

Brechó: Catálogo de Componentes e Serviços de Software

Cláudia Werner¹, Leonardo Murta¹, Marco Lopes¹, Alexandre Dantas^{1,2},
Luiz Gustavo Lopes^{1,3}, Paula Fernandes¹, João Gustavo Prudêncio¹,
Anderson Marinho¹, Ronaldo Raposo¹

¹PESC/COPPE – Universidade Federal do Rio de Janeiro
Caixa Postal 68511 – 21945-970 – Rio de Janeiro – RJ – Brasil

²Petrobras – Petróleo Brasileiro S.A.
Av. República do Chile, 65 – 20031-912 – Rio de Janeiro – RJ – Brasil

³Banco Central do Brasil - Setor Bancário Sul (SBS) Quadra 3 Bloco B
Caixa Postal 08670 CEP. 70074-900 - Brasília - DF

{werner,murta,mlopes,paulacibele,gustavo}@cos.ufrj.br,
{ardantas,andymaninho,ronaldo.raposo}@ufrj.br, luiz.lopes@bcb.gov.br

Abstract. *Component-Based Software Engineering rises as a promising technique for software development in the context of Software Reuse. The complete isolation among component's development and consumers teams can lead to failure of a component-based reuse program. Component libraries are an alternative way to narrow the communication channel between these teams. This paper presents a component library called Brechó, which catalogs components to be further searched and retrieved. The components are cataloged respecting a level-based structure, which makes an automated component packing possible.*

Resumo. *Dentre as técnicas que constituem a disciplina de Reutilização de Software, a Engenharia de Software baseada em Componentes se apresenta como escolha promissora nas organizações que desenvolvem software. O isolamento entre as equipes que desenvolvem e consomem componentes pode acarretar no fracasso de uma política relacionada a componentes. Bibliotecas de componentes se apresentam como uma alternativa para estreitar o canal de comunicação entre essas equipes. Este artigo apresenta a biblioteca de componentes Brechó, que realiza a catalogação de componentes para busca e recuperação. Os componentes são catalogados respeitando uma estrutura de níveis de perspectiva, viabilizando o empacotamento automatizado destes.*

1. Introdução

A Reutilização de software é o uso de software existente ou conhecimento referente a este para construir um novo software [6]. É considerada uma forma de obter melhores indicadores de qualidade, produtividade e manutenibilidade na construção de sistemas complexos, confiáveis e baratos. Para tal, é necessário que esta seja pré-planejada, ampla e formal, envolvendo a sistematização do ciclo de vida do software [6][20].

De acordo com as pesquisas em reutilização de software [6], uma técnica potencialmente promissora é a engenharia de software baseada em componentes [19], que visa a definição de componentes interoperáveis e com interfaces bem definidas, evidenciando os tipos de relacionamentos permitidos por estes componentes.

A engenharia de software baseada em serviços também se destaca como técnica promissora. Os serviços podem ser compostos, orquestrados e invocados remotamente através de interfaces e protocolos padronizados, sendo possível implementá-los concretamente por componentes [18].

No Brasil, a engenharia de software baseada em componentes e serviços é reconhecida pelo governo como uma área estratégica para a indústria brasileira de software [17]. Um mercado de componentes amadurecido, entretanto, ainda não é uma realidade no país. Inibidores críticos a este amadurecimento incluem a baixa disponibilidade de componentes e a imaturidade dos canais de distribuição [2]. Para que o mercado nacional amadureça, torna-se fundamental aumentar a oferta de componentes de software de boa qualidade, que sejam facilmente encontrados, compreendidos, adquiridos e reutilizados a partir de bibliotecas disponibilizadas na Internet.

Desta forma, este artigo apresenta um elemento fundamental neste cenário: a biblioteca de componentes Brechó, que foi concebida visando atender a características de catalogação, busca e recuperação de componentes. O conceito de componente é flexibilizado na Brechó, o que permite sua utilização em diversos contextos.

