

UM MODELO DE REPOSITÓRIO REFERÊNCIA DE COMPONENTE DE SOFTWARE (RRCS)

Fábio Junior Alves¹

RESUMO

Esse artigo apresenta o projeto de um Modelo de Repositório de Referência de Componentes de Software baseado na especificação *Reusable Asset Specification* (RAS), que tem o objetivo de gerenciar as informações dos diversos componentes de software, tornando-se um ponto central em que os participantes de um processo de desenvolvimento de software consigam registrar e buscar as informações dos componentes reusáveis de uma maneira prática e rápida.

PALAVRAS-CHAVE: Modelo de Repositório; Software; *Reusable Asset Specification*.

1 INTRODUÇÃO

Considerando o crescimento e a competitividade do mercado, as empresas de tecnologia estão cada vez mais preocupadas com a qualidade e a produtividade no processo de desenvolvimento de *software* (REDOLFI, 2004). Diante destas considerações, a academia e a indústria de *software* investigam novas estratégias que buscam melhorias significativas.

Resende, Cunha e Resende (2007) afirmam que, a reutilização passa a ser a melhor estratégia para solucionar muitas dificuldades encontradas no desenvolvimento de *software*. Boehm e Sullivan (2000) completam dizendo que, a

¹ Professor do curso de Sistemas de Informação da Faculdade Atenas-Paracatu-MG.

Reutilização de *Software* se destaca como uma disciplina estratégica, pois contribui para melhorar os índices de qualidade e de produtividade na indústria de *software*.

Souza (2000) apresenta em seu trabalho alguns autores como Lim (1994) e Basili, Briand e Melo (1996) descrevendo alguns métodos que comprovam que a reutilização de componentes de *software* é capaz de aumentar a qualidade e a produtividade do processo de desenvolvimento de *software*. Entre estes métodos, destaca-se o Desenvolvimento de Sistemas Baseados em Componentes (DBC) como uma das principais abordagens para tornar a reutilização de *software* efetiva na indústria.

Para que seja possível incrementar a produtividade, reduzir os custos e melhorar a qualidade das empresas desenvolvedoras de *software* é necessário que os componentes produzidos durante o processo de desenvolvimento estejam disponíveis e acessíveis, assim, o reuso torna-se uma realidade dentro destas empresas (OLIVEIRA, 2008).

Neste sentido, os repositórios de componentes de *software* é uma excelente estratégia para alcançar o reuso, pois auxilia no armazenamento e recuperação de componentes. Ou seja, repositórios de componentes de *software* visam facilitar a reutilização de componentes ao longo do ciclo de vida do *software* com o intuito de atingir as metas de custo efetivo e de produtividade (LUQI; GUO, 1999).

Porém, Melo (2007) afirma que a seleção do componente certo é apresentada como uma das dificuldades enfrentadas no DBC, sendo citada como tarefa difícil a atividade de localizar o componente adequado disponível, que execute as funções desejadas pelo reutilizador. De acordo com Oliveira (2008), isto acontece porque os repositórios que adotam abordagens locais ou centralizadas

inibem a existência de uma expressiva quantidade de componentes, limitando a reutilização, a acessibilidade, a disponibilidade e a escalabilidade, interferindo no processo de desenvolvimento distribuído, em que produtores e consumidores geograficamente dispersos compartilham os componentes.

Já em relação aos repositórios disponíveis na *web* Werner et al. (2002) dizem que, seus componentes são produzidos e armazenados em repositórios dos mais variados tipos e independentes entre si (diferentes formatos de armazenamentos, meios de acesso, capacidades de consultas, etc.). Este processo de armazenamento dificulta a localização dos componentes apropriados para solucionar um problema particular.

Segundo Rosa (2009), um dos problemas em localizar componentes reusáveis, está associado à indefinição quanto à descrição necessária. Isto porque, muitos repositórios têm como base abordagens similares as utilizadas por bibliotecas de *software*, apresentando assim, as mesmas deficiências (OLIVEIRA, 2008). Uma das soluções para o problema apresentado, esta em realizar o empacotamento do material reusável junto a toda informação, ou meta-informação, necessária para o reuso, o que poderia ser feito através da utilização da especificação *Reusable Asset Specification* (RAS), a qual irá promover um conjunto de regras e recomendações sobre estrutura, conteúdo e descrições de artefatos reusáveis de *software*. (ROSA, 2009; EZRAN; MORISIO; TULLY, 2002).

