

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

PAULA MONTEAVARO FRANCESCHINI

**Estendendo ReUse: Rumo a uma
Ferramenta Mais Efetiva de Suporte a
Reuso**

Trabalho de Graduação.

Prof. Dr. Marcelo Soares Pimenta
Orientador

Porto Alegre, dezembro de 2008.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitor de Graduação: Prof. Carlos Alexandre Netto

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do CIC: Prof. Raul Fernando Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*"There are only two ways to live your life.
One is as though nothing is a miracle.
The other is as though everything is a miracle."*
Albert Einstein

AGRADECIMENTOS

Agradeço aos meus pais, Carlos e Ana, e a minha irmã, Aline, por estarem sempre ao meu lado, pelo amor, por serem o meu exemplo, por toda motivação e todo apoio.

Aos meus familiares, que tanto me apoiaram e incentivaram durante esses anos de faculdade.

Ao meu orientador, Marcelo Pimenta, por me orientar na execução desse trabalho, por todos os conselhos e pela confiança depositada.

Aos colegas da Mercador, por tudo que eu aprendi e continuo aprendendo nesses anos e por me proporcionarem um ótimo ambiente de trabalho.

Aos amigos e colegas, que me acompanharam nesses cinco anos de faculdade e fizeram meus dias mais divertidos. Especialmente ao Felipe, que revisou cada vírgula desse texto e ao Marcelo, por toda a ajuda (tecnológica ou não).

Por fim, agradeço a todos os que, de certa forma, fizeram parte dessa etapa da minha vida.

Muito Obrigada!

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	8
LISTA DE TABELAS	10
RESUMO.....	11
ABSTRACT.....	12
1 INTRODUÇÃO	13
2 REUSO DE SOFTWARE.....	15
2.1 Reuso de Software: Conceito	15
2.2 Benefícios na Prática do Reuso de Software	15
2.3 Obstáculos na Prática do Reuso de Software.....	16
2.4 Prática do Reuso na Engenharia Software.....	17
2.4.1 Reuso Arquitetado.....	17
2.4.2 Reuso de Padrões	17
2.4.3 Reuso de Frameworks	18
2.4.4 Reuso de Artefatos	18
2.4.5 Reuso de Módulos	18
2.4.6 Reuso de Modelos	19
2.4.7 Reuso de Código	19
2.4.8 Reuso de Assets.....	19
3 PROCESSO DE DESENVOLVIMENTO DE IU E A	
FERRAMENTA REUSE	21
3.1 Casos de Uso: Fundamentos e Elementos.....	21
3.1.1 Casos de Uso: Conceito	21
3.1.2 Atores.....	22
3.1.3 Cenários	22
3.1.4 Relacionamentos	22
3.2 Processo de Desenvolvimento Dirigido a Casos de Uso	24
3.3 Projeto e Modelo de Especificação de Interface com o Usuário	24
3.3.1 Projeto de IU no Processo de Desenvolvimento Dirigido por Casos de Uso	24
3.3.2 Modelos de Especificação de Interface com o Usuário.....	25
3.3.3 CRF: Um Modelo de Referência Para IU.....	26

3.4	Abordagem Proposta	27
3.5	A Ferramenta ReUse.....	28
3.5.1	Artefato Casos de Uso	29
3.5.2	Artefato de Especificação	30
3.5.3	Artefato de Implementação.....	30
3.5.4	Limitações da Ferramenta ReUse	31
4	NOVAS FUNCIONALIDADES PARA A FERRAMENTA REUSE.....	32
4.1	Glossário.....	33
4.2	Mecanismos de Busca.....	33
4.2.1	Remodelagem do Banco de Dados.....	33
4.2.2	Campo de Palavras-Chave	35
4.2.3	Busca por Texto Livre	36
4.2.4	Uso dos Sinônimos do Glossário para Sugerir Palavras-Chave	37
4.2.5	Spreading Activation.....	38
4.3	Feedback da Utilização de Artefatos	40
4.3.1	Classificação de Comentários de Artefatos	40
4.3.2	Ranking de Artefatos.....	41
4.4	Web Services.....	42
4.5	Plug-in para o Eclipse	42
4.6	Controle de Versões.....	44
4.7	Notificação de Eventos	46
4.8	Ajuda sobre a Ferramenta	47
4.9	Exemplo de Inclusão de Artefato Utilizando as Novas Funcionalidades	48
4.9.1	Inserção de Termo no Glossário	48
4.9.2	Inserção de Artefato Utilizando o Campo de Palavras-Chave.....	49
4.9.3	Solicitação de Recebimento de Notificações.....	51
4.9.4	Adicionando Comentários ao Artefato	52
4.9.3	Utilizando o Controle de Versões	52
4.9.6	Busca Utilizando a Tela de Pesquisa Geral	54
4.9.7	Utilizando o Plug-In para o Eclipse	55
5	CONCLUSÕES E TRABALHOS FUTUROS	57
	REFERÊNCIAS.....	58

LISTA DE ABREVIATURAS E SIGLAS

IU	Interface com o Usuário
OMG	Object Group Management
UML	Unified Modeling Language
IO	Interaction Object
CIO	Concrete Interaction Object
AIO	Abstract Interaction Object
UI	User Interface
AUI	Abstract User Interface
SUI	Sketched User Interface
CUI	Concrete User Interface
FUI	Final User Interface
XUI	Executable User Interface
RUCP	Padrão de Caso de Uso Reificado
CRF	Cameleon Reference Framework
IDE	Integrated Development Environment
SVN	Subversion

LISTA DE FIGURAS

Figura 2.1: As três dimensões de assets reusáveis.....	20
Figura 3.1: Relacionamento de associação	23
Figura 3.2: Relacionamento de inclusão	23
Figura 3.3: Relacionamento de extensão	23
Figura 3.4: Relacionamento de generalização.....	23
Figura 3.5: Modelo de Requisito "eixo" e "raios"	24
Figura 3.6: Relação entre objetos CIO e AIO	25
Figura 3.7: O Cameleon Reference Framework Estendido.....	26
Figura 3.8: Tela de Cadastro de Artefatos Casos de Uso.....	29
Figura 4.1: Exemplo de Inclusão de Termo no Glossário.....	33
Figura 4.2: Modelo Entidade-Relacionamento do Estado Atual do Banco de Dados da Ferramenta, para Artefatos de Especificação	34
Figura 4.3: Diagrama Entidade-Relacionamento do Novo Modelo Proposto.....	35
Figura 4.4: Cadastro de Artefato de Especificação com o Campo de Palavras-Chave ..	36
Figura 4.5: Tela de Busca por Texto Livre	37
Figura 4.6: Sugestões de Palavras-Chave	37
Figura 4.7: Artefatos e Palavras-Chave Representados como uma Rede Associativa ...	38
Figura 4.8: Execução do Algoritmo de Spreading Activation	39
Figura 4.9: Aba de Inclusão de um Comentário na Ferramenta Reuse.....	41
Figura 4.10: Inserção de um Comentário do Tipo Avaliação	41
Figura 4.11: Resultado da Pesquisa de Artefatos de Especificação Ordenado por Nota	41
Figura 4.12: Ícone do Plug-in Reuse.....	43
Figura 4.13: Janela de Busca de Artefatos no Plug-In.....	43
Figura 4.14: Listagem de Artefatos no Plug-In	44
Figura 4.15: Janela para Salvar Artefato no Projeto.....	44
Figura 4.16: Opção de Adicionar uma Nova Versão ou Visualizar o Histórico de Versões de um Artefato	45
Figura 4.17: Tela para Inclusão de Nova Versão de um Artefato	45
Figura 4.18: Histórico de Versões de um Artefato	46
Figura 4.19: Configuração de Notificações de Eventos.....	47
Figura 4.20: Tela de Inserção de Termo no Glossário Contendo o Botão de Ajuda	47
Figura 4.21: Janela Contendo Ajuda Sobre a Inserção de Termo no Glossário.....	48
Figura 4.22: Inserção do termo <i>login</i> no glossário	49
Figura 4.23: Listagem de termos cadastrados no glossário.....	49
Figura 4.24: Exemplo de Tela de Login.....	50
Figura 4.25: Código fonte da Tela de Login	50
Figura 4.26: Solicitação de Sugestões no Cadastro do Artefato de Implementação “ <i>Login</i> ”	51

Figura 4.27: Cadastro do Artefato de Implementação “Login” com Todos os Campos da Aba Geral Preenchidos.....	51
Figura 4.28: Inserção de um Comentário do Tipo Sugestão ao Artefato “Login”.....	52
Figura 4.29: Lista de Comentários Adicionados ao Artefato.....	52
Figura 4.30: Interface da Nova Versão do Artefato “Login”.....	52
Figura 4.31: Código-fonte da Nova Versão do Artefato “Login”.....	53
Figura 4.32: Inserção de Nova Versão ao Artefato “Login”.....	53
Figura 4.33: Histórico de Versões do Artefato “Login”.....	53
Figura 4.34: Resultado da Comparação entre as Versões.....	54
Figura 4.35: Busca pelo Artefato “Login”.....	54
Figura 4.35: Busca pelo Artefato “Login” no Eclipse.....	55
Figura 4.37: Download do Artefato “Login”.....	55

LISTA DE TABELAS

Tabela 3.1: Exemplo de Caso de Uso	21
--	----

RESUMO

Apesar de amplamente reconhecido como uma boa prática, o reuso de software tem encontrado diversas dificuldades para se tornar amplamente utilizado. Entre as dificuldades encontradas está a falta de metodologias e de ferramentas que dêem apoio ao reuso.

O objetivo deste trabalho é estender a ferramenta de suporte ao reuso proposta por Jamile Martins em (MAR 2007) adicionando funcionalidades à ferramenta e tornando-a mais eficiente na sua utilização, aprimorando os métodos de cadastro e busca de artefatos.

As novas funcionalidades adicionadas nesse trabalho foram um glossário de termos, um mecanismo de feedback na utilização dos artefatos, um plug-in para o Eclipse, a possibilidade de receber notificações de eventos relacionados a artefatos e uma página de ajuda para cada módulo da ferramenta. Além disso, foram realizadas melhorias no sistema de busca e a integração da ferramenta com um sistema de controle de versões.

Palavras-Chave: reuso de software, engenharia de software, casos de uso, interfaces com o usuário.

Extending ReUse: Towards a More Effective Tool of Reuse Supports

ABSTRACT

Although being widely recognized as a good practice, software reuse has been facing several difficulties to become widely used. Among these difficulties there is the lack of methodologies and tools that can support reuse.

The purpose of this paper is to extend the reuse support tool proposed by Jamilie Martins in (MAR 2007), by adding functionalities to the tool and making its use more efficient, enhancing the inserting methods and the artifacts search.

The new functionalities added in this paper were a glossary of terms, a feedback mechanism in the use of artifacts, a plug-in for Eclipse, the possibility of receiving notifications of events related to artifacts and a page to help each mode of the tool. Besides, improvements were made to the search system and to the integration of the tool with a version control system.

Keywords: software reuse, software engineering, use case, user interface

1 INTRODUÇÃO

O reuso de software consiste no processo de criar sistemas de software a partir de softwares já existentes, ao invés de começar do zero (KRU 92).

Embora esta técnica seja vista, desde os primórdios da engenharia de software (MCI 68), como a mais provável “bala de prata” para solucionar os problemas do desenvolvimento de sistemas, ela ainda encontra dificuldades para ser adotada em toda sua capacidade (COX 92).

Uma aplicação efetiva de aspectos técnicos e não-técnicos é crucial para o sucesso da prática reuso de software. Os fatores não-técnicos incluem questões como educação, treinamento, incentivos e gerenciamento. Por outro lado, os aspectos técnicos compreendem, entre outras coisas, a criação de um ambiente de reuso de software que dê suporte aos desenvolvedores no processo de desenvolver softwares com e para o reuso (ALM 2007).

O reuso de software pode ocorrer em diversos momentos do ciclo de desenvolvimento e envolvendo diferentes elementos de software. O foco, neste trabalho, é o reuso de artefatos relacionados com a interface com o usuário (IU).

Em (MAR 2008), Jamile Martins propôs uma ferramenta que oferece suporte a metodologia de reuso de IU, baseada na proposta de Augusto Moreira (MOR 2007). O trabalho de Augusto propõe uma abordagem de reuso de artefatos de software – com foco nos artefatos relativos a interface com o usuário – dirigida por casos de uso. Nele, as possibilidades de reuso são analisadas sob a perspectiva do caso de uso ao longo de todo o seu ciclo de vida.

A ferramenta ReUse, proposta por Jamile (MAR 2008), possibilita o reuso em diversos níveis da aplicação, agrupados em três níveis essenciais: caso de uso, artefatos de especificação e artefatos de implementação.

O objetivo deste trabalho é realizar melhorias e adicionar novas funcionalidades à ferramenta, visando torná-la mais efetiva no suporte ao reuso. Entre as melhorias estão o aprimoramento do sistema de busca, o uso de um controle de versões para os artefatos, o desenvolvimento de um plug-in para o ambiente de desenvolvimento integrado (Integrated Development Environment - IDE) Eclipse, entre outras.

O trabalho está estruturado conforme descrito a seguir.

No capítulo 2, é apresentada uma discussão sobre a prática do reuso, as dificuldades encontradas na sua realização e as vantagens da utilização desta técnica.

No capítulo 3, são apresentados conceitos de casos de uso, o processo de desenvolvimento dirigido a casos de uso, além da abordagem proposta por Augusto

Moreira (MOR 2007) e o estado atual da ferramenta desenvolvida por Jamile Martins (MAR 2008).

No capítulo 4, é apresentado o protótipo da ferramenta contendo as melhorias e novas funcionalidades desenvolvidas. Além disso, para cada uma delas, são apresentados exemplos práticos de sua utilização.

No capítulo 5, são apresentadas as conclusões e sugestões para trabalhos futuros.

2 REUSO DE SOFTWARE

Neste capítulo são apresentados os principais conceitos de reuso de software, os tipos de reuso existentes e as vantagens e as dificuldades em utilizar essa técnica.

2.1 Reuso de Software: Conceito

O reuso de software pode ser definido como o uso de artefatos de software já existentes na construção de um novo software, em qualquer fase do seu ciclo de vida (YON 98).

No desenvolvimento de um sistema de software, é comum aparecerem diversos problemas similares, que já foram resolvidos diversas vezes por outros desenvolvedores. A prática do reuso consiste em não “reinventar a roda”, ou seja, não resolver problemas similares várias vezes.

O primeiro passo para se obter sucesso na prática do reuso é entender que existe mais de uma opção à disposição. É possível reusar códigos-fonte, componentes, artefatos, padrões e templates (AMB 2004).

É importante que o reuso seja planejado e considerado desde o início do projeto, pois, para ser reusável e reusado, o software deve ser projetado, documentado e implementado visando o reuso (TRA 88). Além disso, ele deve ser escrito e armazenado de forma que permita uma fácil compreensão, indexação e busca.

2.2 Benefícios na Prática do Reuso de Software

São diversos os benefícios trazidos pelo reuso. Entre eles, podemos citar (YON 98, ALM 2007):

- *Produtividade*: o programador precisa produzir menos artefatos, conseqüentemente consegue produzir mais em um intervalo de tempo menor. É comum observar um crescimento de 20 a 40% na produtividade.
- *Qualidade*: como componentes reusáveis geralmente são submetidos a rigorosos testes, eles se tornam praticamente livres de defeitos. Além disso, quanto mais um artefato é reutilizado, a tendência é que sejam encontrados menos erros nele.
- *Confiabilidade*: o uso de componentes bem testados aumenta a confiabilidade do software desenvolvido. Além disso, o uso de um componente em diversos sistemas aumenta a chance de que os erros sejam detectados e fortalece a confiança naquele componente.

