente. Por exemplo, uma lista com os estereótipos “fundamental”, “de apoio” e “organizacional” pode ser definida e associada aos componentes que representam tarefas do processo padrão. Nesse caso, uma tarefa do processo padrão poderá ser classificada de acordo com os estereótipos dessa lista e não poderá receber dois ou mais desses estereótipos concomitantemente, ou seja, uma tarefa não poderá ser “fundamental” e “de apoio” ao mesmo tempo. Assim, além de verificar as regras e restrições da linguagem SPIDER_ML, a Spider-PM também verifica se os elementos respeitam as definições de estereótipos criadas pelos usuários.

2.2. Arquitetura

A arquitetura da Spider-PM foi baseada no padrão arquitetural *Model-View-Controller* [Sun, 2002]. A Figura 2 ilustra a arquitetura da Spider-PM. Nessa figura as setas indicam fluxo dos dados trocado pelos módulos e sub-módulos da ferramenta. Com exceção do sub-módulo de leitura de arquivo, todos os demais módulos consultam as informações contidas no modelo, no entanto apenas o módulo de controle pode modificar essas informações. As setas indicando esses fluxos de informações a partir do modelo não foram representadas na Figura 2.

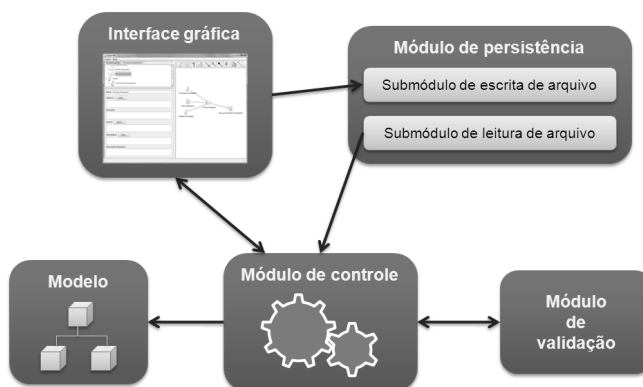


Figura 2. Arquitetura da Spider-PM.

Com base nessa arquitetura, inicialmente, os dados que constituirão o modelo podem ter como origem os arquivos lidos pela ferramenta, ou os eventos provenientes da interface gráfica como consequência das intervenções do usuário na ferramenta.

Essas informações são passadas para o módulo de controle que por sua vez as repassa para o módulo de validação.

O módulo de validação, por sua vez, verifica se existem inconsistências entre essas informações e os dados já existentes no modelo. Caso nenhuma inconsistência seja detectada, o módulo de validação reporta esse fato para o módulo de controle que incorpora as novas informações ao modelo e indica que a interface gráfica deve ser atualizada para exibir o modelo com as modificações realizadas. No caso de uma inconsistência ser detectada pelo módulo de validação, o mesmo tentará corrigi-la. Se a correção for possível, os dados corrigidos são enviados para o módulo de controle que seguirá suas operações, analogamente, como descrito para o caso de não ter sido encontrada nenhuma inconsistência. Quando uma inconsistência não pode ser corrigida, o módulo de validação encaminha esta informação para o módulo de controle. Posteriormente, o módulo de controle descarta essas informações inconsistentes e reporta esse fato para a interface gráfica que alertará o usuário sobre o ocorrido.

Com a modelagem definida, a partir da interface gráfica da ferramenta, é possível indicar para o módulo de persistência que um arquivo com as informações contidas no modelo deve ser gerado.

Todos os módulos da Spider-PM foram implementados em Java. Além disso, outra tecnologia utilizada pela ferramenta é a persistência das modelagens de processo por meio de arquivos XML – *Extensible Markup Language*. A Spider-PM possui licença GPL – *General Public License* e está disponível em <http://www.spider.ufpa.br>, acessando o menu Resultados.

