

RAPDIS: Um Processo MDA para Desenvolvimento de Sistemas de Informação

Gisele P. Morgado¹, Denis S. Silveira², Antônio J. Alencar¹, Priscila M. V. Lima¹, Eber A. Schmitz¹

¹Universidade Federal do Rio de Janeiro, IM/NCE, Rio de Janeiro, RJ, Brasil.

²Faculdades Ibmecc/RJ, Av. Rio Branco 108, 9ª - Centro, Rio de Janeiro, RJ, Brasil.
Universidade Federal do Rio de Janeiro, COPPE/Produção, Rio de Janeiro, RJ, Brasil.

geti@dcc.ufrj.br

***Resumo.** Este artigo apresenta o processo RAPDIS, que auxilia o desenvolvimento de Sistemas de Informação, oferecendo um suporte completo ao MDA. Através do RAPDIS é possível construir o modelo independente de computação e aplicar uma série de transformações sobre este modelo para que se possa chegar até uma plataforma específica. A automatização destas transformações agiliza e facilita as tarefas dos profissionais de Tecnologia da Informação, possibilitando o desenvolvimento de sistemas que atendam de forma mais eficiente aos negócios a que se destinam e que sejam de melhor manutenção, portabilidade e interoperabilidade.*

1 Introdução

Todas as organizações fazem uso de algum tipo de Sistema de Informação (SI). Podemos dizer que a Tecnologia da Informação (TI) é parte integrante de uma organização e que os seus negócios definem os requisitos de um SI. Mas, apesar de todos os recentes avanços tecnológicos, modelar e implementar SIs ainda são atividades árduas e nem sempre bem sucedidas.

Assim, frequentemente as organizações se encontram insatisfeitas com a qualidade dos seus sistemas. Neste sentido, os problemas mais comuns são: a incapacidade destes sistemas de apoiar de forma eficiente e efetiva as operações do negócio, a dificuldade de manutenção dos sistemas e a deficiência na integração com os outros sistemas da organização.

No intuito de buscar uma solução para esse problema, este trabalho apresenta um novo processo de desenvolvimento de SI, denominado de processo RAPDIS (*Rules And Process for the Development of Information Systems*) e um ambiente integrado que fornece o apoio computacional às atividades deste processo. O restante deste texto encontra-se organizado da seguinte forma: na segunda seção apresentamos os conceitos que servem de base para este trabalho; na terceira seção descrevemos o processo RAPDIS; na quinta seção apresentamos o ambiente que apóia este processo; e na quarta seção relatamos as considerações finais.

2 Contextualização

Antes de descrever detalhes mais específicos do trabalho, alguns conceitos envolvidos necessitam ser discutidos. As seções 2.1 e 2.2 apresentam resumidamente estes conceitos.

2.1 Processos de Engenharia de Sistemas

Processos de Engenharia de Sistemas especificam a seqüência de passos a serem seguidos durante o desenvolvimento de um SI. A cada um destes passos, associa-se um conjunto de atividades, seus produtos e as regras de verificação que garantem a passagem para a próxima fase (PRESSMAN, 2001).

Os produtos das diferentes fases de um processo são especificações que descrevem, com um certo grau de abstração, o sistema de informação que está sendo desenvolvido. Esta especificação abstrata é chamada de modelo e ela nos permite uma melhor compreensão do SI que está sendo construído (SCHMITZ; SILVEIRA, 2000).

2.2 MDA - *Model Driven Architecture*

O MDA constitui um *framework* para o desenvolvimento de sistemas baseado em fundamentos já consagrados na disciplina de Engenharia de Software que visam à separação entre a descrição da funcionalidade de um sistema e a especificação da mesma funcionalidade em uma plataforma tecnológica específica (MUKERJI; MILLER, 2003). No MDA o desenvolvimento é dividido em três modelos diferentes: o modelo independente de computação CIM (*Computation Independent Model*), o modelo independente de plataforma PIM (*Platform Independent Model*) e o modelo específico de plataforma PSM (*Platform Specific Model*).

O CIM é o modelo de mais alto nível do sistema, representando as possibilidades e funcionalidades deste que não dependem de computação. Este modelo é importante para o entendimento do problema e serve como fonte do vocabulário a ser usado nos demais modelos. O PIM descreve o sistema, porém não apresenta os detalhes da tecnologia que será usada na implementação. Este modelo oferece especificações formais da estrutura e funcionalidade do sistema, abstraindo-se de detalhes técnicos. O PSM combina a especificação do modelo PIM com detalhes de uma determinada plataforma. Este modelo é o de mais baixo nível de abstração e os seus elementos estão prontos para a geração de código (BLANC *et al*, 2004).