- *Redução no tempo de desenvolvimento:* o reuso de artefatos prontos, como especificações, casos de uso ou componentes, diminui o custo e tempo de desenvolvimento de um sistema.
- *Documentação:* a reutilização de software diminui a quantidade de documentação a ser escrita, uma vez que apenas os novos componentes desenvolvidos precisam ser documentados.
- *Manutenção:* com o aumento da qualidade dos componentes, menos defeitos são encontrados, diminuindo o custo com a manutenção do sistema. O artigo (TRA 88) cita ganhos de até 90% na manutenção de novos sistemas desenvolvidos utilizando técnicas de reuso.
- *Tamanho das equipes:* dobrar o tamanho de uma equipe nem sempre resulta no dobro de produtividade, uma vez que assim podem surgir problemas de comunicação. Se muitos componentes forem reutilizados, sistemas de software podem ser desenvolvidos com equipes menores, aumentando a produtividade e melhorando a comunicação.

2.3 Obstáculos na Prática do Reuso de Software

Embora o reuso de software seja uma estratégia de grande potencial, fatores comportamentais, organizacionais, técnicos e econômicos dificultam que a técnica seja amplamente utilizada. Esses fatores são listados e discutidos abaixo (YON 98, ALM 2007).

- *Fatores Econômicos:* o reuso de software traz vantagens a longo prazo, mas, inicialmente, exige gastos. Estes gastos incluem o custo de fazer algo reusável, o custo de reusar e o custo de definir e implementar um processo de reuso. A adoção do reuso exige custos iniciais com infra-estrutura, metodologia, treinamento, ferramentas e arquivos.
- *Fatores Comportamentais:* a tentativa de adotar um programa de reuso de software pode falhar mais por falta de suporte administrativo e de interesse dos desenvolvedores do que por motivos técnicos. É necessário haver um comprometimento e disposição para realização do reuso de software visto que, inicialmente, a utilização da técnica exige gastos. Portanto, é necessário ter ciência destes gastos e da expectativa de futuros ganhos. Além disso, dificuldades motivacionais dos desenvolvedores também inibem a adoção do reuso. Como, por exemplo, a resistência em compartilhar software ou a síndrome do “não foi inventado aqui” – resistência em utilizar software que foi criado por outros.
- *Fatores Organizacionais:* a adoção do reuso exige uma mudança de infra-estrutura da organização, para incluir novos papéis organizacionais, com a responsabilidade de desenvolvimentos de padrões, manutenção de um repositório de software, e relatório de resultados. Além disso, a falta de incentivo dificulta que os desenvolvedores gastem tempo criando componentes reusáveis, pois o sucesso do desenvolvimento geralmente é avaliado pelo tempo gasto para entregar um projeto. Os maiores ganhos com o reuso de software ocorrem quando o reuso é planejado e organizado em todas as etapas do desenvolvimento.

- *Fatores Técnicos:* incluem a dificuldade em encontrar software para ser reutilizado, ou manter um repositório organizado e a dificuldade de adaptar softwares que não foram desenvolvidos visando o reuso.

2.4 Prática de Reuso na Engenharia de Software

A forma mais conhecida de reuso de software é o reuso de código. Porém existem diversas outras formas de reuso, inclusive em níveis mais abstratos. A seguir são apresentadas algumas categorias de reuso (AMB 2007).

2.4.1 Reuso Arquitetado

O reuso arquitetado consiste na identificação, desenvolvimento e suporte de artefatos reusáveis de larga-escala via uma arquitetura corporativa. A arquitetura corporativa pode definir componentes de domínio reusáveis, coleções de classes de domínio/negócio que funcionem juntas de forma a suportar um conjunto coeso de responsabilidades, ou domínios de serviço, agrupadas em um pacote coeso de uma coleção de serviços.

2.4.2 Reuso de Padrões

Na década de setenta, o arquiteto Christopher Alexander definiu padrões de projeto afirmando que “cada padrão descreve o problema do nosso ambiente e o cerne de sua solução, de tal forma que você possa usar essa solução mais de um milhão de vezes, sem nunca fazê-lo da mesma forma”. O conceito tornou-se popular e amplamente utilizado na área de Engenharia de Software, principalmente após a publicação do livro “Padrões de Projeto: Soluções reutilizáveis de software orientado a objetos” (GAM 2000). Em geral, um padrão possui quatro elementos essenciais:

- **Nome do padrão:** referência utilizada para descrever um problema de projeto, suas soluções e conseqüências em uma ou duas palavras.
- **Problema:** explica o problema e o seu contexto, descrevendo em que situação o padrão pode ser utilizado.
- **Soluções:** descreve os elementos que compõem o padrão de projeto, seus relacionamentos, suas responsabilidades e colaborações.
- **Conseqüências:** descreve os resultados e as vantagens e desvantagens da utilização do padrão.

O uso de padrões aumenta a manutenibilidade e a capacidade da aplicação se tornar melhor por estar usando abordagens em comum para problemas que são reconhecíveis por programadores experientes.

Com o reuso de padrões, o desenvolvedor não reutiliza código, mas sim os conceitos que estão por trás do código. Portanto, padrões não provêm uma solução imediata – o desenvolvedor ainda tem que aplicá-lo corretamente e implementar o código.

2.4.3 Reuso de Frameworks

Um *framework* pode ser definido como uma coleção de códigos-fonte, classes, funções, técnicas e metodologias que facilitam o desenvolvimento de novos softwares (MIN 2007).

Frameworks fornecem um bom ponto de partida para desenvolver uma solução para um problema de um domínio. Eles, freqüentemente, encapsulam uma lógica complexa que levaria muito mais tempo para ser desenvolvida desde o início.

Porém, há uma série de desvantagens na utilização de *frameworks*. A complexidade deles torna-os difíceis de serem dominados, exigindo que se gaste muito tempo no aprendizado. Além disso, os *frameworks* geralmente são de plataformas específicas, amarrando o desenvolvedor a um único fornecedor. E, também, *frameworks* raramente funcionam juntos, pois são comumente construídos sobre bases arquiteturais diferentes.

2.4.4 Reuso de Artefatos

O reuso de artefatos consiste em reutilizar artefatos de desenvolvimento previamente criados para iniciar rapidamente um novo projeto. Esses artefatos podem ser casos de uso, documentos-padrão, modelos de domínios específicos, procedimentos, recomendações ou até mesmo outras aplicações, como aplicações comerciais de “prateleira” (*commercial-off-the-shelf – COTS*).

Existem artefatos resultantes de um processo de análise, chamado reuso de análise, como especificações lógicas completas ou subconjuntos delas (diagramas, descrições de dados ou descrições lógicas de conhecimento). Além disso, existem os artefatos resultantes do processo de projeto, chamado reuso de projeto, como, por exemplo, decisões de projetos (ZIR 95).

O desenvolvedor pode optar por reutilizar o artefato tal como ele é ou utilizá-lo como exemplo para como dar continuidade ao projeto. Além disso, o reuso de artefatos é fácil de ser realizado, pois, com freqüência, as pessoas podem comprar ou baixar da internet muitos artefatos.

2.4.5 Reuso de Módulos

Módulos podem ser definidos como componentes, serviços ou bibliotecas de código. Os módulos são auto-suficientes e encapsulam somente um conceito.

O reuso de módulos consiste em utilizar módulos pré-construídos e completamente encapsulados para o desenvolvimento de aplicações.

A diferença entre o reuso de módulos e o reuso de código é que no reuso de módulos não se tem acesso ao código fonte. Além disso, o reuso de módulos oferece um escopo maior de reusabilidade porque basta “conectar” os módulos para eles estarem funcionando.

É comum que o desenvolvedor não possa modificar um módulo, embora esta situação esteja mudando com o crescimento de popularidade da programação orientada a aspectos (POA). Além disso, às vezes, as cláusulas de licenciamento o tornam difícil de ser utilizado.

2.4.6 Reuso de Modelos

O reuso de modelos consiste na prática de utilizar um conjunto comum de layouts para artefatos-chave de desenvolvimento - como documentos, modelos e códigos-fonte - na organização. Basicamente, o reuso de modelos é o reuso de documentação.

A vantagem do reuso de modelos é que ele aumenta a consistência e a qualidade dos artefatos de desenvolvimento. Porém, as pessoas têm uma tendência a modificar modelos para seu próprio uso e não compartilhar essas alterações com a equipe.

2.4.7 Reuso de Código

A prática de reuso de código é a prática de reuso mais utilizada pelos desenvolvedores. Ela consiste no reaproveitamento de partes de código fonte.

O reuso de código pode acontecer, da melhor forma, com o compartilhamento de classes, coleções de funções e rotinas comuns. No pior caso, o reuso de código é feito copiando e depois modificando código existente, o que pode causar grandes problemas de manutenção.

Esta técnica de reuso reduz esforços do desenvolvimento, pois diminui a quantidade de código real que um programador precisa escrever. Porém, do ponto de vista de um projeto, códigos genéricos/reusáveis são mais complexos do que códigos feitos para um propósito único, tornando-os mais difíceis de serem desenvolvidos e mantidos.

2.4.8 Reuso de Assets

Um *asset*, de acordo com o Object Group Managment (OMG), é uma coleção de artefatos que fornecem uma solução para um problema em um determinado contexto (OMG 2005).

Um artefato pode ser qualquer produto do ciclo do desenvolvimento de um software, tais como documentos de requisitos, modelos, código-fonte, casos de teste, etc.

Um *asset* pode ter pontos de variabilidade que permitem aos usuários customizá-los, através da configuração de diversos parâmetros. Um ponto de variabilidade é uma localização no *asset* que pode ter o valor atribuído ou customizado pelo consumidor do *asset*. Além disso, o *asset* possui regras para uso, que são instruções que descrevem como o *asset* deve ser utilizado. Essas regras diminuem o tempo que os desenvolvedores levam para descobrir, analisar, consumir e testar o *asset*. Um *asset* pode ser um serviço, um *pattern*, um componente ou outro elemento que forneça uma solução.

Existem três dimensões que descrevem *assets* reusáveis: granularidade, variabilidade e articulação, como pode ser visto na figura 2.1.

- Granularidade: descreve quantos problemas particulares ou soluções alternativas cobertas pelo *asset*. Um *asset* mais simples oferece apenas uma solução para um único problema bem definido. Conforme aumenta a granularidade do *asset*, o *asset* resolve múltiplos problemas e/ou pode oferecer diversas soluções a esses problemas. Geralmente, com o aumento da granularidade há também o aumento do tamanho e da complexidade de um *asset*.
- Variabilidade: indica o quão variável um *asset* pode ser. Em um extremo do espectro, o *asset* é invariável, ou seja, não pode ser alterado. Geralmente este

é o caso de *assets* que são componentes binários. Esse tipo de *asset* é chamado de *asset* caixa-preta (*black-box asset*), uma vez que o seu conteúdo interno não pode ser alterado. No outro extremo há os *assets* caixa-branca (*white-box assets*), que são *assets* que são criados com a expectativa de que os consumidores irão editar e alterar a sua implementação. Existe ainda duas variações que estão no meio termo: os *assets* caixa-clara (*clear-box assets*) e os *assets* caixa-cinza (*grey-box assets*). Os *assets* caixa-clara expõem os detalhes de implementação (via modelos, fragmentos de códigos ou documentação), porém não podem ser modificados. Os detalhes são expostos apenas para ajudar o consumidor a entender melhor o funcionamento do *asset*. Os *assets* caixa-cinza expõe e permitem modificações apenas em um subconjunto de artefatos do *asset*, geralmente através dos parâmetros do *asset*.

- **Articulação:** descreve o grau de completeza dos artefatos em fornecer a solução. Os *assets* que especificam uma solução mas não a fornecem tem um baixo grau de articulação. Por outro lado, os *assets* que especificam e implementam a solução juntamente com documentos de suporte – requisitos, casos de uso, artefatos de teste, etc. – possuem um grau de articulação maior.

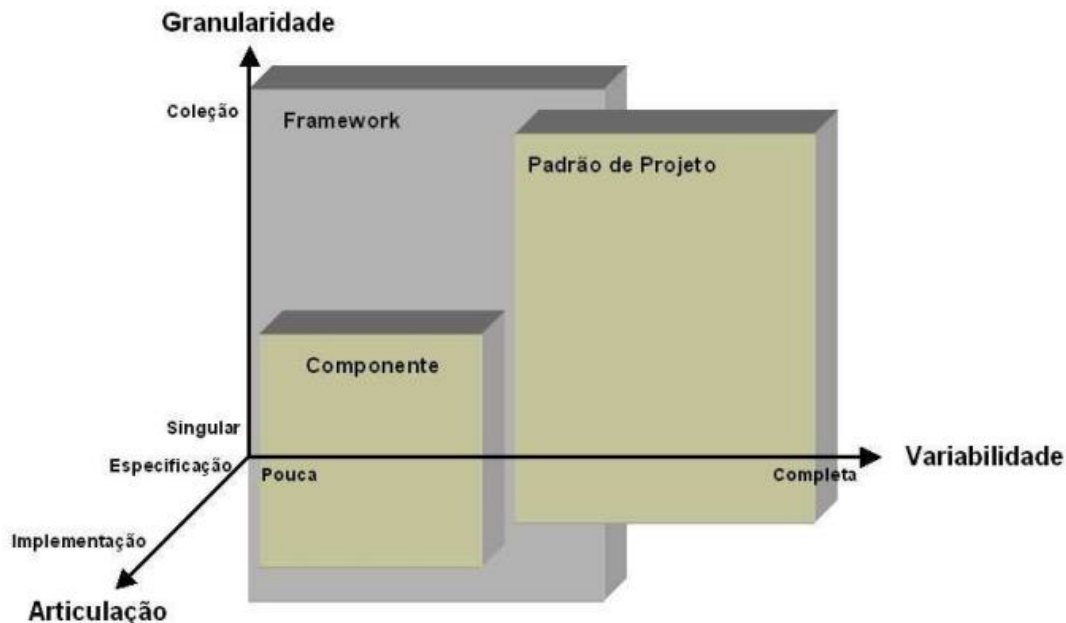


Figura 2.1: As três dimensões de assets reusáveis.

Para que um *asset* seja disponibilizado para reuso ele deve ser documentado e disponibilizado em um repositório. Para documentar, empacotar e catalogar os diversos *assets*, a OMG propôs a especificação RAS (*Reusable Asset Specification*). As quatro seções básicas de um *asset* são: Classificação (facilita a pesquisa e navegação por tópicos), Solução (coleção de artefatos que provêm a solução), Uso (instruções de como utilizar e aplicar o *asset*) e *Assets* Relacionados (define o relacionamento do *asset* com outros *assets*).

3 PROCESSO DE DESENVOLVIMENTO DE IU E A FERRAMENTA REUSE

O foco deste capítulo é revisar os conceitos associados a casos de uso (atores, cenários, relacionamentos, etc.) e as características de um processo de desenvolvimento dirigido a casos de uso.

3.1 Casos de Uso: Fundamentos e Elementos

Nesta seção serão revistos os conceitos de Casos de Uso e dos elementos que os compõem.

3.1.1 Casos de Uso: Conceito

Segundo a UML (OMG 2008), um caso de uso é uma maneira de especificar as funcionalidades de um sistema. Tipicamente eles são usados para capturar os requisitos do sistema, ou seja, o que um sistema deve fazer. Conceitos importantes relacionados ao caso de uso são os conceitos de *ator* e *cenário*, descritos nas próximas seções.

Casos de uso são fundamentalmente uma forma textual, embora possam ser escritos usando fluxogramas, diagramas de seqüências, redes de Petri ou linguagens de programação (COC 2005).