Considerando a disponibilidade da Web como o meio de busca por informações mais utilizado e as dificuldades anteriormente citadas, este trabalho propõe um Modelo de Repositório de Referência de Componentes de *Software* (RRCS) baseado na especificação RAS. Tal repositório objetiva integrar os diversos repositórios de componentes e gerenciar informações sobre componentes de

software, tornando-se um ponto central com o qual os participantes de um processo de desenvolvimento de *software* conseguem registrar e buscar informações dos componentes reusáveis de uma maneira prática e rápida.

Esta seção apresentou uma contextualização e motivação sobre o tema integração de repositórios de componentes de software, juntamente com o respectivo objetivo do trabalho. O artigo encontra-se subdividido nas demais seções, conforme descrito a seguir.

A seção 2 apresenta os trabalhos relacionados ao tema do artigo. A seção 3 apresenta um referencial teórico juntamente com conceitos básicos, proporcionando ao leitor uma visão geral sobre a fundamentação teórica do trabalho. A seção 4 trata sobre a metodologia empregada, evidenciando o tipo de pesquisa, definindo os aspectos teóricos e procedimentos metodológicos. A seção 5 apresenta o modelo proposto utilizado para a integração entre os repositórios. Na seção 6 temos a conclusão do trabalho. E por fim na seção 7 temos as referências bibliográficas utilizadas.

2 TRABALHOS RELACIONADOS

Esta seção tem por objetivo apresentar alguns exemplos de trabalhos similares que buscam soluções para os problemas de integração de repositórios de componentes no apoio ao processo de reutilização.

Iniciamos analisando o trabalho Agora System, que utiliza uma abordagem similar a de repositórios de referência (SEACORD; HISSAM; WALLNAU, 1998). O Agora é um sistema que utiliza um motor de busca para

recuperar componentes reutilizáveis disponíveis na internet, tais como JavaBeans e CORBA. Este sistema utiliza um mecanismo de introspecção para registrar as informações dos componentes. Porém, estes mecanismos apresentam algumas limitações, tais como: a falta de informações descritivas acerca dos componentes, de forma que a busca seja mais precisa. Assim, uma das soluções propostas seria adicionar mais semântica ao componente, de forma que a busca seja mais precisa e facilitando o acesso ao componente pelo usuário.

Outro trabalho relacionado à seleção de componentes foi o de Braga, Mattoso e Werner (1999). Os autores abordam o problema da interoperabilidade entre repositórios de componentes em um ambiente de reutilização, propondo uma camada de integração que foi desenvolvida para auxiliar na busca e identificação de componentes reutilizáveis. Esta camada é baseada na utilização de mediadores e ontologias, proporcionando assim flexibilidade e transparência na recuperação dos componentes. Porém, o projeto desta arquitetura é voltado apenas para integração de sistemas gerenciadores de banco de dados, não levando em conta a integração de uma fonte de informação qualquer na Internet.

Seguindo a mesma lógica de raciocínio dos trabalhos anteriores, Oliveira, Garcia e Nunes (2007) propõem a criação da Rede de Compartilhamento de Componentes *Softwares* (RCCS), que baseia-se em padrões abertos para comunicação, arquitetura e modelagem dos componentes. O principal objetivo desse trabalho é proporcionar às entidades associadas ao desenvolvimento e utilização de componentes de software, uma participação como fornecedoras ou consumidoras dos componentes distribuídos na rede. A rede deve atuar como um repositório ou biblioteca de componentes, de forma totalmente distribuída e independente de entidades centrais.

3 FUNDAMENTAÇÃO TEÓRICA

Esta seção tem por objetivo definir e apresentar a base conceitual fundamentando os pontos essenciais para o trabalho realizado. Sua adequada elaboração tem o propósito de facilitar o entendimento do assunto aqui tratado.

3.1 REUTILIZAÇÃO DE COMPONENTES DE SOFTWARE

Considerando o crescimento e a competitividade do mercado, a cada dia que passa as empresas de tecnologia deparam-se com o desafio de construir produtos cada vez maiores, mais complexos e mais confiáveis (JACOBSON; GRISS; JONSSON, 1997).