Entre cada um destes modelos há uma série de transformações que são aplicadas para que se possa chegar até uma plataforma específica. A transformação de modelos é o ponto forte do MDA e consiste no processo de converter um modelo em outro modelo do mesmo sistema (MUKERJI; MILLER, 2003). A idéia principal é construir modelos no seu mais alto nível de abstração e transformá-los em modelos com baixa abstração, de forma automática ou semi-automática, facilitando e agilizando o processo de desenvolvimento de SI (figura 1).



Figura 1 Modelos e Transformações do MDA.

3 O Processo RAPDIS

O RAPDIS é um processo para desenvolvimento de sistemas que foi concebido a partir dos conceitos do MDA. Nele aplicamos as transformações para a integração dos modelos utilizados no desenvolvimento de um sistema. A figura 2 ilustra o processo, que se inicia com o término da negociação do projeto. Resumidamente, este processo consiste na execução sequencial das seguintes atividades: “construir modelo de negócio”, “construir modelo de sistema de informação” e “implementar sistema”. Com o fim do processo RAPDIS, tem início a fase de homologação do sistema.

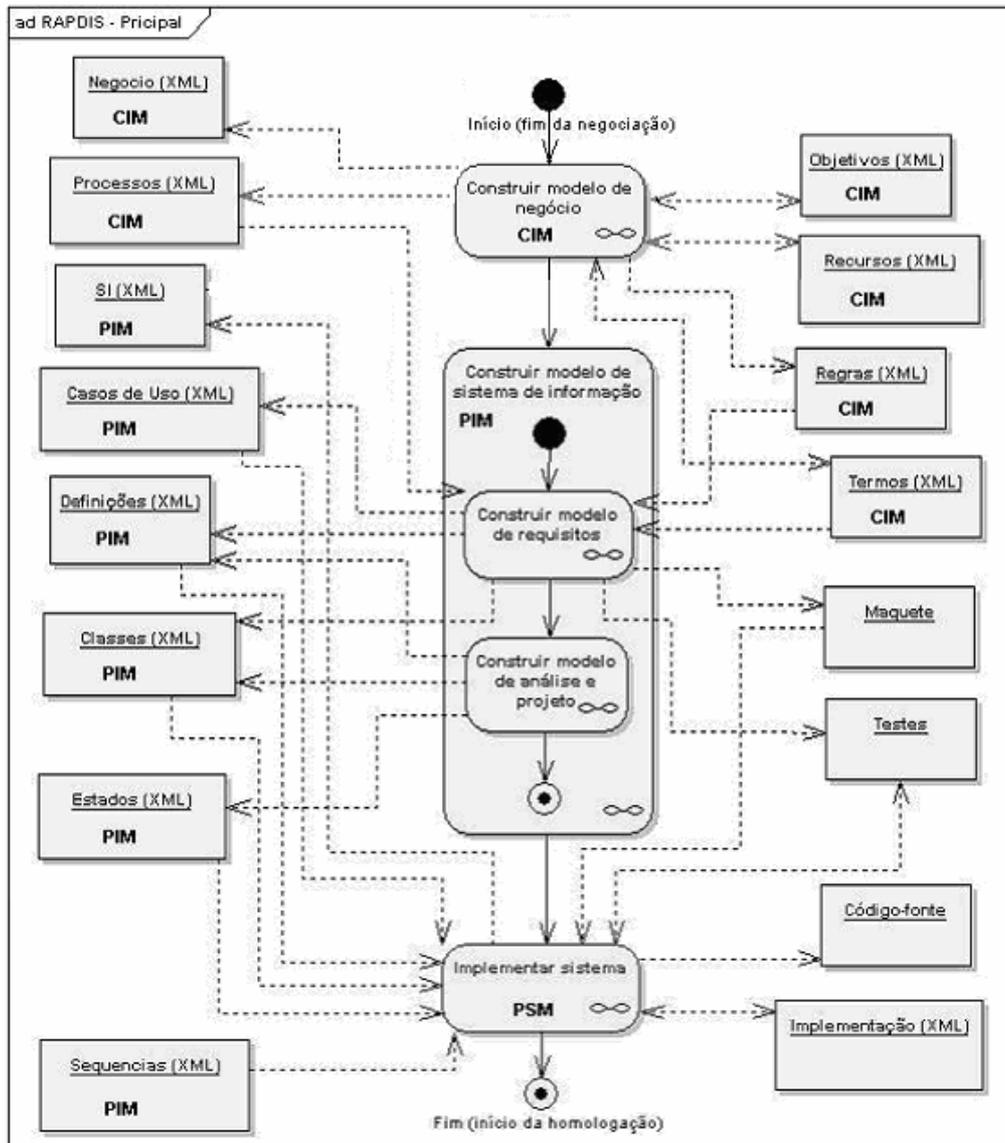


Figura 2 Diagrama Principal do Processo RAPDIS.