O principal objetivo de um caso de uso é especificar o que um sistema deve fazer, sem especificar como ele faz. Ou seja, o caso de uso explica o comportamento pretendido de um sistema, mas não explica detalhes de sua implementação.

Um exemplo de caso de uso pode ser visto na tabela 3.1.

Caso de uso: Realizar Cadastro Atores: Visitante, Administrador Finalidade: Realizar cadastro no sistema Seqüência de eventos	
<i>Ação do ator</i>	<i>Resposta do sistema</i>
1. O caso de uso inicia quando o visitante seleciona a opção de cadastrar-se ao sistema, ou quando o administrador seleciona a opção de cadastrar usuários ou administradores ao sistema.	2. O sistema exibe uma tela de cadastro, com os campos de dados a serem preenchidos.

<p>3. O ator preenche os campos com os dados e submete o cadastro ao sistema.</p> <p>5.O usuário, após receber o e-mail, volta ao sistema e digita o identificador recebido.</p>	<p>4. O sistema verifica se todos os campos obrigatórios foram preenchidos. Em caso positivo, envia um e-mail para o usuário cadastrado, com um identificador, para que seja confirmado o cadastro. Em caso negativo, exibe uma mensagem de erro.</p> <p>6.O sistema atualiza o banco de dados, adicionando o novo usuário, e exibe uma mensagem indicando a confirmação ou não do cadastro.</p>
--	--

Tabela 3.1 : Exemplo de Caso de Uso

3.1.2 Atores

Na UML, um ator é uma entidade externa ao sistema que, de alguma maneira, participa da história do caso de uso. Os atores são representados pelo papel que eles desempenham no caso de uso. Um ator possui um objetivo, e a sua função é, tipicamente, estimular o sistema com eventos de entrada, ou receber eventos do sistema para atingir esse objetivo. (LAR 2004).

Um ator é representado pelo papel que ele desempenha em um sistema, ou seja, o nome do ator deve referenciar o seu papel, ao invés de quem o representa.

3.1.3 Cenários

Um cenário, também chamado de instância de caso de uso, representa uma das diversas maneiras que um caso de uso pode ser realizado. Cada cenário descreve uma situação particular e um fluxo de ações específico percorrido pelo ator, na tentativa de atingir um objetivo.

Um caso de uso pode ser considerado um conjunto de cenários relacionados de sucesso e fracasso, que descrevem um ator usando um sistema como meio para atingir um objetivo.

Todo caso de uso deve conter um *fluxo principal*, que descreve o que acontece normalmente quando o caso de uso é realizado. Além disso, podem existir *fluxos alternativos*, que indicam alternativas ao fluxo principal, mas atingem o mesmo objetivo e *fluxos de exceção*, que descrevem a sequência de ações que ocorrem quando acontece algo inesperado na interação entre o ator e o sistema.

3.1.4 Relacionamentos

Casos de uso podem se relacionar entre si. A UML descreve três tipos de relacionamento entre casos de uso: inclusão, extensão e generalização. Além disso, casos de uso podem se relacionar com atores, e esse relacionamento é chamado de associação. Todos esses tipos de relacionamento são descritos abaixo.

- Associação: apresenta os atores que interagem com o caso de uso. A representação gráfica de uma associação é uma linha sólida ligando o ator e o caso de uso, como pode ser visto na figura 3.1

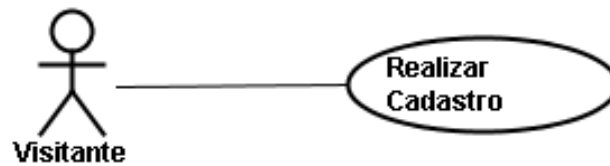


Figura 3.1: Relacionamento de Associação.

- Inclusão: ocorre quando um caso de uso contém o comportamento definido em outro caso de uso. Esse relacionamento estabelece uma relação de dependência entre o caso de uso base (aquele que inclui) e o caso de uso incluído, tornando a execução do caso de uso incluído obrigatória. O relacionamento de inclusão é representado como na figura 3.2.

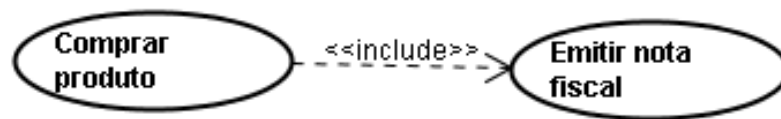


Figura 3.2: Relacionamento de Inclusão.

- Extensão: permite que o comportamento de um caso de uso seja estendido com comportamentos adicionais de outros casos de uso. O caso de uso base (aquele que está sendo estendido) pode conter vários pontos de extensão. Dependendo de certas condições, o caso de uso que fornece a extensão pode ser executado. A representação do relacionamento de extensão é como mostrado na figura 3.3.

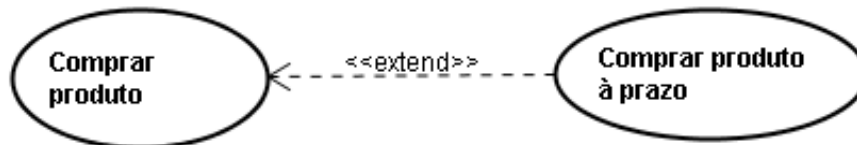


Figura 3.3: Relacionamento de Extensão

- Generalização: é um relacionamento entre casos de uso pai e casos de uso filhos, onde cada caso de uso filho representa uma forma mais especializada do comportamento genericamente definido no caso de uso pai. A representação gráfica do relacionamento de generalização pode ser vista na figura 3.4.

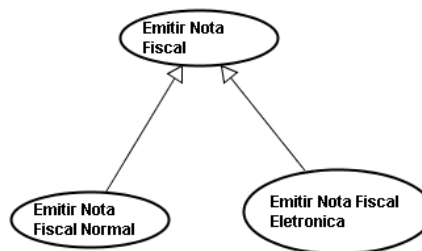


Figura 3.4: Relacionamento de Generalização.

3.2 Processo de Desenvolvimento Dirigido a Casos de Uso

O desenvolvimento iterativo e evolutivo – em contraposição ao ciclo de vida seqüencial ou em cascata – se baseia no aumento e no refinamento sucessivo de um sistema através de múltiplos ciclos de desenvolvimento (LAR 2004).

Baseado nesse modelo, Ivar Jacobson propôs a organização dos ciclos por caso de uso, dando origem ao termo **desenvolvimento dirigido a casos de uso** (*use case driven development*) (JAC 92).

A proposta de Jacobson consiste em atribuir um subconjunto de casos de uso, ou de versões simplificadas de caso de uso, a cada ciclo, para ser desenvolvido. Os casos de uso são geralmente agrupados considerando a prioridade para o cliente e o risco de desenvolvimento.

Os casos de uso ajudam a estruturar informações de planejamento do projeto, tais como data de entrega, equipes, prioridades, *status* do desenvolvimento, etc. Além disso, eles exercem uma especial influência nas especificações de projeto de interface com o usuário (IU), de testes de sistema e de documentação de usuário. Embora não estejam nos casos de uso, esses requisitos são conectados a eles. Segundo Cockburn (COC 2005), “os casos de uso agem como eixo de uma roda, e as outras informações agem como raios levando a diferentes direções”, como representado na figura 3.5.

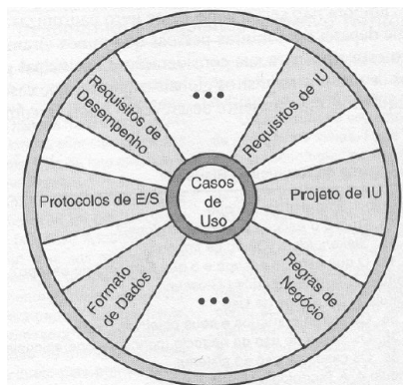


Figura 3.5 Modelo de Requisito “eixo” e “raios”

Fonte: COC 2005

3.3 Projeto e Modelos de Especificação de Interface com o Usuário (IU)

O projeto de interface com o usuário no processo de desenvolvimento dirigido por casos de uso e os principais instrumentos utilizados para modelagem das interfaces são apresentados a seguir.

3.3.1 Projeto IU no Processo de Desenvolvimento Dirigido por Casos de Uso

Em um processo de desenvolvimento dirigido por casos de uso, é a partir das descrições dos casos de uso que se desenvolve o projeto da IU. Utilizando as descrições dos casos de uso, constrói-se um protótipo da interface com o usuário. Este protótipo permite que sejam validados os requisitos funcionais especificados nos casos de uso e os passos para que o objetivo do caso de uso seja alcançado.

A construção de um protótipo depende da definição do ambiente de interação, ou seja, da definição dos elementos de interação que serão utilizados e o contexto no qual se dará a interação. A tecnologia de implementação não precisa, necessariamente, ter sido definida. Após a construção do protótipo, ele é apresentado e avaliado, para então ser corrigido e refinado continuamente até que atinja o nível de aceitação exigido pelos usuários.

Os protótipos e as especificações das IUs constituem um conjunto de artefatos que definem a forma visual e perceptível da descrição dos fluxos dos casos de uso essenciais. Esse conjunto é chamado *casos de uso apresentados*.

Após a definição e validação dos casos de uso apresentados, é iniciada uma nova fase chamada *realização de casos de uso*, em que são traduzidos os elementos da IU em termos de objetos e classes e é especificado como esses objetos colaboram entre si para a realização do caso de uso.

3.3.2 Modelos de Especificação de Interface com o Usuário

Um modelo de interface com o usuário pode ser definido como uma descrição formal, declarativa e livre de implementação da interface (EIS2000).

Uma interface com o usuário pode ser descrita por um conjunto de objetos de interação (*interaction objects - IOs*). Um IO é qualquer elemento que permita ao usuário de uma aplicação visualizar ou manipular informações, ou realizar uma tarefa interativa.

Segundo Vanderdonckt e Bodart (VAN 93), existem dois tipos de objetos de interação: concreto (*concrete interaction object - CIO*) e abstrato (*abstract interaction object - AIO*).

Um CIO, também conhecido como *widget (window gadget)* ou *controle*, é definido como qualquer entidade de IU que um usuário pode perceber (como texto, som, imagem) e/ou manipular (como botões ou menus). CIOs são objetos gráficos para captar e apresentar os dados relacionados à tarefas interativas do usuário. Eles contêm as características do ambiente de interação (por exemplo, Windows XP, Linux, etc.) e as restrições que devem ser respeitadas quando o usuário manipula o objeto.

O AIO consiste em uma visão abstrata do CIO, onde as características físicas são independentes do ambiente de interação a ser utilizado. Um AIO não possui uma aparência gráfica e é conectado a 0, 1 ou muitos CIOs em diferentes ambientes de interação.

A relação entre os objetos CIO e AIO é como mostrada na figura 3.6.

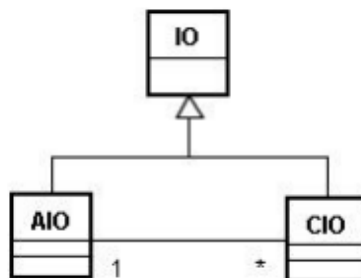


Figura 3.6: Relação entre objetos CIO e AIO.

Os IOs são compostos por três tipos distintos: os que representam espaços de interação, os que representam os elementos primitivos de interação e os que representam mecanismos de interação. Esses elementos de interação podem ser combinados de forma a implementar interações mais complexas. Por exemplo, o conjunto de elementos de interação de caixa de texto, calendário e botão compõe, quando organizados juntos na interface, um mecanismo de interação para o preenchimento de uma data pelo usuário.

3.3.3 CRF: Um Modelo de Referência para IU

Uma IU pode ser descrita através de hierarquias de objetos de interação e da forma como eles se comportam, como foi visto anteriormente. O CRF – *Cameleon Reference Framework* propõe decompor uma IU em artefatos de vários níveis de abstração, que se complementam e se relacionam entre si, formando uma hierarquia que os organiza, a qual chamamos de *árvore de reificação de IU*. O modelo CRF, estendido por Augusto em (MOR 2007), possui seis passos, como mostrado na figura 3.7, que são descritos a seguir.

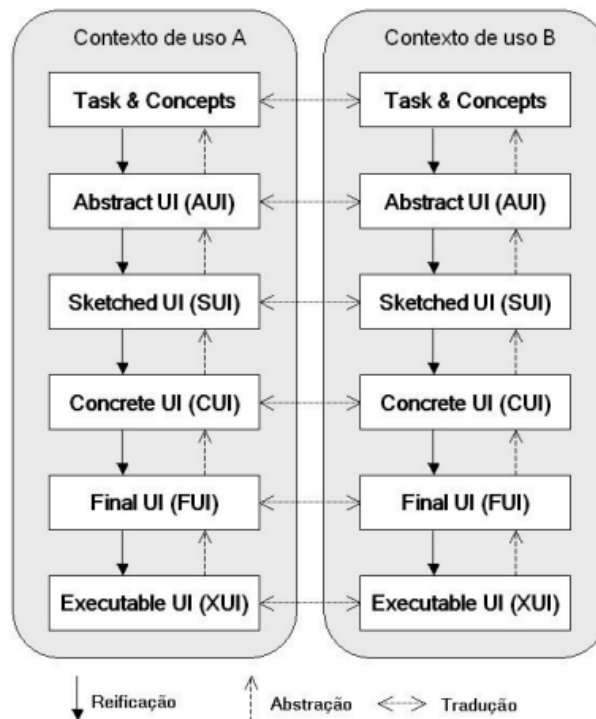


Figura 3.7: O Cameleon Reference Framework Estendido

- *Task and Concepts*: é o nível mais alto, que descreve as tarefas a serem realizadas e os conceitos orientados ao domínio requeridos para o desempenho dessas tarefas. As tarefas e conceitos são obtidos dos padrões de casos de uso e padrões de interação.
- *Abstract UI (AUI)*: é uma expressão canônica da renderização dos conceitos de domínio de funções de forma que seja independente dos objetos de interação disponíveis no ambiente de interação. É considerada uma abstração

de uma CUI, e consiste, basicamente, em uma descrição da interface definida no nível de *Tasks and Concepts*.

- *Sketched UI (SUI)*: define um conjunto de artefatos com o esboço e uma descrição textual da IU sendo modelada. Deve existir apenas uma SUI por modelo de interface. Todos os demais artefatos são formalizações e detalhamentos da IU descrita na SUI.
- *Concrete UI (CUI)*: composta de CIOs, é uma representação abstrata de uma FUI, de forma que seja independente de qualquer plataforma computacional e linguagem de programação. Os artefatos que compõem uma CUI são arquivos em XML escritos em uma Linguagem de Descrição de IU (UIDL).
- *Final UI (FUI)*: consiste em uma IU descrita em uma linguagem de programação, ou seja, são arquivos com os códigos-fonte de linguagens que implementam a IU. Uma FUI pode incluir outras FUIs e invocar chamadas a XUIs.
- *Executable UI (XUI)*: é um programa de computador executável em alguma plataforma, que renderiza a IU.

3.4 Abordagem Proposta

A abordagem proposta por Augusto Moreira (MOR 2007) contribui no sentido de identificar, produzir e consumir artefatos reusáveis, com ênfase no reuso de IHC, em um processo de desenvolvimento dirigido a casos de uso.

Esta abordagem foi inspirada no modelo de desenvolvimento de IUs para aplicações interativas multi-contexto, o Cameleon Reference Framework (CRF) (CAL 2003), apresentado na seção 3.3.3. A proposta do CRF é decompor uma IU em artefatos de vários níveis de abstração que se complementam e se relacionam entre si, formando uma hierarquia que os organiza em uma *árvore de reificação de IU*. Este modelo possibilita a organização de uma hierarquia de artefatos reusáveis que são identificados e (re)utilizados ao longo de todo o ciclo de desenvolvimento da IU.