Diante deste cenário, o conceito de reuso torna-se crucial para que as empresas de software possam produzir seus produtos com maior qualidade e em menos tempo (REDOLFI, 2004).

O trabalho de Tracz (1986, citado por ROSSI, 2004) descreve dados de um estudo relativo ao reuso de código-fonte em diferentes projetos:

- a) 40% a 60% de todo o código são reutilizáveis de uma aplicação para outra;
- b) 60% de projeto e código de aplicações de negócio são reutilizáveis;
- c) 75% das funções de programas são comuns a mais de um programa e;
- d) 15% de código encontrado na maioria dos programas são únicos e

novos em uma aplicação específica.

A partir dessas considerações, nota-se que a reutilização propõe o desenvolvimento de software explorando eficaz e eficientemente as semelhanças entre requisitos e arquiteturas de diversos sistemas. Com a aplicação do reuso e a identificação dessas semelhanças, empresas podem obter melhores resultados no desenvolvimento de softwares, reduzindo tempo e custo de desenvolvimento. Adicionalmente, verifica-se a validade e eficiência das unidades de software a cada aplicação, promovendo um aumento contínuo da qualidade (HOLANDA; SOUZA; MELO, 2001).

Segundo Gimenes e Huzita (2005), os maiores benefícios da reutilização de software se referem ao aumento da qualidade e redução do esforço de desenvolvimento. Por outro lado, os maiores obstáculos são custos de manutenção aumentados, falta de apoio de ferramentas, criação e manutenção de uma biblioteca de componentes e a procura, compreensão e adaptação de componentes reusáveis (SOMMERVILLE, 2007).

3.2 REPOSITÓRIO DE COMPONENTES DE SOFTWARE

Para tornar realidade o reuso de *software nas* empresas, é necessário que os componentes de *software* possam estar amplamente disponíveis e acessíveis, a qualquer momento. Neste sentido, Rossi (2004) diz que vários pesquisadores mostram a necessidade de existir uma ferramenta que forneça suporte ao processo de reuso.

Holanda, Souza e Melo (2001) afirmam ainda que, quando os

componentes são bem organizados dentro de seus repositórios, são facilmente divulgados e compartilhados com outros desenvolvedores. Isto faz com que os repositórios de componentes de *software* atuem como ferramenta de suporte para a prática do reuso.

Dentre os repositórios existentes, Sametinger (1997) distingue três tipos:

a) **repositórios locais**: são utilizados para armazenar componentes de propósito geral;

b) **repositórios específicos a um domínio**: este tipo de repositório é utilizado para manter componentes relacionados a um domínio específico, fornecendo componentes alternativos para tarefas específicas;

c) **repositórios de referência**: utilizados para auxiliar na localização de componentes em outros repositórios. Ou seja, este tipo de repositório arquiva a referência do repositório que contém o componente.

De acordo com Santos (2010), o advento da internet e a disponibilidade de informações distribuídas e heterogêneas, contribuíram para o crescimento de repositórios de referência.

Completando os dizeres de Santos (2010), Gimenes e Huzita. (2005) afirmam ainda que alguns autores da área vêem a interconexão de repositórios através dos repositórios de referências como uma abordagem promissora, pois este tipo de repositório trabalha como um catálogo de repositórios, registrando informações dos componentes e repositórios ao qual o componente pertence, podendo ser consultado a qualquer momento pelos consumidores de componentes de *software*.

3.3 O PADRÃO RAS

De acordo com Sametinger (1997), para que a busca de componentes seja realizada de maneira eficiente, é preciso garantir que estes componentes contenham as informações necessárias à sua recuperação. Desta maneira, na construção destes componentes é preciso preocupar-se com as informações sobre o componente que devem ser armazenadas. Diante destas considerações, o *Object Management Group* (OMG) propôs o padrão *Reusable Asset Specification* (RAS), utilizado para descrever um conjunto de regras e recomendações a que todos os artefatos reusáveis de software devem atender.

Segundo o padrão RAS, artefatos reusáveis de software podem ser código-fonte, classes, casos de uso, etc., ou seja, qualquer produto resultante do processo de desenvolvimento de um software. Já um componente de software representa um ou mais artefatos, que um bem possui (OMG, 2005).