Na figura utilizamos as etiquetas “CIM”, “PIM” e “PSM” para marcar onde os modelos do MDA estão presentes no processo RAPDIS. Além disso, podemos observar que, durante todo o processo, os documentos/modelos produzidos em uma determinada atividade são utilizados como insumos da atividade seguinte. A atividade seguinte, por sua vez, produz com estes insumos novos documentos/modelos. Este fato caracteriza a existência das transformações do MDA dentro das atividades do processo RAPDIS.

3.1 Atividade “Construir Modelo de Negócio”

O primeiro modelo a ser especificado dentro do processo RAPDIS é o modelo de negócio através da atividade “construir modelo de negócio”. O modelo de negócio representa o modelo CIM do MDA. Esta atividade começa com uma breve descrição textual do negócio e segue com a definição dos objetos que compõem o modelo de negócio: objetivos, termos, recursos, regras e processos (ERIKSSON, PENKER; 2000).

Ao final da etapa de definição, deve ser realizada uma revisão e as correções necessárias devem ser efetuadas. Os modelos produzidos nesta etapa são fundamentais para o entendimento do sistema que se pretende desenvolver e formam os insumos da atividade “construir modelo de sistema de informação”.

3.2 Atividade “Construir Modelo Sistema de Informação”

A atividade “construir modelo de sistema de informação”, corresponde à construção do PIM do MDA. É nesta atividade que os componentes do sistema serão especificados. Ela é composta por duas sub-atividades que devem ser executadas em seqüência: “construir modelo de requisitos” e “construir modelo de análise e projeto”.

3.2.1 Sub-atividade “Construir Modelo de Requisitos”

Esta sub-atividade tem como objetivo definir as funcionalidades do sistema através dos seus casos de uso, classes de domínio e maquete. O primeiro passo na construção deste modelo é obter uma versão preliminar das classes de domínio e dos casos de uso a partir das transformações do modelo de negócio. Após obter esta versão preliminar, deve-se executar em paralelo os seguintes passos: refinar/completar os casos de uso, refinar as classes de domínio (modelo de domínio) e construir a maquete (projeto de interface).

A transformação do modelo preliminar das classes do domínio é possível através do mapeamento dos termos do negócio definidos no CIM. (DIAS *et. al.*, 2006; MARTINS, 2006). Como critério de seleção dos termos a serem transformados para classes de domínio usamos o de relacionamento com atributos. Segundo (MELLOR; SHLAER, 1990), um objeto deve possuir atributos para ser considerado um objeto. Desta forma, o mapeamento dos termos para uma classe de modelo conceitual considera apenas os termos que possuem atributos. Já os relacionamentos de herança, parte e associação são representados, respectivamente, por relacionamentos de generalização, composição e associação da UML. A figura 3 a seguir apresenta um exemplo dessa transformação.

Já a transformação dos processos de negócio em casos de uso consiste na aplicação de quatro conjuntos de heurísticas a um determinado modelo de processo (diagrama de atividades), para a obtenção do respectivo modelo de requisitos funcionais (diagrama de casos de uso) (DIAS *et. al.*, 2006; CRUZ, 2004). Os diagramas utilizados são modelos gráficos que possuem a vantagem da facilidade de comunicação, porém não possuem um rigor matemático. Assim, a identificação das heurísticas foi realizada através de experimentos e em conformidade com a literatura especializada de modelagem de processos e requisitos (MARSHALL, 2000; SANTANDER, CASTRO; 2000; SHNIEDER, WINTERS, 2001). A partir dos experimentos, foi possível observar a existência de algumas situações padronizadas de atividades que implicavam em

padrões correspondentes de casos de usos. Estes padrões foram analisados e deles foram extraídas as heurísticas.

Em um processo de negócio, uma atividade é um conjunto de ações detalhadas para o cumprimento do objetivo do processo. Por outro lado, um caso de uso, é um conjunto de passos a serem executados com a finalidade de atingir um objetivo computacional (do sistema). Assim, existe uma correlação entre estes elementos (atividade e caso de uso), sendo esta a base para as heurísticas.