O reuso deve acontecer através da especificação de modelos de reificação de padrões de interação e padrões de casos de uso. Ambos os padrões utilizam árvores de reificação de IUs, as quais contêm uma hierarquia de artefatos reusáveis que implementam em diversos níveis de detalhe as soluções definidas pelos padrões. O foco da ferramenta ReUse, proposta por Jamile (MAR 2008), é os padrões de casos de uso.

Um caso de uso pode conter diversas IUs. Um padrão de caso de uso pode conter diversos níveis de reificação de IU, formando uma árvore de reificação de IUs que modelam e implementam a IU do padrão de caso de uso. O conjunto de artefatos que compõem a hierarquia juntamente com o padrão de caso de uso é chamado padrão de caso de uso reificado (RUCP).

No RUCP, uma SUI é o protótipo que constitui a apresentação do padrão de caso de uso. Os demais elementos constituem os diversos níveis de reificação desse protótipo. O padrão de caso de uso deve ser escrito independente de tecnologia, portanto, é possível associar a esse padrão hierarquias de IUs em diversos tipos de ambientes de interação e de tecnologias de implementação. Por exemplo, um único padrão de caso de uso pode conter SUIs e CUIs que o apresentem em um ambiente gráfico desktop, um ambiente *Web* de interação, um ambiente de um telefone celular, etc. E, para cada um desses

ambientes, podem existir FUIs e XUIs que os implementem em diferentes linguagens de programação e/ou arquiteturas.

A ferramenta ReUse proposta, descrita a seguir, se baseia nessa visão hierárquica dos artefatos que compõe um RUCP.

3.5 A Ferramenta ReUse

A ferramenta Reuse (MAR 2008) é uma ferramenta *web* de suporte ao reuso de software, desenvolvida em PHP (PHP 2008), sobre um *framework*, o phpWAFr (REZ 2008) e utilizando o aplicativo de dados MySQL (MYS 2007).

O objetivo da ferramenta é oferecer a catalogação e busca de diferentes tipos de artefatos, de modo a propiciar o reuso de IUs em diferentes níveis de abstração.

A organização dos artefatos na ferramenta é feita segmentando a hierarquia de artefatos reusáveis que compõe o RUCP em três grandes grupos, onde um ou mais passos do CRF foi designado para cada um desses três grupos. Esses três grupos são:

- Casos de Uso, ou Padrões de Casos de Uso: este grupo contém, essencialmente, os casos de uso ou padrões de casos de uso. Cada caso de uso relaciona-se com pelo menos um artefato de especificação.
- Artefatos de Especificação: são os artefatos referentes ao modelo da IU a que o caso de uso referencia. Este grupo é composto pelos passos AUIs, SUIs e CUIs do CFR. Uma AUI pode ter diversas SUIs assim como uma SUI pode conter várias CUIs.
- Artefatos de Implementação: este grupo se refere às implementações das IUs definidas nos passos anteriores. É composto pelos passos FUI e XUI do CRF. Um artefato do nível de especificação pode se relacionar com inúmeros artefatos do nível de implementação.

A estrutura da ferramenta, baseada na organização descrita acima, é dividida em três módulos: Artefatos de Caso de Uso, Artefatos de Especificação e Artefatos de Implementação. Cada módulo possui a opção de cadastro de artefatos e de busca de artefatos.

O cadastro de cada tipo de artefato é dividido em etapas, representadas na interface da ferramenta através de abas. A figura 3.8 mostra a tela de cadastro de um artefato de caso de uso. As etapas de cada módulo são descritas a seguir.

Caso de Uso
Criação

Salvar Fechar

Geral Associar Coleção Associar Artefato de Especificação Artefatos Relacionados Comentários

Pacote:

Nome:

Data Criação: 15/11/2008

Data Última Alteração:

Descrição:
800 caracteres restantes

Versão:

Pré-Condições:
300 caracteres restantes

Pós-Condições:
300 caracteres restantes

Atores:
200 caracteres restantes (7)

Autor: Paula

Upload de Arquivo

Caso de Uso: Procurar...

Figura 3.8: Tela de Cadastro de Artefatos Casos de Uso

3.5.1 Artefato Caso de Uso

O catálogo de artefatos de casos de uso, como apresentado anteriormente, engloba padrões de casos de uso e casos de uso. O catálogo deste tipo de artefato é dividido em cinco etapas, descritas a seguir.

- **Geral:** Contém as seguintes informações referentes ao artefato:
 - Pacote: referência ao pacote ao qual o caso de uso pertence.
 - Nome: nome do caso de uso.
 - Data de Criação: data em que o caso de uso foi criado.
 - Data da Última Alteração: data da última alteração realizada no artefato
 - Descrição: descrição do artefato.
 - Versão: versão do artefato.
 - Pré-Condições: pré-condições do caso de uso.
 - Pós-Condições: pós-condições do caso de uso.
 - Atores: atores do caso de uso.
 - Autor: usuário que incluiu o artefato no sistema.
 - Caso de Uso: arquivo contendo o caso de uso completo.
- **Associar Coleções:** coleção ao qual o artefato pertence. Pode referenciar uma plataforma, framework, projeto, etc.
- **Associar Artefato de Especificação:** possibilidade de associar artefatos de especificação ao artefato de caso de uso cadastrado.
- **Artefatos Relacionados:** exibe todos os artefatos relacionados com o caso de uso.
- **Comentários:** nesta etapa é possível adicionar comentários referentes ao artefato de caso de uso.

3.5.2 Artefato de Especificação

O catálogo dos artefatos de especificação é referente às etapas AUI, SUI e CUI do CRF e é dividido em 6 etapas.

- **Geral:** Contém as seguintes informações referentes ao artefato:
 - Nome: nome do artefato de especificação.
 - Data de Criação: data em que o artefato foi criado.
 - Data da Última Alteração: data da última alteração realizada no artefato
 - Descrição: descrição do artefato.
 - Versão: versão do artefato.
 - Tipo: tipo do artefato a ser inserido. Pode ser SUI, AUI ou CUI.
 - Autor: usuário que incluiu o artefato no sistema.
 - Artefato de Especificação: arquivo contendo o artefato. Pode ser, por exemplo, no caso de uma SUI, uma imagem.
- **Associar Coleções:** da mesma forma que no caso de uso, indica a coleção ao qual o artefato pertence.
- **Associar Artefato de Especificação:** possibilidade de associar o artefato de especificação a outros artefatos de especificação já cadastrados na ferramenta.
- **Associar Artefato de Implementação:** possibilidade de associar o artefato de especificação a outros artefatos de implementação já cadastrados na ferramenta.
- **Artefatos Relacionados:** exibe todos os artefatos relacionados com o caso de uso.
- **Comentários:** da mesma forma que nos casos de uso, nesta etapa é possível adicionar comentários referentes ao artefato de especificação.

3.5.3 Artefato de Implementação

O catálogo dos artefatos de especificação é referente às etapas FUI e XUI do CRF e é dividido em 4 etapas.

- **Geral:** Contém as seguintes informações referentes ao artefato:
 - Nome: nome do artefato de implementação.
 - Data de Criação: data em que o artefato foi criado.
 - Data da Última Alteração: data da última alteração realizada no artefato
 - Descrição: descrição do artefato.
 - Tipo: tipo do artefato a ser inserido. Pode ser FUI ou XUI.
 - Versão: versão do artefato.

- Linguagem de Programação: linguagem na qual o artefato foi implementado.
- Autor: usuário que incluiu o artefato no sistema.
- Caso de Uso: arquivo contendo o caso de uso completo.
- **Associar Coleções:** da mesma forma que nos artefatos anteriores, indica coleção ao qual o artefato pertence. Pode referenciar uma plataforma, framework, projeto, etc.
- **Artefatos Relacionados:** exibe todos os artefatos relacionados com o caso de uso.
- **Comentários:** nesta etapa é possível adicionar comentários referentes ao artefato de caso de uso.

3.5.4 Limitações da Ferramenta

A ferramenta ReUse é uma ótima solução de suporte ao reuso de software através da catalogação e busca de artefatos. Porém, possui ainda algumas limitações, apresentadas a seguir.

O sistema de busca adotado na ferramenta ReUse é bastante limitado. Não existe uma interface de busca na qual seja possível buscar um artefato de qualquer tipo. As buscas podem ser feitas apenas em cada módulo (casos de uso, especificação e implementação). Além disso, não é possível buscar apenas por palavras-chave de determinado artefato, sendo necessário, para realizar a busca, utilizar os mesmos campos do cadastro de artefatos. Segundo Krueger (KRU 92), “o tempo do processo de busca e adaptação de um artefato já existente deve ser menor do que o tempo de construção de um artefato novo”. Portanto, a ferramenta deve ajudar ao máximo o usuário a encontrar o artefato que deseja.

A ferramenta possui diversos usuários e diferentes pessoas tendem a referenciar um mesmo termo de diferentes maneiras, portanto, a ferramenta deve possuir mecanismos que possam auxiliar os usuários na comunicação. A aba de comentários já possibilita um meio de comunicação, porém pode ser aprimorada, para facilitar ainda mais. Além disso, não existe na ferramenta um modo de avaliar os artefatos e classificá-los de modo a saber quais são os melhores artefatos para serem escolhidos para o reuso.

Atualmente, na ferramenta ReUse, não é possível manter diversas versões de um mesmo artefato em um único cadastro. Para inserir uma nova versão de um artefato, é necessário realizar um novo cadastro ou apagar o arquivo do cadastro atual e inserir um novo, perdendo, assim, a versão anterior.

Outra limitação é a falta de integração com um ambiente de desenvolvimento. Essa integração permitira aos desenvolvedores buscarem artefatos sem sair do ambiente em que trabalham, facilitando ainda mais a adoção do reuso.

Por fim, a ferramenta poderia dispor de um sistema de ajuda, pois a sua utilização não é intuitiva para usuários iniciantes, e a falta de uma explicação sobre o sistema dificulta o uso.

4 NOVAS FUNCIONALIDADES PARA A FERRAMENTA REUSE

Neste capítulo são apresentadas as novas funcionalidades e melhorias implementadas na ferramenta ReUse, na seguinte ordem: um glossário de termos, com o objetivo de aprimorar a comunicação entre os usuários da ferramenta; melhorias no mecanismo de busca da ferramenta; classificação de comentários e ranking de artefatos; criação de *web services* para disponibilizar serviços da ferramenta; desenvolvimento de um *plug-in* para o eclipse, para facilitar o reuso no ambiente de desenvolvimento; integração da ferramenta com um sistema de controle de versões, para poder armazenar diversas funções de um mesmo artefato; notificação de eventos, para auxiliar no acompanhamento das versões e utilizações de um artefato e a criação de uma tela de ajuda para auxiliar o uso da ferramenta.

4.1 Glossário

Diferentes grupos de pessoas podem referenciar um mesmo termo de diferentes maneiras. Uma das formas de solucionar esse problema é a utilização de um glossário. Um glossário, ou dicionário de dados, lista e define todos os termos que requerem esclarecimentos, de modo a melhorar a comunicação e reduzir o risco de mal entendidos entre os autores e os utilizadores dos artefatos (LAR 2004).

Os atributos escolhidos para compor o glossário da ferramenta ReUse foram:

- *Termo*: termo cuja definição é considerada relevante;
- *Categoria*: categoria ao qual o termo pertence, por exemplo, nome de Caso de Uso, conceito, etc.
- *Definição*: definição do termo utilizando linguagem simples e compreensível;
- *Sinônimos*: outros termos considerados equivalentes ao termo em questão.
- *Observações*: comentários gerais.

Um exemplo da tela de inclusão de termos no glossário pode ser visto na figura 4.1.

Além da adição de termos no glossário, as outras operações possíveis são: a busca de termos (através de uma tela de busca), a remoção e a edição de termos.

Glossário
Criação

Geral

Termo:	Validação
Categoria:	
Definição:	Validar, verificar se está correto
Sinônimos:	legitimar; autenticar; consistir; homologar
Obs.: Separar os sinônimos usando ponto-e-vírgula (;).	
Observação:	

Figura 4.1: Exemplo de Inclusão de Termo no Glossário.

4.2 Mecanismos de Busca

No processo de classificação de artefatos, é necessária a utilização de metadados para descrever o artefato a ser inserido na ferramenta. Metadados podem ser definidos como “dados sobre dados”, ou seja, eles representam informações que caracterizam o artefato.

Ao inserir um artefato na ferramenta, o usuário inclui metadados sobre esse artefato. Da mesma forma, outro usuário, ao buscar um artefato, precisa escolher os termos que acredita serem adequados para identificar o artefato.

Porém, como mencionado anteriormente, as pessoas tendem a conceitualizar e nomear o mesmo objeto de formas muito diferentes. Os resultados de uma pesquisa em (HEN 94) indicam que a probabilidade de duas pessoas escolherem as mesmas palavras-chave para um objeto familiar é apenas de 10 a 15%. Mesmo limitando a escolha em 15 sinônimos, a concordância resulta em apenas 60 a 80%. Essa pesquisa aponta para uma potencial discordância entre os termos utilizados para cadastrar o artefato no sistema e os termos escolhidos pelo usuário ao buscar um artefato.

Na intenção de diminuir essa distância e aprimorar a busca por artefatos, algumas técnicas foram implementadas na ferramenta ReUse e são descritas a seguir.

4.2.1 Remodelagem do Banco de Dados

A estrutura atual do banco de dados da ferramenta ReUse é dividida em três módulos, um para cada tipo de artefato – casos de uso, especificação, implementação.

Atualmente, para cada um desses tipos de artefato existem três tabelas, uma contendo os dados do artefato, outra os comentários e outra as palavras-chave relacionadas ao artefato. A figura 4.2 mostra o diagrama Entidade-Relacionamento da estrutura atual do banco de dados para os artefatos do tipo especificação. Portanto, existem no banco de dados três modelos similares ao da figura 4.2: um para os artefatos do tipo caso de uso, outro para os de especificação e um último para artefatos do tipo

implementação. O editor utilizado para fazer a modelagem do banco de dados foi o MySql Workbench 5.0 (MYS 2008).

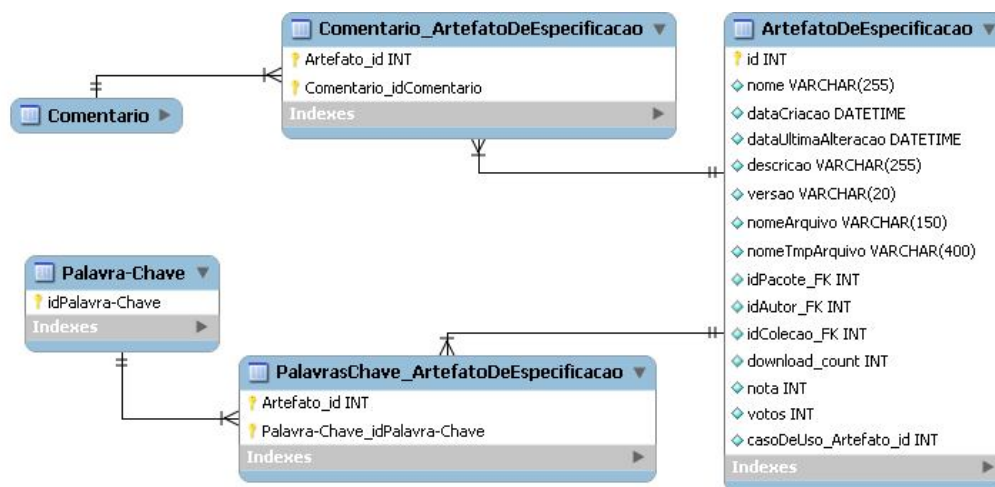


Figura 4.2: Modelo Entidade-Relacionamento do Estado Atual do Banco de Dados da Ferramenta, para Artefatos de Especificação.