Segundo a OMG (2005), o padrão RAS pode ser utilizado tanto para descrição de componentes quanto como um guia no desenvolvimento de repositórios baseados nesta especificação. De acordo com Miranda *et al.*, (2008), os repositórios ampliam sua capacidade de reutilização e a visibilidade de um componente quando são compatíveis com a especificação, devido à possibilidade de trocas de informações.

De acordo com a OMG (2005), a RAS é definida em duas categorias: *Core RAS* e *Profiles*. O *Core RAS* identifica elementos fundamentais da especificação do componente de *software*; já os *Profiles* representam extensões a estes elementos fundamentais.

O *Core RAS* possui um *Asset* cujo objetivo é utilizado para descrever o bem, através dos seus atributos como nome, id, status, etc (OMG, 2005).

O principal objetivo do RAS é especificar a quantidade mínima de meta informações que um bem deve possuir. Sendo assim, o RAS armazena suas informações em um arquivo chamado *manifest*, que é uma descrição em XML de várias características presentes no bem. Esta descrição é utilizada, pois facilita sua extensão uma vez que um arquivo XML é facilmente extensível.

4 METODOLOGIA DE PESQUISA

Sempre que se inicia uma pesquisa científica, o primeiro passo deve ser a delimitação do método científico e do tipo de pesquisa, pois de acordo com Vergara (2004) “o método é um caminho, uma forma, uma lógica de pensamento”, ou seja, é o método da pesquisa que norteia o pesquisador na realização de seu trabalho.

Sendo assim, esta seção descreve o tipo de pesquisa e os procedimentos planejados e realizados para o desenvolvimento do presente trabalho.

3.1 TIPO DE PESQUISA

Esta pesquisa classifica-se como qualitativa, de natureza tecnológica, pois segundo Jung (2004) e Silva e Menezes (2000) este trabalho, objetiva combinar conhecimentos para gerar um novo produto e modificar práticas existentes.

Em relação aos objetivos, é de caráter exploratório e descritivo pois visa os estudos exploratórios iniciais e a descoberta de novas práticas, relacionadas a

desenvolvimento baseado em componentes e reusabilidade de software (GIL, 2002; JUNG, 2004).

Quanto aos procedimentos, é um estudo de caso, Jung (2004), permitindo aplicar o modelo proposto dentro de um contexto local e real, possibilitando definir limites mais claramente, facilitando sua validação e avaliação.

3.2 PROCEDIMENTOS METODOLÓGICOS

O processo de estudo e especificações do referente trabalho podem ser abordados de maneira geral em quatro etapas, nas quais cada uma das etapas apresentam objetivos e requisitos específicos em suas especificações.

Na etapa 1 temos basicamente a definição do problema uma das etapas fundamentais do processo. Para definição desta etapa foram feitos vários estudos envolvendo repositórios de componentes de software, averiguando quais eram os principais assuntos que envolviam, quais foram às ferramentas que estavam sendo desenvolvidas, etc.

Na etapa 2 foi feito uma seleção de todos os trabalhos relacionados ao tema proposto. Buscamos identificar na literatura alguns trabalhos relacionados a este trabalho, como forma de comparação dos resultados posteriormente.

A etapa 3 refere-se à definição do modelo RRCS, descrevendo sua aplicação. Para fins de validação do modelo RRCS, pretende-se realizar um estudo de caso envolvendo a empresa Sensedia e a Fábrica de Software da Faculdade Atenas. Como o foco principal deste trabalho é integrar os diversos repositórios de componentes, será criado dois repositórios fictícios diversificando o modelo do *Rasset.xml*, com isso será

garantido a validação do modelo proposto.

5 O MODELO DE REPOSITÓRIO REFERÊNCIA DE COMPONENTE DE SOFTWARE – RRCS

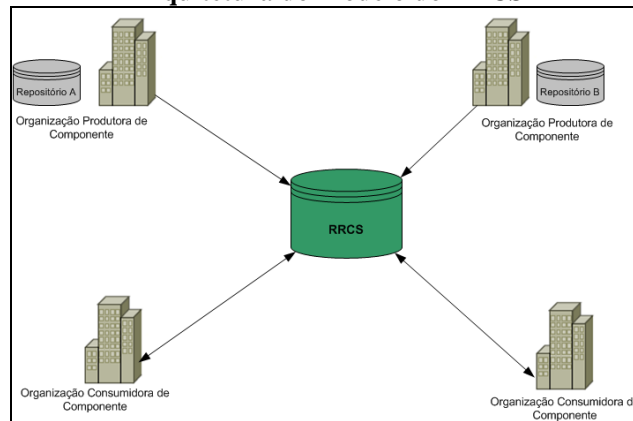
Esta seção descreve a modelagem do Repositório de Referência de Componentes de Software – RRCS principal contribuição deste trabalho.