Vale notar que nesta parte do processo RAPDIS ocorre a primeira transformação do MDA, a transformação do CIM para o PIM. Esta transformação consiste em dois passos: a transformação da definição dos termos do negócio (CIM) em um diagrama de classes de domínio (PIM) e a transformação dos processos de negócio (CIM) em um diagrama de casos de uso (PIM). A transformação facilita o trabalho de construção do modelo de requisitos, porém os modelos gerados precisam ainda ser refinados pelo analista de sistemas.

3.2.2 Sub-atividade “Construir Modelo de Análise e Projeto”

O objetivo desta sub-atividade é produzir um plano que permita a construção do sistema. Para isso, são definidos as demais classes do sistema, os diagramas de estado e os diagramas de seqüência.

Primeiramente, para cada pacote de casos de uso definido no Modelo de Requisitos, deve ser definido um diagrama de classes. Esse diagrama de classes deve mostrar os relacionamentos entre as classes de domínio, interface e controle. Depois disto, para cada pacote de casos de uso, deve ser definido um diagrama de seqüência. A criação dos diagramas de seqüência é um passo fundamental para a definição da implementação dos casos de uso do sistema. Ao definir os serviços dos objetos participantes do caso, o projetista vai refinando as definições das classes do sistema, previamente, definidos no Modelo de Domínio. Antes de prosseguir, os devem passar por uma revisão e devem ser efetuadas as correções necessárias.

O passo seguinte dentro desta sub-atividade é criar um diagrama de transição de estados para as classes de domínio, interface e controle que tenham ciclos de vida não-triviais. A determinação dos diagramas de estado pode levar a descoberta de novas relações de herança entre as classes. Ao terminar este passo, o Modelo de Análise e Projeto deve ser novamente revisado para tirar os eventuais erros de consistência.

3.3 Atividade “Implementar Sistema”

Nesta atividade, o Modelo de Sistema de Informação construído deve ser implementado. Para tal, iniciamos o processo a partir da transformação dos modelos produzidos na atividade anterior em código-fonte do programa, representado através de uma linguagem de programação padrão (Pascal, C, C++, Java, etc.).

Esta atividade de implementação do sistema é responsável por outras duas transformações do MDA: a transformação do PIM para o PSM e do PSM para o código. No processo RAPDIS este conjunto de transformações é representado pela geração de código a partir do Modelo de Sistema de Informação e facilita o trabalho dos desenvolvedores por eliminar alguns passos repetitivos da fase de codificação.

4 O Ambiente RAPDIS

O ambiente RAPDIS é um software gratuito para fins acadêmicos desenvolvido em *Object Pascal* para plataforma *Windows*. Ele se encontra disponível para *download* no seguinte endereço eletrônico: www.geti.dcc.ufrj.br/projetos.php.

Para facilitar a apresentação das suas funcionalidades, dividimos este ambiente em duas partes: o Modelo de Negócio e o Modelo de Sistema de Informação. Cada uma destas partes possui um editor e ferramentas para realizar transformação entre os modelos. Todos os modelos produzidos no RAPDIS são armazenados através de documentos XML (*eXtensible Markup Language*) (YERGEAU *et al*, 2004).

Na parte relativa ao Modelo de Negócio é possível especificar os termos, objetivos, recursos, processos e regras do negócio. Além disso, esta parte do RAPDIS dispõe de dois geradores: o gerador de classes de domínio e o gerador de casos de uso. O gerador de classes de domínio realiza a transformação da definição dos termos do negócio em um diagrama de classes e o gerador de casos de uso transforma um processo de negócio em um diagrama de casos de uso. As figuras 3 e 4 ilustram a utilização destes dois geradores.

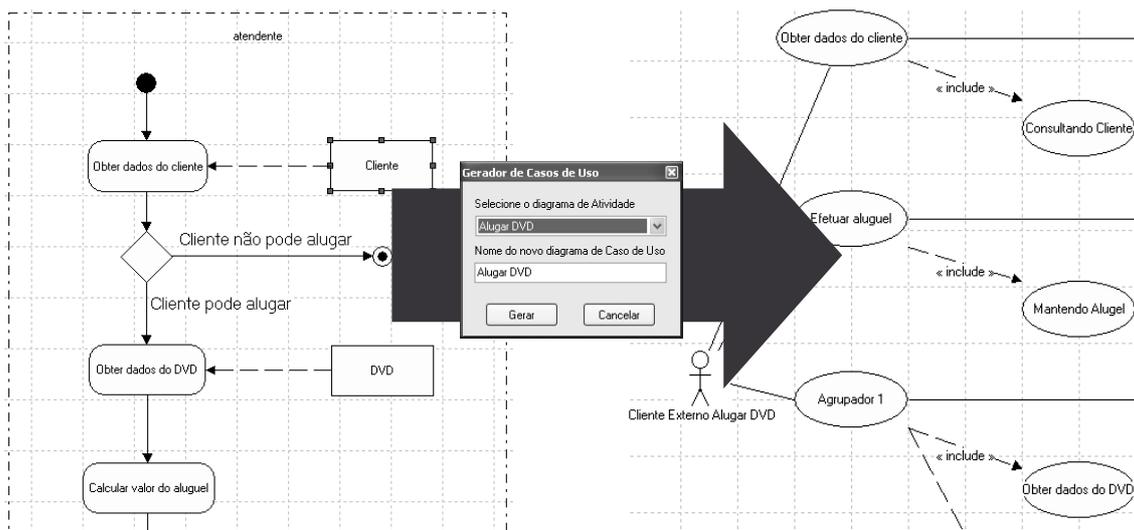


Figura 3. Utilização do Gerador de Casos de Uso.

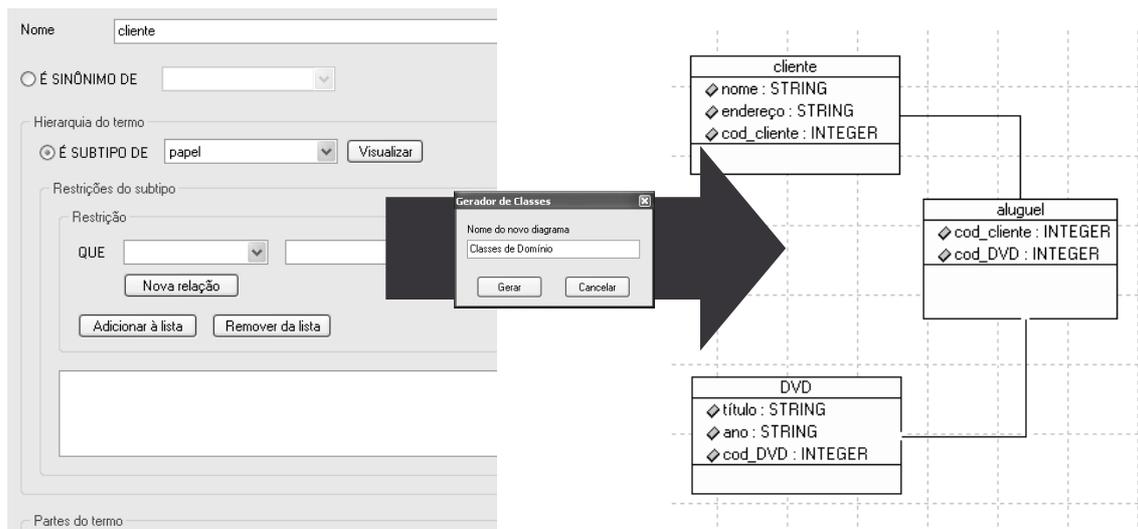


Figura 4. Utilização do Gerador de Classes de Domínio.

A parte relativa ao Modelo de Sistema de Informação permite a edição dos principais diagramas da UML: o diagrama de casos de uso, o diagrama de classes, o diagrama de estados e o diagrama de seqüência. Esta parte apresenta também um gerador de código que transforma estes diagramas no código-fonte do esqueleto da arquitetura do sistema de informação modelado para as plataforma Delphi e Java. A figura a seguir apresenta a utilização deste gerador.

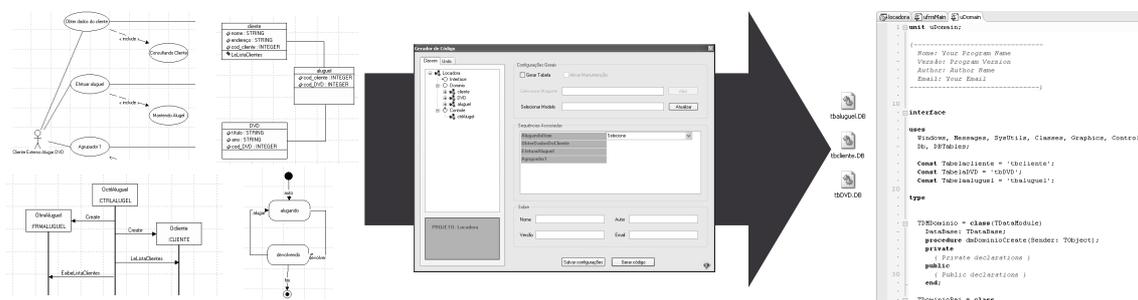


Figura 5. Utilização do Gerador de Código.

5 Considerações Finais

Este artigo apresentou o processo RAPDIS para o desenvolvimento de SI e o ambiente que serve de apoio para este processo. A definição do processo RAPDIS oferece uma forma estruturada de se utilizar o MDA no desenvolvimento de sistemas. A utilização do MDA possibilita um aumento da produtividade, devido às transformações automáticas entre os modelos; da portabilidade, devido à construção de modelos independente de plataforma; da interoperabilidade, devido à possibilidade de criação de pontes entre os modelos específicos de plataforma; e da facilidade de manutenção e documentação, devido à geração automática de código a partir do modelo construído.

A construção do ambiente RAPDIS fornece o suporte computacional necessário para aplicação do processo, automatizando suas atividades. Estudos realizados com um grupo de alunos de graduação mostraram que os usuários apresentam uma percepção positiva do processo e o ambiente RAPDIS. Na visão dos alunos o RAPDIS não só

auxiliou o desenvolvimento de sistemas de informação, como também contribuiu como uma ferramenta de ensino à modelagem de SI.

Como trabalhos futuros apontamos a implementação de um construtor de maquetes dentro do ambiente RAPDIS; a implementação de um mecanismo que permita o rastrear as transformações efetuadas; o suporte ao OCL (*Object Constraint Language*); a geração automática dos casos de teste a partir do modelo (*Model Driven Test*); e a inclusão de “assistentes inteligentes” que permitam a interação do projetista durante o processamento das transformações.

Referências

BLANC, X.; GERVAIS, M.; SRIPLAKINCH, P. (2004) “Model Bus: Towards the interoperability of modeling tools”, Proceedings of MDA: Foundations and Applications.

CRUZ, P. O. S. (2004) “Heurísticas para Identificação de Requisitos de Ambientes de Informações a partir de Modelos de Processos”. Dissertação (Mestrado em Informática), NCE, DCC, IM, UFRJ, Rio de Janeiro RJ, Brasil.

DIAS, F.; MORGADO, G.; OSCAR, P.; SILVEIRA, D.; ALENCAR, A. J.; LIMA, P.; SCHMITZ, E. (2006) “Uma Abordagem para a Transformação Automática do Modelo de Negócio em Modelo de Requisitos”, WER, Rio de Janeiro, RJ, Brasil.

ERIKSSON, H., PENKER, M. (2000), “Business Modeling with UML: Business Patterns at Work”. John Wiley & Sons.

MARSHALL, C. (2000) “Enterprise Modeling with UML” Addison-Wesley.

MARTINS, A. E. (2006) “Em direção à captura e representação sistemática das definições dos termos das Regras de Negócio”. Dissertação (Mestrado em Informática), NCE, DCC, IM, UFRJ, Rio de Janeiro RJ, Brasil.

MELLOR, S., SHLAER, S. (1990) “Análise de Ambientes Orientada para Objetos”. São Paulo: MacGraw-Hill.

MUKERJI, J.; MILLER, J. (2003) “MDA Guide”, Versão 1.0.1, OMG, Disponível em: “<http://www.omg.org/docs/omg/03-06-01.pdf>”. Acesso em: 22 nov. 2005 18:11:00.

PRESSMAN, R. S. (2001) “Software Engineering: A Practitioner's Approach”, McGraw-Hill, Quinta edição.

SANTANDER, V. F. A., CASTRO, J. F. B. (2000) “Desenvolvendo Use Case a partir de Modelagem Organizacional”. WER.

SHNIEDER, G., WINTERS, J. P. (2001) “Applying Use-Case, A practical Guide” Second Edition. Addison-Wesley.

SILVEIRA, D. S.; SCHMITZ, E. A. (2000) “Desenvolvimento de Software Orientado a Objetos”, Rio de Janeiro: Brasport, 2000.

YERGEAU, F.; BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C. M.; MALER, E. (2004) “Extensible Markup Language (XML)”. Versão 1.0, terceira edição, W3C. Disponível em: “<http://www.w3.org/TR/REC-xml/>”. Acesso em: 23 nov. 2005, 18:52:00.