Porém, as tabelas de comentários e artefatos são apenas tabelas que indicam o relacionamento entre um comentário ou palavra-chave e um artefato. São, portanto, iguais para todos os tipos de artefato. Além disso, as três tabelas contendo os dados do artefato são bastante similares, contendo poucos atributos diferentes, que são os atributos específicos de cada artefato.

Por exemplo, os atributos como nome, data da inclusão, data de modificação, versão, nota, etc. são iguais para as três tabelas. Porém, a tabela de artefato de implementação possui ainda um atributo chamado linguagem de programação, que não existe na tabela de artefatos de caso de uso nem de artefatos de especificação.

Essa separação das tabelas do banco de dados em módulos dificulta uma busca mais genérica, pois para buscar um artefato de qualquer tipo é necessário realizar a busca em três tabelas diferentes.

Buscando otimizar a estrutura do banco de dados e a forma como a busca de artefatos é realizada, foi feita uma remodelagem, de forma a utilizar apenas uma estrutura para todos os tipos de artefato.

O modelo proposto é mostrado na figura 4.3. Nele, foi criada uma tabela genérica, para todos os tipos de artefato, contendo os atributos comuns a todos os artefatos. Também foram criadas três tabelas específicas - caso de uso, especificação e implementação - cada uma contendo apenas os atributos específicos de cada tipo de artefato. Além disso, apenas uma tabela de comentário e uma de palavras-chave foram utilizadas para todos os tipos de artefato. Por fim, foi criada uma tabela com os tipos de artefato existentes e um atributo na tabela artefato que indica qual o tipo do artefato (caso de uso, especificação ou implementação).

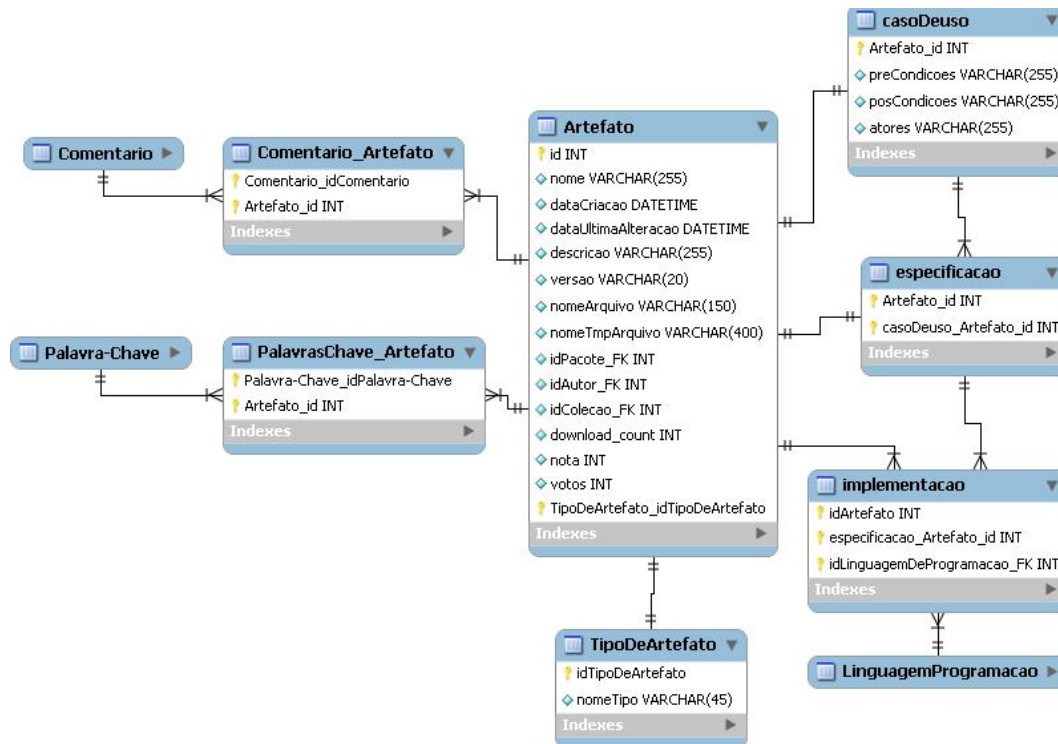


Figura 4.3: Diagrama Entidade-Relacionamento do Novo Modelo Proposto.

Este modelo melhora o desempenho e facilita a busca genérica de artefatos, pois todos os dados necessários para busca estão na tabela artefato. Os dados específicos de cada artefato serão buscados apenas na visualização ou edição do cadastro do artefato.

4.2.2 Campo de Palavras-Chave

A inclusão de qualquer artefato – casos de uso, especificação e implementação – exige que o usuário preencha diversos campos, para classificar o artefato.

Além dos campos já existentes, foi adicionado um campo de palavras-chave, como pode ser visto na figura 4.4. A inclusão do campo ajuda a adicionar mais informações sobre o componente. Nele o usuário pode inserir palavras que sejam relevantes para o artefato em questão, mas que não fazem parte do nome do artefato.

The image shows a web form for registering an artifact specification. It has four tabs at the top: 'Geral', 'Associar Coleção', 'Associar Artefato de Especificação', and 'Associar Artefato de Implementação'. The 'Geral' tab is active. The form contains the following fields and elements:

- Nome:** A text input field.
- Data Criação:** A date picker showing '28/10/2008'.
- Data Última Alteração:** A date picker.
- Descrição:** A large text area with a '200 caracteres restantes' (200 characters remaining) indicator.
- Palavras-Chave:** A text input field with a 'Sugestões...' button to its right.
- Versão:** A text input field.
- Tipo:** A text input field.
- Autor:** A text input field containing the name 'Paula'.
- Notification:** A checkbox with the text 'Quero ser notificado de eventos relacionados a este artefato.'
- Upload de Arquivo:** A section header for the file upload area.
- Artefato de Especificação:** A text input field with an 'Arquivo...' button to its right.

Figura 4.4: Cadastro de Artefato de Especificação com o Campo de Palavras-Chave

A inserção deste campo ajuda na busca dos artefatos, pois adiciona mais informação sobre eles, aumentando as chances de o artefato ser encontrado. Além disso, a adição deste campo colabora para algumas das próximas técnicas descritas.

4.2.3 Busca por Texto Livre

Com o objetivo de proporcionar uma busca mais flexível, criamos uma nova forma de busca, onde o usuário digita quaisquer palavras e o sistema retorna artefatos potencialmente relevantes.

A tela de busca livre é mostrada na figura 4.5, e serve como complemento para as formas de busca já existentes na ferramenta e específicas de cada artefato. O usuário insere no campo de busca termos que descrevem o artefato que ele está procurando e o sistema busca artefatos que contenham essas palavras em qualquer campo dos artefatos cadastrados. Os termos são pesquisados em todos os artefatos cadastrados no sistema – casos de uso, especificação e implementação – a menos que seja especificado um deles pelo usuário.

A busca por texto livre facilita a pesquisa, pois o usuário não precisa saber exatamente onde o termo que ele procura foi cadastrado - o termo pode estar, por exemplo, no nome do artefato, na palavra-chave ou na descrição.

Além disso, o sistema classifica os artefatos encontrados por uma ordem de relevância.

Buscar palavras:	<input type="text"/>
<input type="checkbox"/>	Casos de Uso
<input type="checkbox"/>	Artefatos de Especificação
<input type="checkbox"/>	Artefatos de Implementação

Figura 4.5: Tela de Busca por Texto Livre

4.2.4 Uso dos Sinônimos do Glossário para Sugerir Palavras-Chave

O uso dos sinônimos cadastrados nos termos do glossário pode ser de grande utilidade para a classificação de artefatos na ferramenta.

A sugestão de sinônimos da ferramenta ReUse utiliza dois atributos da ferramenta: os sinônimos cadastrados no glossário e o campo de palavras-chave. Ao cadastrar um artefato, o usuário tem a opção de solicitar ao sistema sugestões para o preenchimento do campo de palavras-chave.

As sugestões são baseadas no nome atribuído ao artefato pelo usuário e os sinônimos dos termos que compõe o nome. Por exemplo, ao cadastrar um artefato de especificação com o nome de “Validar de datas” e solicitar ao sistema sugestões para o campo de palavras-chave, o sistema irá sugerir os sinônimos cadastrados para os termos “validar” e “datas”, como pode ser visto na figura 4.6.

Sugestões de Keywords

Adicionar Fechar

Sugestões

<input type="checkbox"/>	Termos
<input type="checkbox"/>	autenticar
<input type="checkbox"/>	consistir
<input type="checkbox"/>	dia
<input type="checkbox"/>	homologar
<input type="checkbox"/>	legitimar
<input type="checkbox"/>	época

Figura 4.6: Sugestões de palavras-chave.

A sugestão de sinônimos ajuda no acréscimo de informação a respeito do artefato. Ao invés de o usuário precisar, sozinho, pensar em quais outras formas o artefato que ele está cadastrando pode ser nomeado, o sistema ajuda a indicar essas possíveis formas. Além disso, o usuário tem a opção de escolher apenas as sugestões que considera realmente relevantes para o seu artefato.

4.2.5 Spreading Activation

A maioria dos mecanismos de busca, que utilizam palavras-chave, retornam apenas componentes cujas palavras-chave são exatamente iguais as da busca. Porém, usuários podem utilizar diversas palavras diferentes para referenciar o mesmo item. Além disso, algumas vezes, pode não haver uma igualdade direta entre as palavras-chave de uma busca e as utilizadas nos componentes que são relevantes para a busca (ASW 2005).

O mecanismo de spreading activation torna a busca mais flexível, uma vez que retorna não apenas os componentes que combinam exatamente com a pesquisa, mas também componentes potencialmente relevantes. O algoritmo de spreading activation, utilizado pela ferramenta ReUse na busca por texto livre, foi adaptado do artigo (HEN 97) e é descrito a seguir.

A ferramenta ReUse utiliza sua estrutura para montar uma rede associativa, na qual os artefatos e as palavras-chave são os nodos e as arestas representam o relacionamento entre eles. Um exemplo pode ser visto na figura 4.7.

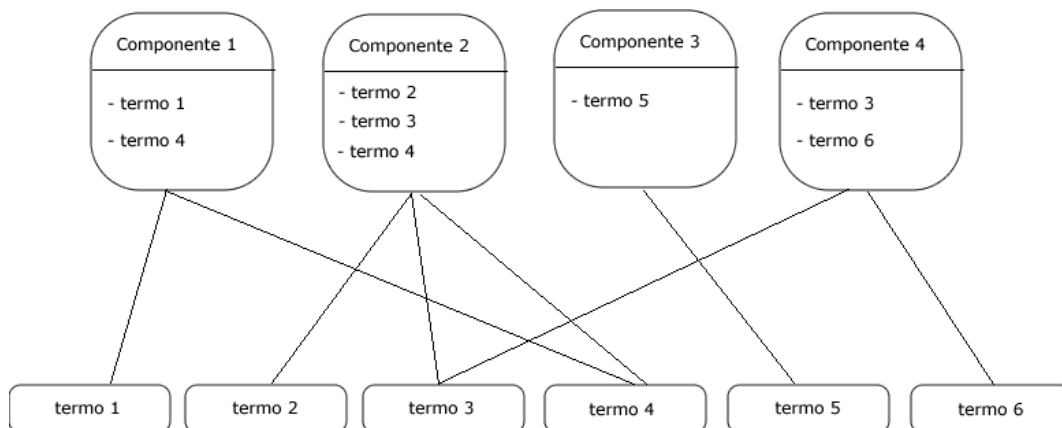


Figura 4.7: Artefatos e palavras-chave representados como uma rede associativa.

O processo de spreading activation inicia quando o usuário digita no campo de busca palavras pelas quais deseja pesquisar. O texto digitado na busca é dividido em termos, e todos os artefatos que possuem aquele termo como palavras-chave são ativados.

Se um nodo possui um valor de ativação maior do que zero, o valor é transmitido para cada um dos nodos ao qual ele está conectado. Caso contrário, nenhum valor é transmitido.

A cada ciclo, são ativados os artefatos referenciados pelas palavras-chave que já estão ativadas e suas respectivas palavras-chave.

Os artefatos ativados no primeiro ciclo recebem um valor de ativação igual a 100. A cada ciclo, esse valor de ativação é reduzido em 25%. Se um artefato que já foi ativado é novamente ativado por uma outra palavra-chave, ele soma ao seu valor de ativação o novo valor recebido. Caso o artefato ainda não tenha sido ativado, ele recebe o valor de ativação referente ao ciclo no qual foi ativado.

O algoritmo pode ser executado até os valores estabilizarem, ou até um número pré-determinado de ciclos ser alcançado. No caso da ferramenta ReUse, utilizamos 4 ciclos.

Um exemplo do funcionamento do algoritmo pode ser vista na figura 4.8, onde inicialmente foi selecionado pelo usuário a palavra-chave “termo 1”.

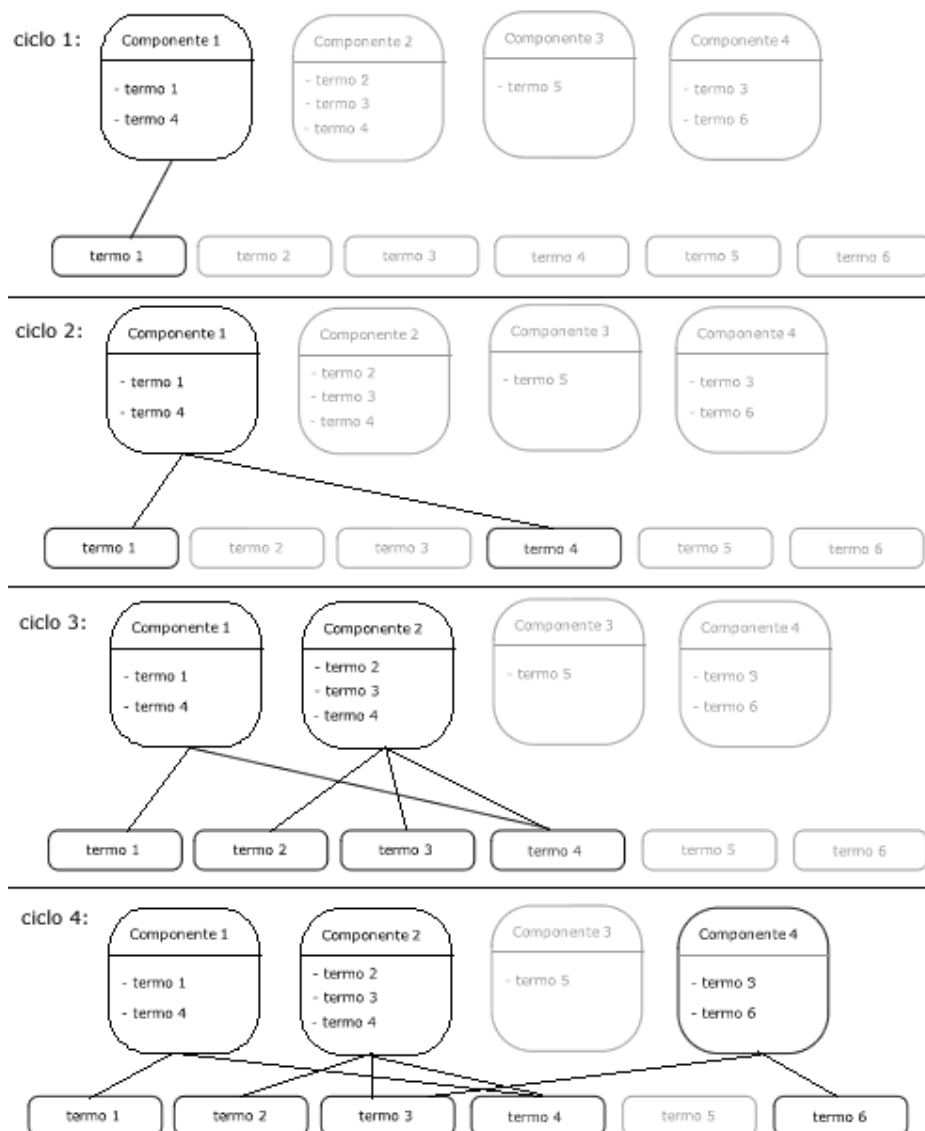


Figura 4.8: Execução do algoritmo de Spreading Activation

4.3 Feedback da Utilização de Artefatos

Os serviços de feedback permitem a identificação de artefatos de melhor qualidade, além de outras informações como “onde” e “como” o artefato está sendo usado, problemas encontrados na utilização, etc (ALM 2007).