O RRCS é um Repositório de Referência de Componentes de *Software*, que tem como objetivo gerenciar as informações dos componentes através da divulgação e busca das informações dos componentes de *software*.

Visou-se modelar um Repositório de Referência que pudesse trazer, de fato, funcionalidades que impulsionem e incentivem a utilização de componentes. Para isso, é proposto que qualquer organização possa participar como produtoras de componentes de *software* e/ou consumidora dos componentes cadastrados no RRCS.

A figura 1 ilustra a forma com que o modelo RRCS foi projetado, para auxiliar as organizações interessadas em compartilhar ou simplesmente buscar informações dos componentes de *software*.

FIGURA 1
Arquitetura do Modelo do RRCS



Fonte: Elaboração do autor.

Conforme é mostrado pela figura 1, os produtores de componentes são organizações que são definidas como os responsáveis pela publicação das informações dos componentes que estão hospedados nos repositórios. Por outro lado, os consumidores são organizações que adquirem os componentes disponibilizados nos repositórios. Estes papéis, porém, não são exclusivos. Um usuário do RRCS pode ser um produtor, publicando as informações do seu componente, e também um consumidor, que adquiriu um componente para a construção do seu próprio.

Através deste modelo como é possível ver as organizações conseguiram compartilhar os componentes de software de maneira rápida e prática.

6 CONCLUSÕES

A fim de superar as limitações das propostas existentes, este trabalho propôs uma nova solução para o problema da integração de repositórios de componentes de software em ambientes de reuso. Este trabalho apresentou um modelo de repositório de referência de componentes de software denominado (RRCS), que tem o objetivo

gerenciar as informações dos diversos componentes de software, tornando-se um ponto central em que os participantes de um processo de desenvolvimento de software consigam registrar e buscar as informações dos componentes reusáveis de uma maneira prática e rápida, com isso conseguindo a integração dos diversos repositórios de componentes existentes.

Conforme foi discutido neste trabalho, para a reutilização de componentes de software aconteça é necessário que existam componentes de software reutilizáveis e que os projetistas tenham acesso a eles. Sendo assim, espera-se que este trabalho possa contribuir com o processo de localização de componentes, tornando esta tarefa mais simples, agilizando assim, o processo de construção de sistemas.

7 REFERÊNCIAS

REDOLFI, G. Especificação de uma arquitetura para repositórios de informações de componentes de software. 2004. Dissertação de Mestrado. Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2004.

RESENDE, A. R. M. de L; CUNHA, A. M.; RESENDE, A. M. P. de. Um modelo de processo para seleção de componentes de software. Lavras: Ed. UFLA, 2007.

BOEHM, B.; SULLIVAN, K. J. Software Economics: A Roadmap. In: 22nd INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 2000, Limerick, Ireland. Proceedings... Limerick, Ireland, 2000. p. 319-344.

BASILI, V.R.; BRIAND, C.; MELO, W.L.. How reuse influences productivity in object-oriented systems. Communications of the ACM, v.39, n.10, p.104-126, Oct. 1996.

OLIVEIRA, J. P. F. de.. X-CORE: um serviço de repositório distribuído e compartilhado de componentes de software. 2008. 111 f. Dissertação (Mestrado em Informática) – Centro de Ciências Exatas e da Natureza, Universidade Federal da Paraíba, João Pessoa, 2008.

LUQI GUO, J.. Toward Automated Retrieval for a Software Component Repository. In: PROCEEDINGS OF THE CONFERENCE AND WORKSHOP ON ENGINEERING OF COMPUTER-BASED SYSTEMS, 1999, Nashville. Proceedings... Nashville: [s.n.], 1999. p. 99-105.

MELO, C. de O.. Classificação semi-automática de componentes java. 2007. 101 f. Dissertação (Mestrado em Ciência da Computação) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2007.

WERNER, C. et al.. OdysseyShare: Um ambiente para o Desenvolvimento Cooperativo de Componentes. In: XVI SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 16., 2002, Gramado, RS. Anais... Gramado, RS: [s.n.], 2002. Disponível em < <http://reuse.cos.ufrj.br/prometeus/publicacoes/OdysseyShare-TS2002.pdf> >. Acesso em: 30 de Mai 2011.

ROSA, F. R. da. RASPUTIN. Uma Infra-estrutura de Suporte para Promover o Reuso de Software através do Padrão RAS. Universidade Federal Do Rio Grande Do Sul Instituto De Informática Curso De Ciência Da Computação. Porto Alegre, 2009.

RAS - Reusable Asset Specification, OMG Available Specification. Object Management Group, 2011. Disponível em: <<http://www.omg.org/technology/documents/formal/ras.htm>> . Acesso em: 5 jan. 2011.

EZRAN, M.; MORISIO, M.; TULLY, C.. Practical Software Reuse. London: Springer, 2002. 216 p.

SEACORD, R. C.; HISSAM, S. A.; WALLNAU, K. C. Agora: A search engine for software components. Software Engineering Institute, v. 2, n. 6, p. 1-24, Dec. 1998.

BRAGA, R. M. M.; MATTOSO, M.; WERNER, C. M. L.. The use of mediators for component retrieval in a reuse environment. In: WORKSHOP ON COMPONENT-BASED SOFTWARE ENGINEERING PROCESS – IEEE, 1999, California. Proceedings...California: [s. n.], 1999. p. 542–546.

OLIVEIRA, M.; GARCIA, I.; NUNES, A. RCCS: uma Rede de Compartilhamento de Componentes de Software. In: XXV SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 2007, Belém, Pará. Anais... Belém, Pará: [S. n.], 2007.

JACOBSON, I.; GRISS, M.; JONSSON, P.. Software reuse: architecture, process and organization for business success. New York: ACM Press, 1997. 497 p.

REDOLFI, G. Especificação de uma arquitetura para repositórios de informações de componentes de software. 2004. Dissertação de Mestrado. Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2004.

TRACZ, W. Software Reuse: motivators and inhibitors. In: WORKSHOP ON FUTURE DIRECTIONS IN COMPUTER ARCHITECTURE AND SOFTWARE, 1986, Charlston, SC. Proceedings... Charlston, SC: Stanford University, 1986.

HOLANDA, C. B. S.; SOUZA, C. A. de A. de; MELO, W. L. ProReuso: Um Repositório de Componentes para Web Dirigido por um Processo de Reuso. In XV SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 2001, Rio de Janeiro. Simpósio...Rio de Janeiro: [S. n.], 2001. p. 208-223.

GIMENES, I. M. de S.; HUZITA, E. H. M.. Desenvolvimento baseado em componentes: conceitos e técnicas. Rio de Janeiro: Ciência Moderna, 2005. 304 p.

SOMMERVILLE, I.. Engenharia de Software. 8 ed. São Paulo: Pearson Addison-Wesley, 2007.

SAMETINGER, J. Software engineering with reusable components. New York: Springer, 1997. 271p.

ROSSI, A. C. Representação do componente de software na FARCISOFT: Ferramenta de Apoio à Reutilização de Componentes de Software. 2004. 237 f. Dissertação (Mestrado em Engenharia) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2004. Disponível em: < <http://www.teses.usp.br/teses/disponiveis/3/3141/tde-03062004-101200/pt-br.php>>. Acesso em: 21 Jan. 2011.

SANTOS, R. P. dos. BRECHÓ-VCM: uma abordagem baseada em valor para mercados de componentes. 2010. Dissertação (Mestrado em Engenharia de Sistemas e Computação). Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.

OMG. 2005. Reusable Asset Specification. Disponível em <<http://www.omg.org/docs/formal/05-11-02.pdf>>. Acesso em: 25 fev. 2010.

VERGARA, S. C. Projetos e relatórios de pesquisa em administração. 5. ed. São Paulo: Atlas, 2004.

SILVA, E. L.; MENEZES, E. M. Metodologia da pesquisa e elaboração de dissertação. Florianópolis: [S.n.], 2000.

JUNG, C. F. Metodologia para pesquisa e desenvolvimento: aplicada a novas tecnologias, produtos e processos. [S.l]: Axcel Books do Brasil, 2004. 328 p.

GIL, A. C. Como elaborar projetos de pesquisa. 4. ed. São Paulo: Atlas, 2002.