O artigo está organizado da seguinte maneira: a Seção 2 apresenta a biblioteca Brechó; os trabalhos relacionados à biblioteca Brechó são expostos na Seção 3; e a Seção 4 conclui o artigo apresentando alguns trabalhos futuros.

2. A Biblioteca de Componentes Brechó

Uma biblioteca de componentes de software deve representar um local de publicação busca e recuperação de componentes e serviços, de forma que possam ser aplicados no desenvolvimento de novos sistemas [20]. Adicionalmente, a aquisição de componentes pode ser mediada por um contrato entre fornecedores e consumidores, provendo papéis e responsabilidades bem definidas para questões de tarifação, propriedade intelectual, qualidade e manutenção. Com o desenvolvimento de uma comunidade ao redor da reutilização de componentes, o amadurecimento técnico dos próprios componentes e de relações mais homogêneas em um mercado de componentes de software é favorecido.

2.1 Mecanismos Fundamentais da Brechó

A biblioteca Brechó, em seu núcleo inicial, é um sistema de informação para Web com uma base de dados de componentes, fornecedores e consumidores. Este núcleo define os mecanismos de documentação, armazenamento, pesquisa, e recuperação.

Os mecanismos de publicação e documentação consideram um conceito flexível de componente, além do binário, visando uma representação que inclua os possíveis artefatos produzidos durante o desenvolvimento do componente (como especificações, código fonte, manuais, etc.). Desta forma, a Brechó torna-se capaz de permitir a aquisição de diferentes combinações de conjuntos de artefatos empacotados, a partir de licenças personalizadas e configuráveis.

A estrutura de documentação é fundamentada em categorias e formulários dinâmicos e configuráveis ao invés de um formato prefixado. Desta forma, é possível classificar um componente em várias categorias, além de criar e associar diferentes formulários de documentação a cada categoria, permitindo a construção da documentação do componente como um mosaico.

A flexibilidade de publicação dos componentes influencia diretamente os mecanismos de pesquisa e recuperação. Inicialmente, a biblioteca permite a busca por componentes por filtros de categorias e por palavra-chave. A partir dos resultados de pesquisa por componentes, torna-se possível a sua disponibilização (*download*) para os consumidores, de acordo com as políticas de recuperação definidas pelos fornecedores.

2.2 Organização Interna da Brechó

A organização interna da Brechó é dividida em níveis (a), exibidos na Figura 1, para representação de um componente, bem como os cortes associados a cada nível (b) e um exemplo (c) destes cortes. Os níveis definidos são: **Componente**, **Distribuição**, **Release**, **Pacote**, **Licença**.

O nível **Componente** representa conceitualmente as entidades armazenadas na Brechó, sem as informações concretas sobre as implementações dessas entidades. O nível **Distribuição** representa um corte funcional sobre as entidades, fornecendo conjuntos de funcionalidades que são desejadas por grupos específicos de usuários. O nível

Release representa um corte temporal sobre as distribuições, no qual define versões dos artefatos que implementam as entidades em um determinado instante no tempo. A partir desse nível, as entidades passam a ter informações concretas sobre suas implementações. Essas implementações concretas das entidades usualmente existem em diferentes níveis de abstração (e.g.: documentação do usuário, análise, projeto, código, binário, etc.), e diferentes empacotamentos podem ser definidos para possibilitar a reutilização de parte dos níveis de abstração disponíveis (e.g. documentação do usuário e binário). Desta forma, o nível **Pacote** permite que seja feito um corte em níveis de abstração, possibilitando que sejam agrupados artefatos de acordo com o público alvo de reutilização. O nível **Licença** possibilita a definição de níveis de serviço sobre os pacotes. Para cada pacote podem ser estabelecidas licenças específicas, que garantem regras entre os produtores e os consumidores dos componentes.