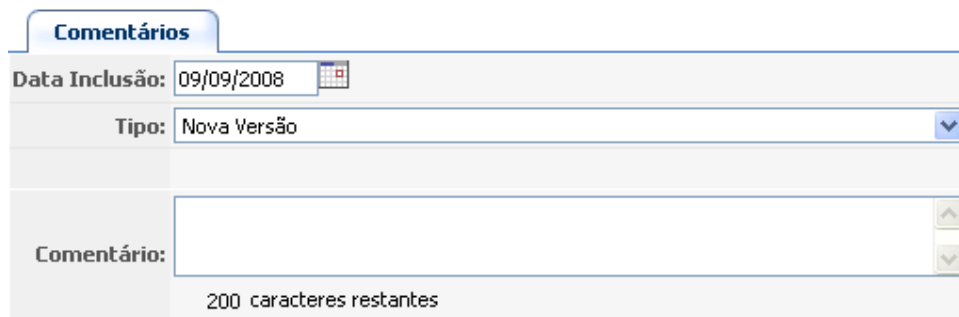
No cadastro de qualquer artefato na ferramenta ReUse há a aba Comentários, que tem por finalidade fornecer esse feedback a respeito do artefato. Quanto mais pessoas estiverem trabalhando e utilizando um artefato, mais necessário é aprimorar a comunicação entre essas pessoas, a fim de evitar conflitos e desentendimentos. Algumas melhorias foram implementadas na forma de cadastro dos comentários, e são apresentadas a seguir.

4.3.1 Classificação de Comentários de Artefatos

A classificação dos comentários em diversos tipos permite uma melhor identificação do assunto do comentário envolvendo o artefato. A inclusão de um comentário pode ser vista na figura 4.9.

Na ferramenta ReUse, um comentário pode ser classificado com os seguintes tipos:

- *Inserção*: Utilizado no momento de inserção de um artefato na ferramenta. Pode conter informações adicionais sobre o artefato, além das que já estão presentes nos outros campos.
- *Bug Encontrado*: Utilizado para reportar algum problema encontrado no artefato.
- *Nova Versão*: Selecionado quando o usuário está incluindo uma nova versão do artefato, para informar quais as novidades ou melhorias dessa nova versão.
- *Sugestão*: Utilizado para sugerir melhorias ou novas funcionalidades ao artefato.
- *Observação Geral*: Utilizado para inserir uma observação qualquer a respeito do artefato.
- *Avaliação*: Utilizado para avaliar o artefato. Ao ser selecionado esse tipo de comentário, o sistema permite que o usuário avalie o artefato com uma nota de 1 a 10. Além disso, há o campo de texto livre, para que o usuário insira uma breve justificativa a respeito da nota. Um exemplo de avaliação de artefato pode ser visto na figura 4.10.



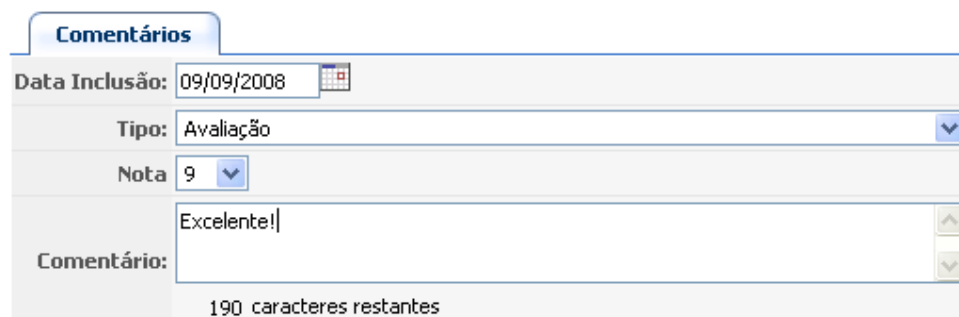
Comentários

Data Inclusão: 09/09/2008

Tipo: Nova Versão

Comentário:
 200 caracteres restantes

Figura 4.9: Aba de Inclusão de um Comentário na Ferramenta ReUse.



Comentários

Data Inclusão: 09/09/2008

Tipo: Avaliação

Nota: 9

Comentário: Excelente!
 190 caracteres restantes

Figura 4.10: Inserção de um Comentário do Tipo Avaliação.

4.3.2 Ranking de Artefatos

A avaliação dos artefatos ajuda a identificar os artefatos com maior qualidade para serem reusados, e os de menor qualidade, que podem, eventualmente, ser excluídos do repositório.

A ferramenta ReUse utiliza uma técnica simples para fazer um ranking dos artefatos. Depois de utilizar um artefato, o usuário pode avaliá-lo na aba de comentários, dando uma nota de 1 a 10. A ferramenta faz uma média de todas as notas dadas ao artefato e disponibiliza, na listagem, o ordenamento por nota, além de exibir o número de avaliações que levaram a essa nota. Desta forma, o usuário pode, ao buscar um artefato, ter uma idéia da qualidade dele antes de decidir utilizá-lo ou não.

Na figura 4.11, temos um exemplo dos artefatos ordenados por nota.

Resultado da Pesquisa

<div> Novo Artefato de Especificação Excluir </div>								
<input type="checkbox"/>	Ident.	Data Últ. Alteração	Nome	Tipo	Descrição	Nota	Avaliações	Autor
<input type="checkbox"/>	38	2008-10-28 00:00:00	Artefato 2	SUI	Artefato para teste 2	9	5	Paula
<input type="checkbox"/>	39	2008-10-28 00:00:00	Artefato 3	SUI	Artefato para teste 3	8	2	Paula
<input type="checkbox"/>	37	2008-10-28 00:00:00	Artefato 1	SUI	Artefato para teste 1	6	3	Paula

Figura 4.11: Resultado da Pesquisa de Artefato de Especificação ordenado por nota.

4.4 Web Services

Um *web service* é um serviço que fica disponível na Internet, utiliza um sistema de padronização XML e não é ‘amarrado’ a nenhum sistema ou linguagem de programação (CER 2002). Utilizando esta tecnologia é possível realizar a integração e a comunicação de diferentes aplicações, que podem ter sido desenvolvidas em diferentes plataformas.

Para criar os *web services* na ferramenta ReUse, foi utilizada a biblioteca nusoap (NUS 2007) disponível para a linguagem PHP. Os seguintes *web services* foram implementados:

1. **SearchArtifacts:** retorna uma lista com os dados dos artefatos que atendem aos critérios de busca. O serviço recebe como parâmetro palavras-chave, tipos de artefato e linguagens de programação. A ferramenta realiza a busca utilizando as palavras-chave recebidas e a técnica de *spreading activation*. São retornados apenas os artefatos do tipo selecionado pelo usuário – no caso de nenhum tipo selecionado, todos os tipos são considerados. Além disso, no caso de artefatos de implementação, se forem especificada alguma linguagem de programação, o serviço retorna apenas os artefatos que atendem a esse filtro também.
2. **DownloadArtifact:** recebe como parâmetro o identificador do artefato no banco de dados e retorna o endereço do artefato no servidor, para que possa ser realizado o download do mesmo.
3. **GetProgrammingLanguages:** sem receber nenhum parâmetro, retorna uma listagem contendo as linguagens de programação cadastradas na ferramenta.

Os serviços de *web service* da ferramenta ReUse permite que possam ser utilizadas funções da ferramenta pro outras aplicações, como é o caso do plug-in desenvolvido para o Eclipse, onde foram utilizados todos os serviços implementados.

4.5 Plug-in para o Eclipse

A integração do repositório com o ambiente de desenvolvimento utilizado pelo programador facilita na busca e reuso de artefatos. A criação de um plug-in para um ambiente de desenvolvimento integrado é uma boa opção para disponibilizar os artefatos no próprio ambiente.

Por esse motivo, desenvolvemos um plug-in da ferramenta ReUse para a plataforma Eclipse (ECL 2008). O eclipse é uma IDE que permite o desenvolvimento de software em qualquer linguagem. O propósito da plataforma Eclipse é fornecer os serviços necessários para a integração de ferramentas de desenvolvimento de software, que são implementadas como plug-ins. Esse modelo baseado em plug-ins torna a plataforma eclipse extensível permitindo que novas funcionalidades sejam facilmente adicionadas (GAL 2003).

O plug-in para o Eclipse da ferramenta ReUse foi desenvolvido utilizando a linguagem Java e o ambiente de desenvolvimento de plug-ins (PDE) da própria ferramenta eclipse. O plug-in desenvolvido permite que o usuário liste artefatos, através de um filtro de busca, e selecione artefatos para download.

Ao instalar o plug-in, é exibido na barra de ferramentas o ícone da ferramenta, como mostrado na figura 4.12, que permite acessar a listagem de artefatos. Inicialmente, a listagem é exibida em branco. O usuário pode, então, realizar uma busca, selecionando

o ícone de busca ou o acessando pelo menu da listagem, como pode ser visto na imagem 4.13.

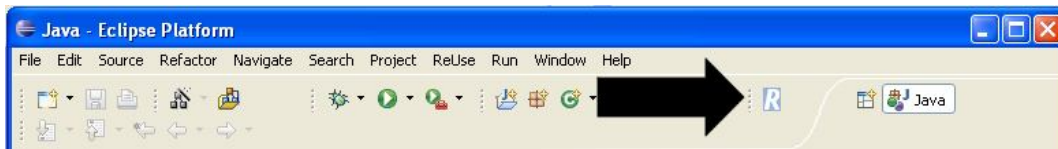


Figura 4.12: Ícone do plug-in reuse.

Após selecionar a opção de busca, o eclipse exibe a tela de filtragem (figura 4.13), onde o usuário informa os dados que deseja buscar. A busca pode ser realizada utilizando três parâmetros: palavras-chave, tipo de artefato e linguagem de programação (apenas para artefatos de implementação). A listagem de linguagens de programação disponíveis é obtidas através de uma chamada ao *web service* *getProgrammingLanguages*, descrito anteriormente.

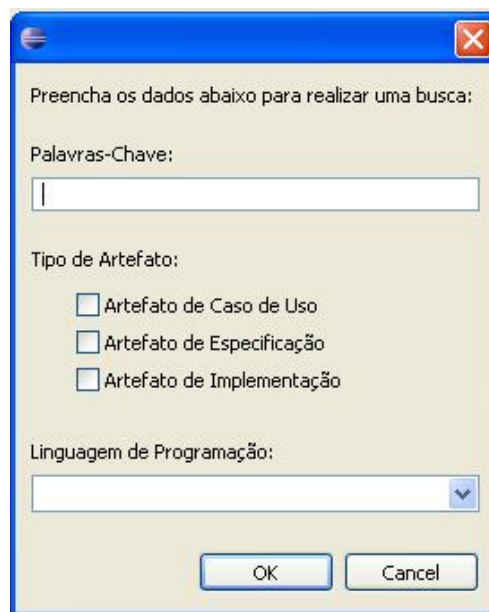
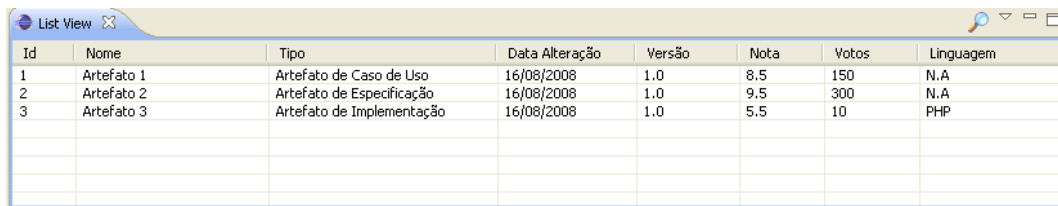


Figura 4.13: Janela de Busca de Artefatos no Plug-in.

A busca é realizada através de uma chamada ao *web service* de busca, *SearchArtifacts*, disponibilizado pela ferramenta ReUse. Após realizar a busca, o plug-in lista todos os artefatos que atendem aos critérios selecionados pelo usuário, como mostrado na figura 4.14.



Id	Nome	Tipo	Data Alteração	Versão	Nota	Votos	Linguagem
1	Artefato 1	Artefato de Caso de Uso	16/08/2008	1.0	8.5	150	N.A
2	Artefato 2	Artefato de Especificação	16/08/2008	1.0	9.5	300	N.A
3	Artefato 3	Artefato de Implementação	16/08/2008	1.0	5.5	10	PHP

Figura 4.14: Listagem de Artefatos no Plug-in.

O usuário pode, então, solicitar o download do artefato, clicando duas vezes na linha correspondente a ele na tabela. Quando solicitado o download, o plug-in faz uma chamada ao *web service* de download da ferramenta, que retorna a localização do arquivo no servidor. Para o usuário, é exibida uma janela que possibilita salvar o arquivo, como pode ser visto na figura 4.15. Após salvo, o artefato pode ser utilizado junto com o projeto.

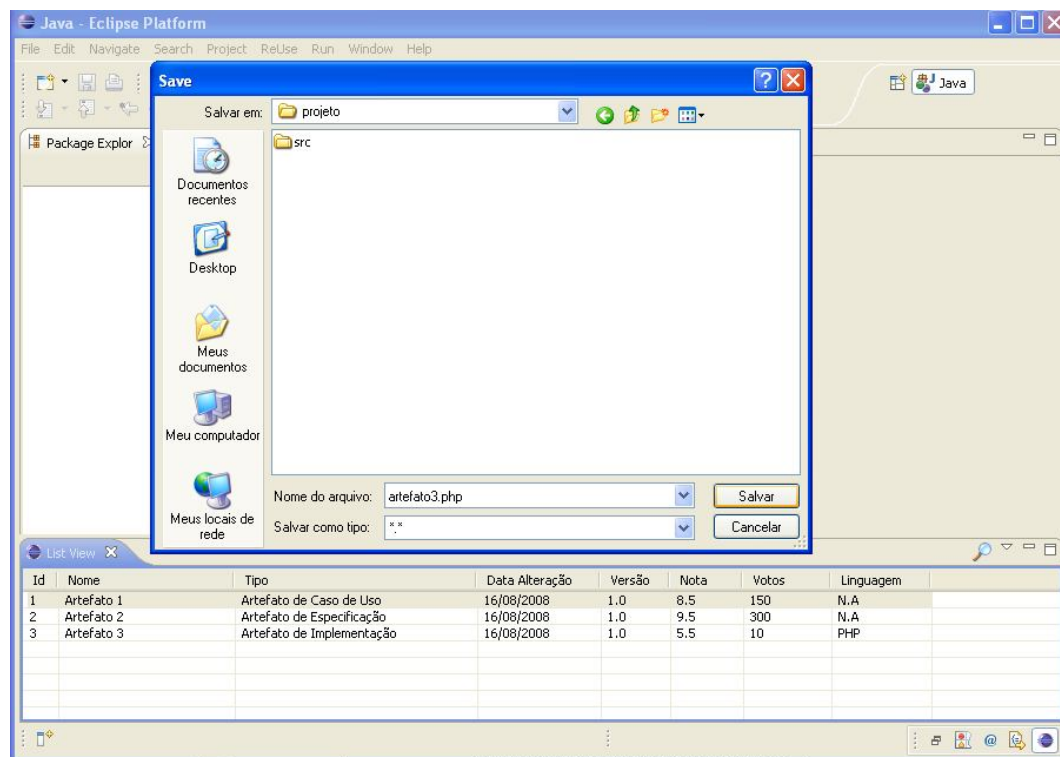


Figura 4.15: Janela para salvar artefato no projeto.