3. Trabalhos Relacionados

Como mencionado, existem diversas linguagens de modelagem e ferramentas de apoio à modelagem de processos de software, no entanto, são poucas as ferramentas construídas objetivando a utilização das notações da SPEM apesar dessa ser a linguagem padrão da OMG para modelagem de processos de software. De um modo geral, as ferramentas para modelagem de processos de software apresentam linguagens próprias e as ferramentas que permitem a modelagem de processos usando a SPEM são ferramentas desenvolvidas para construção de diagramas UML, ou outros tipos de diagramas, que apresentam um módulo adicional para modelagem usando SPEM. A Tabela 1 mostra uma comparação entre as seguintes ferramentas que possibilitam a modelagem de processos usando SPEM: Spider-PM 1.0 (S-PM); Enterprise Architect 8.0 (EA); Objecteering Free Edition 6.1 (OFE); Objecteering Enterprise Edition 6.1 (OEE); StarUML 5.0 (S-UML); Eclipse Process Framework Composer 1.5 (EPF); e APES 2.5 (APES).

Com exceção da Spider-PM, APES e EPF, as demais ferramentas são modeladores UML que apresentam um módulo adicional para modelagem usando as notações gráficas da SPEM. Durante a análise da ferramenta EPF, a interface e documentação da ferramenta indicavam que a mesma realiza uma validação dos elementos utilizados no processo modelado, no entanto, ao acessar essa funcionalidade, não fica claro quais os dados que foram validados. Diferentemente da EPF e das outras ferramentas que realizam validação, a Spider-PM não permite que essa funcionalidade seja acessada diretamente, durante a modelagem do processo de software, a Spider-PM reporta qualquer construção inválida efetuada.

Além das ferramentas anteriormente citadas, duas outras soluções para modelagem de processos usando SPEM foram encontradas durante as pesquisas realizadas. Uma das soluções consiste em um arquivo extensão para o programa Microsoft Visio que permite a utilização das notações gráficas da SPEM. Essa solução não foi avaliada uma vez que a ferramenta Microsoft Visio não é uma distribuída gratuitamente e não apresenta uma versão gratuita para avaliação. A outra solução é uma aplicação *web* denominada Spemmet desenvolvida para modelagem por meio da SPEM. A Spemmet não foi avaliada uma vez que não se encontra acessível para uso.

Tabela 1. Comparação entre Ferramentas que suportam a Modelagem de Processos usando SPEM.

	S-PM	EA	OFE	OEE	S-UML	EPFC	APES
Plataforma:	Java 1.6	Windows Linux Macintosh	Windows Linux	Windows Linux Solaris	Windows	Windows Linux	Java 1.6
É gratuita?	Sim	Não	Sim	Não	Sim	Sim	Sim
Foca a modelagem de processos usando SPEM?	Sim	Não	Não	Não	Não	Sim	Sim
Utiliza as notações da SPEM 2.0?	Sim	Sim	Não	Não	Sim	Sim	Não
Apresenta campos específicos para detalhamento dos elementos acordo com o tipo do elemento?	Sim	Sim	Sim	Sim	Não	Sim	Não
Separa os elementos reutilizáveis do processo?	Sim	Não	Não	Não	Não	Sim	Não
Valida o processo modelado?	Sim	Não	Sim	Sim	Não	Não Avaliado	Sim
Publica os processos modelados em um formato para visualização não modificável?	Não	Não	Não	Sim	Não	Sim	Não

A versão atual (1.0) da Spider-PM não publica os seus modelos de processos. No entanto, pretende-se que futuramente a Spider-PM apresente uma funcionalidade que permita a exportação dos modelos de processos de software para um formato navegável, visualizável, não modificável e acessível a partir de um navegador *web* comum. Essa funcionalidade tem como objetivo permitir que as organizações possam comunicar seus modelos de processos mais facilmente.

4. Resultados e Trabalhos Futuros

Atualmente, a versão 1.0 da Spider-PM foi lançada, estando o seu uso em validação pela equipe do projeto SPIDER a partir do controle dos processos de desenvolvimento dos projetos de pesquisa, relacionados à construção de software. Com base nas pesquisas realizadas, observa-se que a Spider-PM sistematiza a construção de modelos de processo de software de maneira mais simplificada se comparada com as soluções analisadas em [Oliveira, 2006] uma vez que ela apresenta um pequeno conjunto de elementos e diagramas. Essa simplificação, aliada com as validações que ocorrem durante a construção dos modelos de processos e a reutilização de notações UML nesses modelos, contribui para a rápida assimilação da semântica dos elementos da modelagem e entendimento dos modelos construídos.

Além das facilidades já obtidas com relação à modelagem de processos, espera-se que, futuramente, a Spider-PM seja integrada com um ambiente para simulação dos processos por ela modelados e também a integração com um ambiente automatizado para avaliação e melhoria do processo de software.

Vale ressaltar que as práticas de modelagem de processos citadas nesse trabalho encontram-se em institucionalização no Projeto SPIDER [Oliveira, 2009], a fim de

validar a efetividade do fluxo de atividades da ferramenta Spider-PM a partir da elaboração de estatísticas sobre os resultados obtidos com o seu uso. Desta forma, é possível um relato das experiências e lições aprendidas com a implantação deste ferramental e um Manual de Implementação do processo de Definição do Processo Organizacional (DFP) do MPS.BR adotando a Spider-PM.

5. Considerações Finais

A modelagem de processos traz para uma organização benefícios. No entanto, as principais soluções para modelagem de processos possuem foco acadêmico, são de difícil utilização ou são proprietárias. Tais características podem implicar em dificuldades para se adotar essas soluções. Para reduzir essas dificuldades, a Spider-PM oferece um mecanismo para modelagem de processos com foco na indústria de software e nas práticas que a mesma utiliza sendo ao mesmo tempo de fácil utilização e coerente com os processos de software que de fato são utilizados pela indústria.

Assim, a Spider-PM facilita o atendimento dos objetivos da modelagem de processos para uma organização contribuindo positivamente para o aperfeiçoamento, comunicação, gerenciamento e entendimento dos processos de software das organizações que a utilizarem. Dessa forma, a Spider-PM deve promover a melhoria do processo de software e conseqüentemente contribuir para o aumento da qualidade dos softwares uma vez que a melhoria do processo de software tem sido um fator determinante para a obtenção da qualidade do produto de software final [Oliveira, 2006]

Referências

- Barros, R. S., Oliveira, S. R. B. (2009) “Especificação Técnica da Linguagem SPIDER_ML, versão 1.0”. Universidade Federal do Pará. Disponível em www.spider.ufpa.br.
- the International Organization for Standardization and the International Electrotechnical Commission (2008) “ISO/IEC 15504-2 Information Technology – Process Assessment – Part 2 – Performing and Assessment”. Geneve.
- Kellner, M. I., Hansen, G. A. (1988) “Software Process Modeling”. Software Engine Institute – Carnegie Mellon University. Pittsburgh.
- Oliveira, S. R. B. (2006) “Processo de Software: Princípios, Ambiente e Mecanismos de Execução”. Exame de Qualificação do Programa de Doutorado. Centro de Informática – Universidade Federal de Pernambuco. Recife. Disponível em www.ufpa.br/srbo
- Oliveira, S. R. B. (2009) “SPIDER: Uma Proposta de uma Solução Sistêmica de Apoio à Implementação do programa MPS.BR”. Projeto de Pesquisa. Faculdade de Computação – Instituto de Ciências Exatas e Naturais – Universidade Federal do Pará. Belém. Disponível em www.spider.ufpa.br
- OMG – Object Management Group (2008) “Software & Systems Process Engineering Meta-Model Specification”.
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro (2009) “MPS.BR – Melhoria de Processo do Software Brasileiro – Guia Geral”.
- Sun Microsystems (2002) “Java BluePrints – Model-View-Controller”.