Além disso, dentro de uma biblioteca de componentes, torna-se necessário deixar bem claras as relações entre componentes. A relação de dependência estabelece quais componentes são requeridos para que um determinado componente funcione adequadamente. Tendo esta preocupação, a Brechó provê ferramentas para o estabelecimento e inferência de dependências em diferentes níveis de abstração. Para viabilizar o uso das dependências, a Brechó fornece ferramentas de apoio à recuperação das dependências de um componente.

2.3 Oferta de Serviços WEB

A execução remota de serviços de componentes de software (compilados) disponíveis se apresenta como uma das funcionalidades da Brechó. Esta execução é viabilizada por meio de Serviços Web [3], que são associados ao componente no momento do seu cadastro.

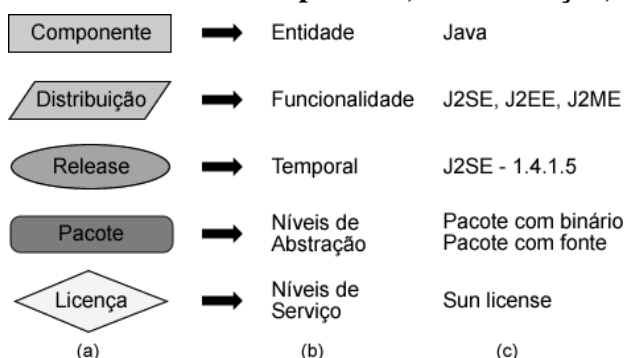


Figura 1 - Organização Interna da Brechó

A biblioteca de componentes Brechó também oferece parte das suas próprias funcionalidades por meio de Serviços Web. Dessa forma, a execução de serviços específicos a determinados componentes da Brechó pode ser realizada de maneira automática por outras ferramentas. Atualmente, esses serviços estão sendo utilizados na integração da biblioteca com o ambiente de reutilização Odyssey [13]. O mecanismo de carga dinâmica do Odyssey [11] permite que novos componentes sejam recuperados da Brechó e adicionados em tempo de execução ao ambiente.

2.4 Mapa de Reutilização

Para auxiliar na atividade de identificação de responsabilidades de manutenção, a Brechó possui o Mapa de Reutilização [10]. O Mapa de Reutilização mantém as informações dos contratos firmados entre os produtores e consumidores, para cada componente reutilizado. Ele também possui um cadastro de licenças de software, e cada contrato de reutilização está associado a uma licença, que descreve de forma mais genérica as regras de utilização de um componente.

Quando um componente é publicado, são cadastrados o seu fornecedor e as dependências que ele possui com outros componentes. Quando um componente é adquirido, o usuário seleciona uma das licenças disponíveis para ele e confirma a criação do contrato de reutilização, tornando-se um consumidor do componente.

De posse destas informações, as equipes envolvidas no processo de manutenção podem determinar de quem é a responsabilidade por uma modificação específica em um componente, através da identificação do produtor e das cláusulas contratuais referentes àquele componente. Também podem consultar quem são os consumidores, identificar quais são os que podem ter uma nova *release*, e disseminar informações a respeito de novas versões e modificações realizadas nos componentes de interesse.

2.5 Implementação

A biblioteca de componentes de software Brechó é uma aplicação web implementada com a tecnologia Java J2EE, utilizando o *framework* Struts [1] para implementar uma arquitetura MVC (*Model View Controler*). Para a persistência de objetos, foi utilizado o *framework* Hibernate [14] e o banco de dados MySQL [12].

Na versão atual da Brechó, existem dois papéis de usuários: (i) administrador, que tem poderes para cadastro de usuários e administração em geral e (ii) usuário, que é quem cadastra e recupera os componentes disponíveis. A ferramenta Brechó está disponível em <http://reuse.cos.ufrj.br/brecho>. A Figura 2 apresenta sua tela inicial na visão do administrador.

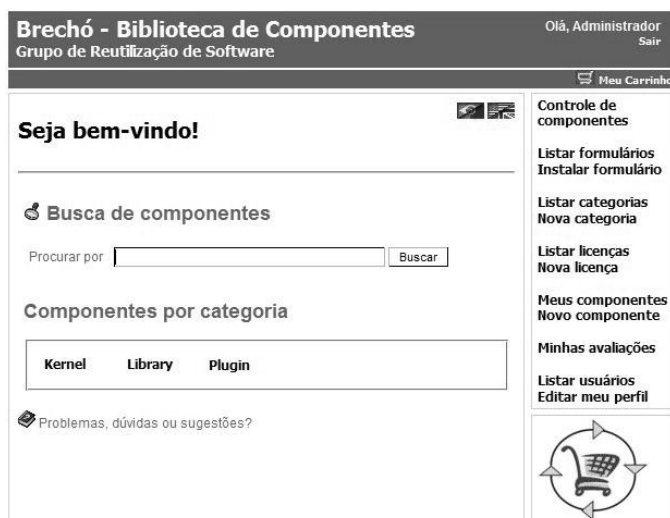


Figura 2 - Tela Inicial (visão administrador)

3. Trabalhos Relacionados

Atualmente, muitos grupos de pesquisa (e.g: [5] e [15]) estão trabalhando em temas relacionados à biblioteca de componentes. Porém, trabalhos destes grupos não levam em consideração o empacotamento automático e flexível dos artefatos de um componente, além de não implementarem em conjunto os conceitos de mapa de reutilização e oferta de serviços.

Na web, existem muitos repositórios de componentes disponíveis para uso [4][8][7]. Este conjunto de repositórios se caracteriza por catalogar apenas componentes já empacotados ou de prateleira.

Finalmente, *Logidex* [9] e *Select Component Manager* [16] são gerenciadores comerciais de componentes que mantêm informações sobre os componentes reutilizáveis de uma organização. Eles armazenam informações como o consumo dos componentes e suas interdependências. Diferentemente da Brechó, esses gerenciadores consideram que cada componente é reutilizado sob a mesma licença por todos os seus consumidores.

4. Conclusões

Este artigo apresentou a biblioteca de componentes Brechó, que tem como objetivo estabelecer um canal entre produtores e consumidores em que componentes possam ser catalogados e recuperados. Estes mecanismos são potencializados através das seguintes características: (i) um mecanismo de mapa de reutilização; (ii) uma estrutura de organização interna de componentes definida em níveis; e (iii) a possibilidade de disponibilizar serviços da biblioteca através de Serviços Web.

Como trabalhos futuros, vislumbra-se a criação de um mecanismo de sugestão e administração de categorias, auxiliando os usuários a encontrar uma categoria em que seu componente possa ser mais bem enquadrado. Além disso, novas funcionalidades relacionadas à busca de componentes serão implementadas, como a utilização de filtros para auxiliar os usuários a realizar uma busca e um esquema de correção de palavras.

Limitações relacionadas à oferta de serviços Web estão sendo tratadas em trabalhos em andamento. Atualmente, o produtor do componente pode cadastrar o endereço para os serviços oferecidos pelo componente, porém a Brechó não possui nenhum controle sobre a chamada destes serviços. Na implementação em desenvolvimento, a Brechó irá fornecer mecanismos mais sofisticados de publicação e geração automatizadas de serviços Web, tornando possível a tarifação sobre os mesmos.

Agradecimentos

Os autores agradecem à CAPES e ao CNPq pelo apoio financeiro, e aos participantes dos projetos Brechó e Odyssey que direta ou indiretamente ajudaram neste trabalho.

Referências Bibliográficas

- [1] APACHE, Apache Struts. Disponível em <<http://struts.apache.org/>>. Visitado em 17 de Maio de 2007.
- [2] BASS, L. et al., “Market Assessment of Component-Based Software Engineering, Technical Report, Software Engineering Institute”, CMU/SEI-2001-TN-007, 2000.

- [3] BOOTH, D., HAAS, H., MCCABE, F., et al. "Web Services Architecture - W3C Working Group Note". Disponível em <<http://www.w3.org/TR/ws-arch>>. Visitado em 19 de Maio de 2007
- [4] COMPONENTSOURCE, Repositório de Componentes ComponentSource. Disponível na em <<http://www.componentsource.com/>>. Visitado em 17 de Maio de 2007.
- [5] COMPOSE, Grupo de Pesquisa COMPOSE (UFPB). Disponível em <<http://compose.ufpb.br/>>. Visitado em 17 de Maio de 2007.
- [6] FRAKES, W.B., KYO KANG, K., "Software Reuse Research: Status and Future", IEEE Transactions on Software Engineering, v.31, n.7, 2005.
- [7] FUNCTIONX, Repositório de Componentes Visual Component Library. Disponível em <<http://www.visualcomponentlibrary.com/>>. Visitado em 17 de Maio de 2007.
- [8] JIDE, Repositório de Componentes Jide Software. Disponível em <<http://www.jidesoft.com/>>. Visitado em 17 de Maio de 2007.
- [9] LOGIC LIBRARY, Logidex. Disponível em <<http://www.logiclibrary.com>>. Visitado em 07 de Maio de 2007.
- [10] LOPES, L.G.B., MURTA, L.G.P., WERNER, C.M.L., Odyssey-CCS: A Change Control System Tailored to Software Reuse. In: International Conference on Software Reuse, Torino, Itália, 2006, pp. 170-183.
- [11] MURTA, L.G.P., VASCONCELOS, A.P.V., BLOIS, A.P.T.B., et al., "Run-time Variability through Component Dynamic Loading". In: XVIII Simpósio Brasileiro de Engenharia de Software, Sessão de Ferramentas, Brasília, DF, Brasil, outubro, 2004, pp. 67-72.
- [12] MYSQL AB, Sistema Gerenciador de Banco de Dados MySQL. Disponível em <<http://www.mysql.com>>. Visitado em 19 de Maio de 2007.
- [13] ODYSSEY, Projeto Odyssey. Disponível em <<http://reuse.cos.ufrj.br/odyssey>>. Visitado em 06 de Maio de 2007.
- [14] REDHAT, *Framework* Hibernate. Disponível em <<http://www.hibernate.org>>. Visitado em 17 de Maio de 2007.
- [15] RISE, Grupo de Reutilização em Engenharia de Software (Cin-UFPE). Disponível em <<http://www.rise.com.br/>>. Visitado em 17 de Maio de 2007.
- [16] SELECT BUSINESS SOLUTIONS, Select Component Manager. Disponível em <<http://www.selectbs.com>>. Visitado em 07 de Maio de 2007.
- [17] SOFTEX, Perspectivas de desenvolvimento e uso de componentes na Indústria Brasileira de Software e Serviços. Disponível em <[http://golden.softex.br/portal/softexweb/uploadDocuments/Componentes\(3\).pdf](http://golden.softex.br/portal/softexweb/uploadDocuments/Componentes(3).pdf)>. Visitado em 28 de Abril de 2007.
- [18] STOJANOVIC, Z., DAHANAYAKE, A., Service-Oriented Software System Engineering: Challenges and Practices. Idea Group Publishing, 2005.
- [19] SZYPERSKI, C., GRUNTZ, D., MURER, S., "Component Software – Beyond Object-Oriented Programming", 2a. edição, Addison-Wesley e ACM Press, 2002.
- [20] WERNER, C.M.L., BRAGA, R.M.M., "Desenvolvimento baseado em Componentes". In: Anais do Simpósio Brasileiro de Engenharia de Software, Minicursos, João Pessoa, PB, Brasil, outubro, 2000, pp. 297-329.