4.6 Controle de Versões

Visando aprimorar o controle de versões da ferramenta ReUse, utilizamos o sistema de controle de versões subversion (SUB 2008).

O subversion (SVN) é um sistema de controle de versões gratuito que permite o gerenciamento de arquivos e diretórios, e de mudanças realizadas neles. Ele permite a recuperação de antigas versões dos dados armazenados, ou que se examine o histórico de mudanças dos arquivos.

A integração da ferramenta ReUse com o SVN permite que o usuário adicione diversas versões a um único artefato, visualize o histórico de versões e realize o download de uma versão antiga. A integração foi feita utilizando a camada de interface VersionControl_SVN (VER 2008). Esta camada de interface permite que a ferramenta execute os comandos SVN utilizando a linguagem PHP.

Para realizar a integração da ferramenta com o SVN, foi criado um repositório SVN para os arquivos contidos no diretório de upload de arquivos da ferramenta.

Ao cadastrar um artefato no sistema, o usuário escolhe o arquivo que deseja enviar para a ferramenta. A ferramenta, então, faz o upload do arquivo e o adiciona ao repositório do sistema de controle de versões.

Ao acessar um artefato já cadastrado na ferramenta, o usuário pode escolher entre as opções de adicionar uma nova versão, através do botão de *Adicionar versão* ou visualizar o histórico de versões do artefato através do botão *Versões anteriores*, como pode ser visto na figura 4.16.



Figura 4.16: Opção de adicionar uma nova versão ou visualizar o histórico de versões de um artefato.

Ao selecionar a opção de adicionar uma nova versão, é exibida uma tela como a mostrada na figura 4.17 ao usuário, onde ele seleciona o arquivo e adiciona um comentário sobre a versão que irá incluir. O arquivo incluído deve ter o mesmo nome do arquivo cadastrado, caso contrário, a ferramenta alerta a diferença e não adiciona o novo arquivo.

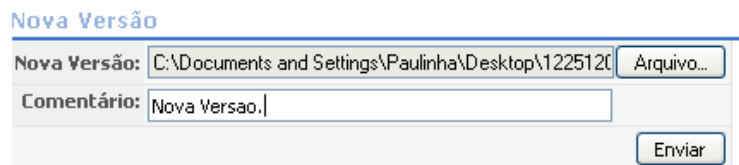


Figura 4.17: Tela para inclusão de nova versão de um artefato.

Na opção de visualizar versões anteriores, uma tela como exemplificada na figura 4.18 é exibida. Nela, o usuário pode verificar um histórico de versões do artefato, e os comentários sobre cada versão. Além disso, ele pode realizar o download de uma versão anterior.

Versões Anteriores

Data	Revisão	Comentário	Download
28/10/2008 12:41	40	Nova Versao.	Baixar Arquivo
28/10/2008 12:37	39	Bug Corrigido.	Baixar Arquivo
28/10/2008 12:36	38	Nova Funcionalidade.	Baixar Arquivo
28/10/2008 12:36	37	Correcao de bug.	Baixar Arquivo
28/10/2008 12:35	36	Arquivo Adicionado	Baixar Arquivo

Figura 4.18: Histórico de versões de um artefato.

Existe também a opção de o usuário visualizar as diferenças entre duas versões do artefato. Na tela com o histórico de versões (figura 4.18), o usuário deve selecionar as duas versões que deseja comparar e clica no botão “Comparar Versões”. A ferramenta exibe, numa nova janela, o arquivo de comparação de versões, obtido através do comando *diff* do svn. Na figura seção 4.9.3 há um exemplo concreto da comparação de versões.

4.7 Notificação de Eventos

Além das novas funcionalidades descritas acima, ainda foi implementado na ferramenta um sistema de notificação de eventos ao usuário. Desta forma, o usuário pode solicitar ser notificado por e-mail ao serem adicionadas novas versões ao artefato ou ao serem adicionados comentários ao artefato.

O usuário pode solicitar a notificação de eventos de duas formas, descritas a seguir.

A primeira delas permite que o usuário seja notificado de eventos relacionados a um artefato específico. Ao visualizar um artefato, o usuário pode selecionar a opção de “*Quero ser notificado de novos eventos correspondentes a este artefato*”. Desta forma, sempre que for incluída uma nova versão ou um novo comentário ao artefato em questão o usuário receberá um e-mail. Esta opção pode ser vista também na figura 4.4.

A segunda forma, que pode ser vista na figura 4.19, permite que o usuário seja notificado de eventos relacionados aos artefatos incluídos por ele. O usuário seleciona, no menu de configurações a opção de “*Quero ser notificado de novos eventos correspondentes a artefatos incluídos por mim.*”. Desta forma, sempre que ocorrer algum evento relacionado a algum artefato incluído por este usuário, ele será notificado por e-mail.

Figura 4.19: Configuração de Notificações de Eventos.

4.8 Ajuda Sobre a Ferramenta

Na intenção de ajudar o usuário a entender o funcionamento da ferramenta, em cada tela há um botão de ajuda, com o qual o usuário pode obter mais detalhes sobre a tela que está utilizando.

Por exemplo, na tela de glossário, ao clicar no botão de ajuda, mostrado na figura 4.20, o sistema exibe a tela de ajuda mostrada na figura 4.21.

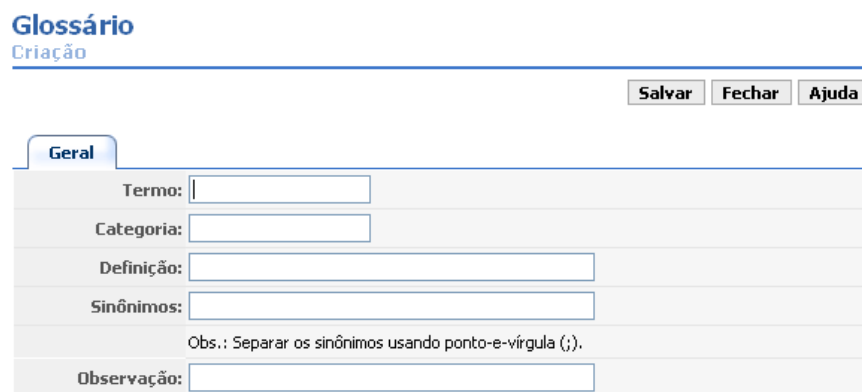


Figura 4.20: Tela de Inserção de Termo no Glossário Contendo o Botão de Ajuda

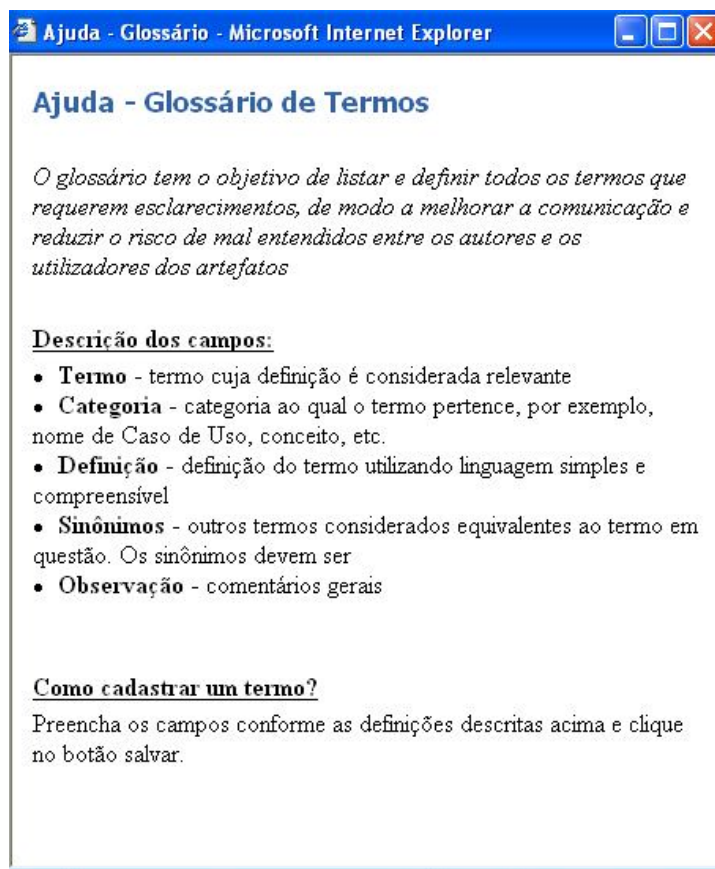


Figura 4.21: Janela Contendo Ajuda Sobre a Inserção de Termo no Glossário

4.9 Exemplo de Inclusão de Artefato Utilizando as Novas Funcionalidades

Nesta seção é apresentado um exemplo de uso de cada uma das funcionalidades adicionadas e das melhorias realizadas na ferramenta. Conforme possível, os exemplos estão relacionados entre si, para demonstrar o relacionamento entre cada um dos módulos.

4.9.1 Inserção de Termo no Glossário

A inserção de um termo no glossário inicia quando o usuário seleciona no menu a opção Glossário. Neste menu são apresentadas duas opções: cadastro e pesquisa. Na opção de cadastro, é exibida para o usuário a tela mostrada anteriormente na figura 4.20.

O usuário preenche os campos de acordo com o termo que deseja inserir. A figura 4.22 mostra o exemplo da inserção do termo “login” no glossário. No caso, foram cadastrados os sinônimos *sign in*, acesso, credenciais e identificação para o termo login. Após preencher os campos e clicar no botão “Salvar”, é exibida para o usuário a tela (figura 4.23) com a listagem de todos os termos cadastrados. Nesta tela o usuário pode filtrar os termos, além de acessar o formulário de edição de um determinado termo (semelhante ao de cadastro).

Glossário

Criação

Salvar

Fechar

Geral

Termo:	login
Categoria:	formulário
Definição:	formulário para acesso ao sistema
Sinônimos:	sign in; acesso; credenciais; identificação
Obs.:	Separar os sinônimos usando ponto-e-vírgula (;).
Observação:	

Figura 4.22: Inserção do termo *login* no glossário

Glossário

Pesquisa

Pesquisar

Identificador:	
Termo:	
Categoria:	
Definição:	
Sinônimos:	
Observação:	

Resultado da Pesquisa

Novo Termo

Excluir

<input type="checkbox"/>	Ident.	Termo	Categoria	Definição	Observações
<input type="checkbox"/>	11	data		data	
<input type="checkbox"/>	22	login	formulário	formulário para acesso ao sistema	
<input type="checkbox"/>	8	Validar		validacao	

Figura 4.23: Listagem de termos cadastrados no glossário

4.9.2 Inserção de Artefato Utilizando o Campo de Palavras-Chave

Todos os passos da inserção de um artefato no sistema são mostrados em (MAR 2008). Porém, a inserção do campo de palavras-chave modificou o primeiro passo (na interface, a primeira aba - Geral). Neste campo o usuário tem a opção de digitar palavras-chave relacionadas com o artefato que está inserindo e também de pedir que a ferramenta sugira palavras que possam estar relacionadas.

Para a sugestão de palavras, a ferramenta busca, no glossário, sinônimos para as palavras digitadas pelo usuário no campo “Nome”.

Ao inserir, por exemplo, o artefato de implementação “*Login*”, exibido na figura 4.24 (interface) e 4.25 (código-fonte), o usuário utiliza a tela de cadastro de artefatos de implementação. Nesta tela, ele pode solicitar sugestões de palavras-chave ao sistema.



LOGIN

Usuário:

Senha:

Figura 4.24: Exemplo de Tela de Login

```
<html>
<head>
  <title>Tela de Login</title>
  <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body>
<table>
<tr>
  <td colspan="2">LOGIN</td>
</tr>
<tr>
  <td>Usuário:</td>
  <td><input type="text"></td>
</tr>
<tr>
  <td>Senha:</td>
  <td><input type="password"></td>
</tr>
<tr>
  <td colspan="2"><input type="button" value="Enviar"></td>
</tr>
</table>
</body>
</html>
```

Figura 4.25: Código fonte da Tela de Login

Ao solicitar sugestões, o sistema abre uma janela com a lista de palavras que podem estar relacionadas ao sistema. No exemplo, o nome do artefato cadastrado é “*Login*”. O sistema, portanto, exibirá na listagem palavras cadastradas como sinônimas à “*Login*”, se existentes. No exemplo, cadastramos, na seção anterior (4.8.1), sinônimos à palavra *login*, portanto, esses sinônimos serão sugeridos, como pode ser visto na figura 4.26.

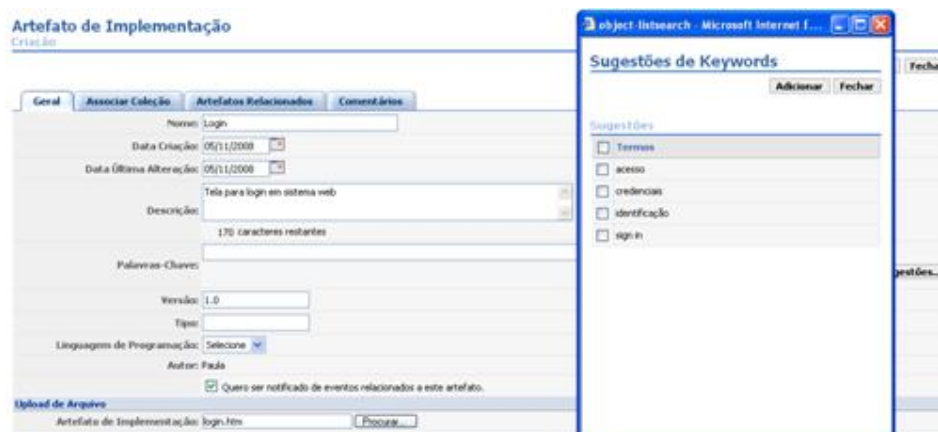


Figura 4.26: Solicitação de Sugestões no Cadastro do Artefato de Implementação “Login”.

O usuário então pode selecionar as palavras que considera relevantes para o artefato que está cadastrando. Além disso, o preenchimento do campo de palavras-chave não é restrito às palavras sugeridas pelo sistema, ou seja, o usuário pode, ainda, adicionar qualquer palavra que considerar relevante. A figura 4.27 mostra a tela de cadastro com todos os campos da aba Geral preenchidos.

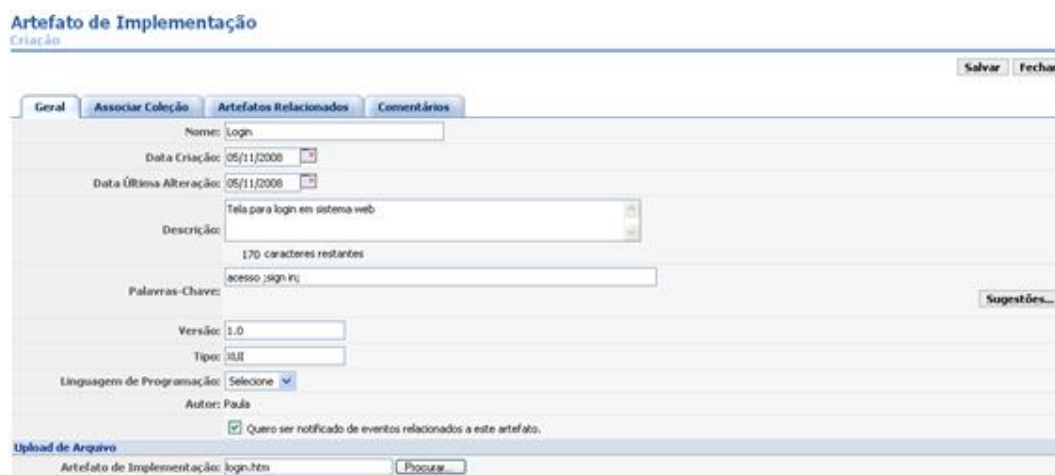


Figura 4.27: Cadastro do Artefato de Implementação “Login” com Todos os Campos da Aba Geral Preenchidos.

4.9.3 Solicitação Recebimento de Notificações

Para solicitar o recebimento de notificações de alterações no artefato “Login”, o usuário, acessando a tela de visualização (ou edição, para o caso do autor) do artefato, pode selecionar a checkbox “Quero ser notificado de eventos relacionados a este artefato”, como pode ser visto na figura 4.27, mostrada na seção anterior.

Desta forma, sempre que o artefato for alterado, ou seja, for inserida uma nova versão ou adicionado um novo comentário, o usuário receberá um e-mail da ferramenta informando o artefato no qual ocorreu a alteração.

4.9.4 Adicionando Comentários ao Artefato

Um exemplo de inserção de um comentário ao artefato “*Login*” pode ser visto na figura 4.28. O usuário seleciona, através da combo “Tipo”, o tipo de comentário que deseja incluir (no exemplo, Sugestão).

A figura 4.29 mostra a listagem de comentários já cadastrados ao artefato.

Comentários

Identificador: 38

Data Inclusão: 15/11/2008

Tipo: Sugestão

Comentário: Poderia alinhar o botão à direita

166 caracteres restantes

[Lista de comentários relacionados ao artefato](#)

Figura 4.28: Inserção de um Comentário do Tipo Sugestão ao Artefato “*Login*”

Lista de comentários relacionados ao artefato			
Data de Inclusão	Autor	Tipo	Comentário
2008-11-11 00:00:00	Paula	Sugestão	Poderia colocar o botão alinhado à direita.
2008-11-12 00:00:00	Paula	Avaliação	muito bom!

Figura 4.29: Lista de Comentários Adicionados ao Artefato

Após a inserção de um comentário, a ferramenta envia um e-mail a todos os usuários que solicitaram notificações a respeito daquele artefato, como exemplificado na seção anterior (4.6.5).

4.9.5 Utilizando o Controle de Versões

Para inserir uma nova versão ao artefato “*Login*”, é preciso acessar a tela de edição deste artefato e clicar no botão “Nova Versão”. A figura 4.32 mostra um exemplo de inserção de uma nova versão do artefato. A nova versão é mostrada na figura 4.30 e o código fonte é mostrado na figura 4.31

LOGIN

Usuário:

Senha:

Figura 4.30: Interface da Nova Versão do Artefato “*Login*”

```

<html>
<head>
  <title>Tela de Login</title>
  <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body>
<table>
<tr>
<td colspan="2"><strong>LOGIN</strong></td>
</tr>
<tr>
<td colspan="2" style="border-top: 1px solid #000000 ">&nbsp;</td>
</tr>
<tr>
<td>Usuário:</td>
<td><input type="text"></td>
</tr>
<tr>
<td>Senha:</td>
<td><input type="password"></td>
</tr>
<tr>
<td colspan="2" align="right"><input type="button"
value="Enviar"></td>
</tr>
</table>
</body>
</html>

```

Figura 4.31: Código-fonte da Nova Versão do Artefato “Login”

Nova Versão

Nova Versão:	<input type="text" value="login"/>	<input type="button" value="Procurar..."/>
Comentário:	<input type="text" value="melhorias na interface"/>	
		<input type="button" value="Enviar"/>

Figura 4.32: Inserção de Nova Versão ao Artefato “Login”

Após adicionar uma versão, o usuário pode visualizar o histórico de versões. A tela de histórico de versões do artefato “Login” pode ser vista na figura 4.33. A partir desta tela, o usuário pode ver a diferença entre duas versões. No caso, podemos selecionar as duas versões inseridas nos exemplos acima e clicar no botão “Comparar”. O resultado da comparação é exibido na figura 4.34. Nela, podemos ver o resultado do comando *diff* do SVN entre o código-fonte exibido na figura 4.25 e o exibido na figura 4.32.

Versões Anteriores

	Data	Revisão	Comentário	Download
<input type="checkbox"/>	15/11/2008 16:51	50	melhorias na interface	Baixar Arquivo
<input type="checkbox"/>	15/11/2008 16:49	49	Arquivo Adicionado	Baixar Arquivo

Figura 4.33: Histórico de Versões do Artefato “Login”.

```

Comparação de arquivos

<pre>Index: 1226774942login.txt
=====
--- 1226774942login.txt (revision 49)
+++ 1226774942login.txt (revision 50)
@@ -6,9 +6,12 @@
<body>
<table>
<tr>
-   <td colspan="2">LOGIN</td>
+   <td colspan="2"><strong>LOGIN</strong></td>
</tr>
<tr>
+   <td colspan="2" style="border-top:1px solid #000000 ">&nbsp;</td>
+</tr>
+<tr>
    <td>Usuário:</td>
    <td><input type="text"></td>
</tr>
@@ -17,7 +20,7 @@
    <td><input type="password"></td>
</tr>
<tr>
-   <td colspan="2"><input type="button" value="Enviar"></td>
+   <td colspan="2" align="right"><input type="button" value="Enviar"></td>
</tr>
</table>
</body></pre>

```

Figura 4.34: Resultado da Comparação entre as Versões

4.9.6 Busca Utilizando a Tela de Pesquisa Geral

Além das telas de pesquisa específicas de cada artefato, já existentes na ferramenta, foi adicionada uma tela de pesquisa geral. Um exemplo de busca pelo artefato de especificação “Login” na tela de Pesquisa Geral pode ser visto na figura 4.35. Nela, o usuário digita o termo “login” e o sistema retorna uma lista de artefatos.

Pesquisa

Geral

Pesquisar

Buscar palavras:

☒ Casos de Uso

☒ Artefatos de Especificação

☒ Artefatos de Implementação

Resultado da Pesquisa

Filtro ativo [Limpar]

<input type="checkbox"/>	Ident.	Data Última Alteração	Nome	Tipo	Descrição
<input type="checkbox"/>	48	05/11/2008	Login	Artefato de Implementação	Tela para Login em Sistema Web

Figura 4.35: Busca pelo Artefato “Login”

4.9.7 Utilizando o Plug-In para o Eclipse

A instalação do plug-in para o eclipse é feita de forma simples: basta colocar os arquivos do plug-in na pasta de plug-ins onde o Eclipse está instalado.

Após a instalação, o usuário pode buscar e baixar os artefatos dentro da própria ferramenta. A figura 4.36 mostra um exemplo da busca do artefato “Login” no Eclipse. O resultado da busca é exibido numa listagem, como mostrada na figura 4.37, onde também é exibida a tela de download. Após o download, o artefato está disponível para ser utilizado no eclipse.

Figura 4.36: Busca por Artefato “Login” no Eclipse

5 CONCLUSÕES E TRABALHOS FUTUROS

A ferramenta ReUse oferece um suporte ao reuso em qualquer etapa do desenvolvimento. As melhorias implementadas na ferramenta permitem que o reuso possa se tornar ainda mais fácil de ser adotado.

As melhorias no sistema de busca diminuem a dificuldade em encontrar artefatos já existentes para o reuso, facilitando ao desenvolvedor encontrar artefatos que possam ser úteis no desenvolvimento. A inclusão do glossário, notificação de eventos e classificação de comentários melhoram a comunicação entre autores e utilizadores dos artefatos, o que ajuda no processo de reutilização. O controle de versões permite um melhor gerenciamento dos artefatos da ferramenta. A integração com o ambiente de programação Eclipse permite que os desenvolvedores possam buscar artefatos reusáveis sem sair do ambiente em que trabalham, o que facilita ainda mais a adoção do reuso. Além disso, as telas de ajuda permitem um melhor entendimento da ferramenta e, portanto, facilitam o uso.

A eficiência da ferramenta tem uma forte dependência com a maneira como ela é utilizada. A sugestão de palavras-chave, por exemplo, só funcionará efetivamente se forem incluídos diversos termos no glossário, e se forem incluídos de forma clara e correta. A busca também depende de como os artefatos foram cadastrados na ferramenta.

Uma sugestão para trabalhos futuros seria a implementação de outros comandos *svn*, além de estender o controle de versões para as coleções, de forma que um usuário possa fazer *checkout* de toda uma coleção, não apenas de um artefato.

Outra possibilidade de trabalho futuro seria a extensão do plug-in para o eclipse, de forma que o usuário possa também cadastrar e enviar artefatos para o repositório, dentro do próprio ambiente de trabalho.

Enfim, são diversas as possibilidades de extensão e melhoria da ferramenta, desde a interface até cada um dos seus módulos.

O importante é que a ferramenta seja eficiente no suporte ao reuso de software, permitindo o desenvolvimento com e para o reuso, e a facilidade na busca por artefatos reusáveis. No reuso de software, o tempo de busca e de adaptação de um artefato já existente deve ser menor do que o tempo de construção de um artefato novo. Portanto, o uso de uma ferramenta eficiente é crucial para o sucesso da aplicação desta técnica.

REFERÊNCIAS

- [ALM 2007] ALMEIDA, E. et al. **Component Reuse in Software Engineering**. [S.l.]:R.I.S.E, 2007. Acessado em (2008). Disponível em <<http://cruise.cesar.org.br/>>
- [AMB 2004] AMBLER, S. W. et al. **Extending the rup with the strategic reuse discipline**. Acessado em (2008). Disponível em: <<http://www.enterpriseunifiedprocess.com/essays/strategicReuse.html>>.
- [AMB 2007] AMBLER, S.W. et al. **Types of Reuse in Information Technology**. Acessado em (2008). Disponível em <<http://www.ambysoft.com/essays/typesOfReuse.html>>
- [ASW 2005] ASWATH, D. et al Boosting Item Keyword Search with Spreading Activation. **Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence**. Washington, DC, USA, p. 704-707, 2005.
- [CAL 2003] CALVARY, G. et al. A Unifying Reference Framework for Multi-Target User Interfaces. **Interacting with Computers**, v.15, n.3, p.289–308, 2003.
- [CER 2002] CERAMI, E. **Web Services Essentials**. O'Reilly, 2002.
- [COC 2005] COCKBURN, A. Escrevendo Casos de Uso Eficazes: um guia prático para desenvolvedores de software. Bookman, 2005
- [COX 90] COX, B. J. Planning the Software Industrial Revolution. **IEEE Softw.**, LosAlamitos, CA, USA, v.7, n.6, p.25–33, 1990.
- [ECL 2008] **Eclipse.org home**.Acessado em (2008) Disponível em: <<http://www.eclipse.org/>>.
- [EIS 2000] EISENSTEIN, J.; VANDERDONCKT, J.; PUERTA, A. Adapting to Mobile Contexts With User-Interface Modeling. In: **THIRD IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS (WMCSA'00)**, 2000, Washington, DC, USA. **Proceedings**. . . IEEE Computer Society, 2000. p.83
- [GAL 2003] GALLARDO, D. et al **Eclipse in Action: a Guide for Java Developers**. Manning Publications Co, 2003
- [GAM 2000] GAMMA, E. et al. **Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos**. [S.l.]:

Bookman, 2000

- [HEN 94] HENNINGER, S. Using Iterative Refinement to Find Reusable Software. **IEEE Software**, Los Alamitos, CA, USA, v.11, n.5, p.48-59, 1994.
- [HEN 97] HENNINGER, S. An Evolutionary Approach to Constructing Effective Software Reuse Repositories. **ACM Transactions on Software Engineering and Methodology (TOSEM)**, New York, NY, USA, v.6, n.2, p.111-140, 1997
- [JAC 92] JACOBSON, I. **Object-Oriented Software Engineering: A Use Case Driven Approach**. Addison-Wesley, 1992.
- [KRU 92] KRUEGER, C. W. Software Reuse. **ACM Comput. Surv.**, New York, NY, USA, v.24, n.2, p.131–183, 1992
- [LAR 2001] LARMAN, C. **Utilizando UML e Padrões: Uma introdução à Análise e ao Projeto Orientados a Objetos**. Bookman, 2001.
- [LAR 2004] LARMAN, C. **Utilizando UML e Padrões: Uma introdução à Análise e ao Projeto Orientados a Objetos**. Bookman, 2004.
- [MAR 2007] MARTINS, J. **ReUse – Uma Ferramenta de Suporte ao Reuso**. 2008. Projeto de Diplomação (Bacharelado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre
- [MCI 68] MCILROY. Mass produced software components. **Report on a conference by the NATO Science Committee**, p.138–150, 1968
- [MIN 2007] MINNETO, E. L. **Frameworks para Desenvolvimento de Software em PHP**. Novatec, 2007.
- [MYS 2008] **MySQL Workbench 5.0**. Acessado em (2008). Disponível em <[HTTP://dev.mysql.com/workbench/](http://dev.mysql.com/workbench/)>
- [MOR 2007] MOREIRA, A. A. **Reuso de IHC Orientado a Padrões, Dirigido por Casos de Uso e Integrado a um Processo de Desenvolvimento Baseado em UML**. 2007. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul.
- [NUS 2007] AYALA, D. et al **NuSOAP – SOAP Toolkit for PHP**. Acessado em (2008). Disponível em <<http://sourceforge.net/projects/nussoap/>>
- [OMG 2005] OMG. **OMG - Object Management Group, Reusable Asset Specification**. Acessado em (2008). Disponível em:<<http://www.omg.org/docs/formal/05-11-02.pdf>>
- [OMG 2008] OMG. **OMG Unified Modeling Language (OMG UML), Superstructure**. Acessado em (2008). Disponível em <<http://www.omg.org/spec/UML/2.1.2/>>
- [SUB 2008] **Subversion**. Acessado em (2008). Disponível em <subversion.tigris.org>

- [TRA 88] TRACZ, W. Software Reuse Myths. **SIGSOFT Softw. Eng. Notes**, New York, NY, USA, v.13, n.1, p.17–21, 1988.
- [VAN 93] VANDERDONCKT, J. M.; BODART, F. Encapsulating knowledge for intelligent automatic interaction objects selection. In: CHI '93: PROCEEDINGS OF THE INTERACT '93 AND CHI '93 CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1993, New York, NY, USA. **Anais. . . ACM**, 1993. p.424–429.
- [VER 2008] **VersionControl_SVN**. Acessado em (2008). Disponível em <http://pear.php.net/package/VersionControl_SVN/>
- [YON 98] YONGBEON K. STOHR, E. Software Reuse: Survey and Research Directions. **Journal of Management Information Systems**, v.14, n.4, p.113–147, 1998.
- [ZIR 95] ZIRBES, S. F. **A Reutilização de Modelos de Requisitos de Sistemas por Analogia**: Experimentação e Conclusões. 1995. Tese (Doutorado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul.