

Reutilização de Objetivos na Engenharia de Requisitos

Karla Donato Fook, Zair Abdelouahab

Departamento de Engenharia de Eletricidade, Universidade Federal do Maranhão,

E-mail:ki@elo.com.br, zair@dee.ufma.br, zair@elo.com.br

Resumo. Este artigo apresenta um método de reutilização de requisitos através os objetivos, chamado MROBJER. O método permite a reutilização de requisitos de um sistema através a reutilização de seus objetivos. A reutilização de objetivos pode ocorrer entre aplicações pertencentes ao mesmo domínio denominada “reutilização interdomínio” e também entre aplicações de diferentes domínios denominada “reutilização intersistemas”. MROBJER prove também o reuso parametrizado de objetivos. Isto é, os objetivos genéricos possam ser reutilizados por outras aplicações passando-se como parâmetros fatores diferenciadores dentro dos objetivos. Para auxiliar o desenvolvedor, uma ferramenta foi desenvolvida, chamada FROBJER. Um exemplo na área bancária é ilustrado para mostrar o uso do método.

1. Introdução

Com o advento da Orientação a Objetos, a reutilização de código foi abordada através da herança e desde então o reuso vem sendo bastante difundido na Engenharia de Software.

Benefícios como a redução de tempo e de custos durante o processo de desenvolvimento passaram a caracterizar a reutilização. Propostas para o reuso de outros artefatos como requisitos, especificações e processos têm sido apresentadas.

Na Engenharia de Requisitos, primeira fase do ciclo de vida do desenvolvimento de software, um dos grandes desafios é a obtenção de requisitos que sejam os mais precisos possível para possibilitarem um bom entendimento do que é desejado e esperado do produto final do processo[14][15][17]. Sob esta perspectiva, a reutilização está sendo abordada como uma forma de obter requisitos mais confiáveis, uma vez que já foram implementados previamente em outros sistemas.

As abordagens objetivam obter o aproveitamento máximo da reutilização na Engenharia de Requisitos através de artefatos como cenários[12], requisitos[3] e discriminantes[11].

Entretanto, isto não significa que a reutilização possa ser realizada de qualquer maneira. Ela requer bastante cuidado, pois se deve saber o quê reutilizar, como e porquê. A reutilização deve primar pela qualidade para não proporcionar erros como o Ariane 5[2].

Neste artigo, será abordado o MROBJER, Método para Reutilização de Objetivos na Engenharia de Requisitos.

Este método comporta a reutilização de objetivos entre aplicações de um mesmo domínio, denominada Intersistemas, e entre aplicações pertencentes a diferentes domínios, denominada Interdomínios. Outro ponto que permite certa flexibilidade ao reuso é a parametrização que permitirá, no MROBJER, que objetivos genéricos

possam ser reutilizados por outras aplicações passando-se como parâmetros fatores diferenciadores dentro dos objetivos.

O método conta com o auxílio de uma ferramenta de software que será manuseada pelo engenheiro de requisitos em conjunto com o usuário. Tal ferramenta consiste basicamente de uma base de dados contendo os objetivos genéricos reutilizáveis e comporta uma estrutura que permite facilitar a seleção destes.

A seção 2 cita alguns dos trabalhos propostos para a reutilização na Engenharia de Requisitos. O MROBJER é melhor descrito nas seções 3 e 4.

A seção 5 mostra como é realizada reutilização utilizando-se a ferramenta de software, o FROBJER (Ferramenta para Reutilização de Objetivos na Engenharia de Requisitos). O produto desta ferramenta deverá ser um documento que contenha os objetivos da Aplicação a ser desenvolvida.

2. Trabalhos Relacionados à Reutilização na Engenharia de Requisitos

A reutilização tem sido vista como uma das formas de se obter requisitos mais precisos e corretos na Engenharia de Requisitos. Isto porque se parte do princípio de que os requisitos que irão ser reutilizados já foram previamente validados e implementados.

São várias as abordagens de reutilização na Engenharia de Requisitos que estão sendo propostas atualmente. Pode-se citar:

- MRAM (*Method for Requirements Authoring and Management*) [Mannion 1998]: se propõe a reutilizar requisitos e discriminantes (tipo especial de requisitos que diferencia um sistema de outro) dentro de uma Família de Aplicações;
- FORE (*Family of Requirements*) [5]: objetiva desenvolver uma família de requisitos genéricos e reutilizáveis;
- RARE (*Reuse-Assisted Requirements Engineering*) [3]: constrói um repositório de requisitos reutilizáveis;
- *Scenario Method Chunks* [12]: trabalha com blocos de cenários como sendo componentes reutilizáveis;

Entretanto, a reutilização deve ser feita considerando-se vários critérios para ser executada com sucesso como o contexto do artefato a ser reutilizado [6], por exemplo.

As propostas possuem peculiaridades, mas algumas características são comuns a determinadas abordagens, tais como:

- Limitação do Domínio da Aplicação [5][11], agrupa os sistemas em famílias de produtos.
- Orientação a Objetivos [8][1][4], que se concentra na identificação de objetivos típicos em um domínio, com base em que sistemas individuais em um domínio particular são provavelmente compartilhadores de muitos objetivos.
- Parametrização [8][7], que trabalha basicamente com requisitos genéricos, onde estes possuem parâmetros que permitem a sua reutilização por outros sistemas. Os detalhes específicos de sistemas são fatorados de fora como parâmetros nos requisitos.

- Analogia é também usada como alternativa para a reutilização na etapa de análise dos requisitos[9][10]. Ela sugere que ao se fazer analogia entre dois sistemas complexos de software, devem surgir especificações a serem reutilizadas. O reuso através da analogia considera o conhecimento do domínio um dos fatores de relevância para o sucesso da reutilização.

A abordagem Modelagem de domínio orientada a objetivo[8] se utiliza de uma característica similar à abordagem de Famílias de Aplicação, que emprega a limitação de um domínio, isto é, trabalham com sistemas de naturezas semelhantes.

Assim como na Modelagem de domínio orientada a objetivo[8], o KAOS propõe Métodos Baseados em Objetivos[4]. Outra abordagem utilizando objetivos é apresentada pelo grupo CREWS¹ desenvolveu uma abordagem usando Objetivos. Trata-se da modelagem de Objetivos usando cenários onde o objetivo é particionado e suas partes combinadas para a descoberta de novos objetivos[13]. Para tal foi desenvolvida a ferramenta de suporte L'ECRITOIRE[16].

O MROBJER delimita o domínio da aplicação para se chegar a objetivos genéricos a serem reutilizados por novas aplicações dentro e fora deste domínio, ou seja, reutilização Intersistemas e Interdomínio, respectivamente.

Os objetivos são descritos através de cenários no MROBJER e são tratados como sendo genéricos ou específicos. Objetivos genéricos são aqueles com possibilidade de serem reutilizados por outros sistemas. Objetivo específicos são aqueles que são inerentes à Aplicação.

Pode-se então dizer que as características apontadas de Famílias de Aplicação, Modelagem de domínio orientada a objetivo e Requisitos Parametrizados nortearam a abordagem aqui proposta.

3. O MROBJER

O método descrito nesta seção se propõe a efetivar a reutilização na Engenharia de Requisitos através de Objetivos do Sistema. A reutilização em si é propiciada pela análise de objetivos e cenários genéricos contidos em um repositório. Os objetivos são obtidos a partir de um mapeamento de uma estrutura que descreve o contexto da aplicação em desenvolvimento.

A aplicabilidade do mapeamento ocorrerá em dois momentos distintos. O primeiro deles é uma fase inicial que objetiva o carregamento do repositório. É este repositório que será consultado para a reutilização objetivos de aplicações de domínios semelhantes ou não. Inicialmente algumas estruturas deverão ser mapeadas para que o repositório seja carregado.

Um segundo momento para o mapeamento ocorrerá quando for necessário incluir um novo domínio que não pertença ao repositório. Neste caso, o engenheiro terá que acrescentar os objetivos encontrados no novo mapeamento.

Após encontrados, os objetivos e cenários genéricos são formalizados. A genericidade destes é denotada por parâmetros que poderão adequar os objetivos ao novo sistema quando estes forem ser reutilizados. A partir da aplicação a ser desenvolvida tem-se duas etapas a se realizar:

- a) Encontrar o seu domínio e subdomínios afins seguindo um roteiro composto por questões e diretrizes;
- b) Descrever a aplicação através de agregações funcionais e objetivos. Os objetivos serão detalhados sob a forma de cenários.

A segunda etapa é completada ao se distinguir objetivos genéricos dos demais. A figura a seguir denota os níveis da estrutura proposta.

Descrevendo de forma mais detalhada o mapeamento da estrutura tem-se que, uma vez encontrado o domínio, será montada uma estrutura em árvore que terá subdomínios em um próximo nível. Estes representam ambientes ou plataformas de aplicabilidade dos sistemas pertencentes ao domínio. Tais ambientes englobarão diferentes aplicações, entre elas a própria aplicação em desenvolvimento.

A etapa seguinte, é desmembrar a aplicação em agregações funcionais que a compõem e então listar os objetivos de cada agregação funcional, ilustrando-os através de cenários.

Os cenários são classificados em dois níveis: compostos e primitivos. A princípio, cenários compostos ilustrarão o comportamento do sistema em uma determinada agregação funcional. As ações de um cenário composto poderão ser melhor detalhadas. Estas ações de detalhamento constituirão os cenários primitivos.

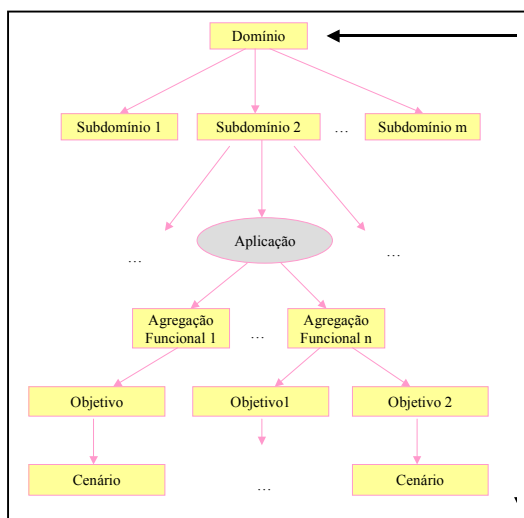


Figura 3.1 – Representação da estrutura

A idéia do mapeamento dos cenários na estrutura está representada na figura 3.1. Além de pertencerem à estrutura, os cenários serão descritos de forma textual no repositório.

Uma vez listados os objetivos e seus respectivos cenários, parte-se para a identificação de objetivos genéricos. Em seguida, procura-se distinguir as partes variáveis dentro dos requisitos genéricos e ilustrá-las como parâmetros.

Após isto, tem-se as estruturas mapeadas e o engenheiro pode partir para a reutilização.

As etapas do mapeamento serão vistas com mais detalhes a seguir.

3.1 Mapeamento do Domínio

Na primeira etapa ocorre a construção de uma árvore de decomposição na qual tem-se representado um domínio com seus subdomínios e as aplicações pertencentes a estes.

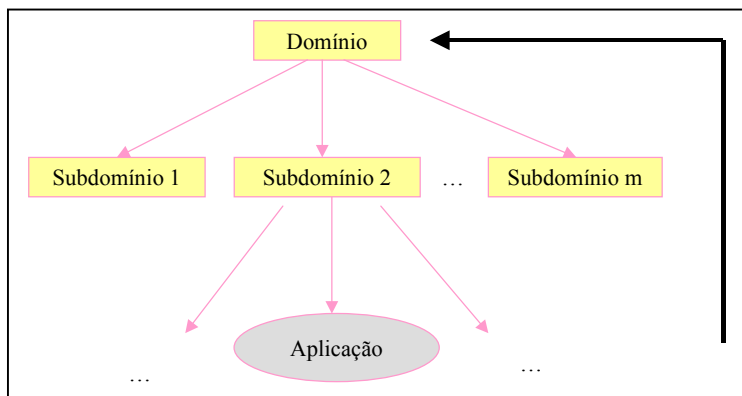


Figura 3.2 Mapeamento do Domínio

A árvore é elaborada com o Domínio(raiz) sendo seu nível superior, um nível intermediário composto de Subdomínios, que descrevem as sub-áreas de aplicação do sistema e a sua plataforma(sistemas corporativos, sistemas net, auto-atendimento, etc.), e, por fim, o nível que contém a aplicação a ser desenvolvida.

O ponto de partida para o mapeamento da estrutura é a aplicação a ser desenvolvida e o primeiro passo é encontrar o seu Domínio. A figura 3.2 exhibe a estrutura obtida nesta primeira fase, enfatizando o ponto de partida com uma seta que parte da Aplicação até o Domínio.

Um documento com algumas diretrizes poderá servir de roteiro para que o engenheiro de requisitos encontre o domínio da aplicação a ser desenvolvida.

3.1.1 – Diretrizes de busca do domínio

- O domínio caracteriza-se por ser o nível mais geral. Refere-se à área na qual o sistema será aplicado descrita de forma mais abrangente. Por exemplo: Área Médica, Bancária, Educação, Turismo, Judiciário, etc.
- O domínio deve conter várias sub-áreas de aplicação, como sistemas corporativos, auto-atendimento, internet, *help-desk* e outros específicos a cada domínio.
- Algumas perguntas-chave podem ser feitas para se descobrir o domínio de um sistema, tais como:
 - 1.1 – Qual é o ramo da empresa ou instituição aonde o sistema funcionará?
 - 1.2 – Qual é a aplicação do sistema dentro da empresa ou instituição?

Após obter as respostas, é necessário analisar se estas não compõem algum nível mais geral. Se isto ocorrer, terá sido encontrado um subdomínio da aplicação.

3.1.2 – Diretrizes para identificação dos Subdomínios

O detalhamento deste nível depende da abstração na decomposição do domínio. Não se tem um número fixo de subdomínios, pois eles podem variar de um domínio para outro.

Para se encontrar os subdomínios, pode-se lançar questões como:

2.1 – Em quais sub-áreas os sistemas pertencentes a este domínio são aplicados?

2.2 – Quais são as áreas de concentração ou de aplicabilidade que o domínio pode ter?

Diretrizes para a construção do nível das Aplicações Neste nível serão mostradas as aplicações que se encaixam em cada sub-área ou Subdomínio. Algumas questões que podem auxiliar ao engenheiro de requisitos a contemplar outras aplicações dentro de um subdomínio seriam:

3.1 – Que outras aplicações, além da aplicação em foco, podem ser implantadas neste Subdomínio?

3.2 – Que outros problemas poderiam ser melhorados ou solucionados por aplicações neste Subdomínio?

Os quesitos do questionário deverão, a princípio, ser respondidas pelo usuário/especialista para nortear o engenheiro de requisitos a mapear o domínio desejado. Entretanto, nem sempre o usuário/especialista saberá respondê-los a contento. Neste caso, cabe ao próprio engenheiro de requisitos buscar novas fontes de informação para a obtenção das respostas.

Uma vez mapeada a estrutura Domínio -> Subdomínios -> Aplicações, é concluída a primeira fase do mapeamento da estrutura.

3.2 Mapeamento da Aplicação

Para complementar a estrutura, volta-se as atenções para a aplicação em desenvolvimento e faz-se agora uma relação das funcionalidades da mesma. Isto pode ser traduzido em diversas agregações funcionais que comporão a aplicação.

A seguir, enumera-se os objetivos de cada agregação funcional. Estes são descritos através de cenários. Estes cenários poderão ser compostos ou primitivos. São chamados compostos aqueles que podem ser descritos com um maior detalhamento e primitivos aqueles que consistam de ações mais específicas.

Identificação das Agregações Funcionais. As agregações funcionais correspondem às atividades macro da aplicação em descrição. Uma agregação funcional retrata as funcionalidades da aplicação, como por exemplo funções de cadastro, segurança, consultas, etc. Algumas questões podem nortear a identificação das agregações funcionais:

1.1 – Quais são as funcionalidades gerais do Sistema?

1.2 – Que nomes poderiam agrupar objetivos afins da aplicação?

Identificação dos Objetivos das Agregações Funcionais. Esta identificação de objetivos contidos nas agregações funcionais da nova aplicação será realizada em conjunto com o usuário. As descrições dos objetivos das agregações funcionais da aplicação são obtidas com base nesta identificação. As questões seguintes podem auxiliar na obtenção destes objetivos:

2.1 – Quais são os principais objetivos a serem alcançados em determinada agregação funcional?

2.2 – Que metas a aplicação deve atingir (por agregação funcional)?

Definidos os objetivos, estes deverão ser descritos em forma de cenários, aqui tratados como sendo uma seqüência de ações a serem executadas para se alcançar um determinado objetivo. O engenheiro de requisitos deve mapear a estrutura em árvore do domínio, dos subdomínios, aplicações e agregações funcionais, para posterior apresentação desta ao usuário. As camadas da estrutura obtidas podem ser vistas na figura 3.3.

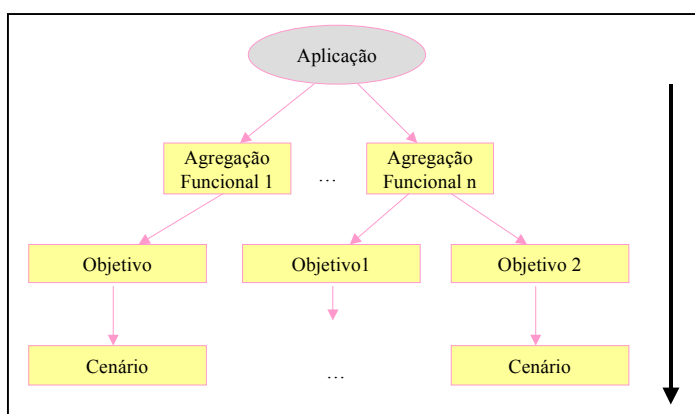


Figura 3.3 Estrutura do mapeamento da Aplicação

Caso o usuário não responda satisfatoriamente ao questionário, cabe ao engenheiro de requisitos buscar respostas a partir de novas fontes.

Ressalta-se que na primeira etapa, elabora-se a parte superior da estrutura, até o nível da aplicação e na segunda, descreve-se a aplicação através de agregações funcionais e objetivos gerais.

A reutilização em si advém do reaproveitamento de objetivos e cenários considerados como genéricos pertencentes a uma estrutura resultante do mapeamento de uma determinada aplicação por uma nova aplicação. Esta última pode pertencer a um mesmo domínio ou não.

Se um cenário composto é reaproveitado, o engenheiro de requisitos deve estar atento para o fato de que os cenários primitivos decorrentes deste cenário composto poderão ser também reaproveitados na íntegra ou não conforme a necessidade do novo sistema.

3.3 Identificação de Objetivos Genéricos

Ao se concluir o mapeamento da estrutura proposta e a identificação de objetivos, faz-se uma analogia entre as aplicações presentes em busca de similaridades entre as agregações funcionais e objetivos nas aplicações do domínio[8].

O engenheiro de requisitos deve identificar, dentre aqueles enumerados, os que poderão ser considerados como genéricos.

O próximo passo é identificar partes fixas e partes variáveis dentro dos objetivos e cenários com esta genericidade. As partes fixas geralmente são desmembradas de maneira a se obter um verbo principal e um objeto principal. As partes que forem identificadas como variáveis são definidas como parâmetros, ou seja, partes que tenham a possibilidade de aparecer em outros sistemas. Os parâmetros são identificados pelos delimitadores $\langle \rangle$. Ressalta-se que o MROBJER não contempla tratamento para sinônimos.

Há ainda partes pertencentes aos objetivos e cenários que não se enquadram nas partes identificadas. Estas são denominadas de complementos.

A formação que os objetivos e cenários devem contemplar é:

Verbo principal + Objeto principal [+ complemento + parâmetros $\langle 0..n \rangle$],

onde

- $\langle 0..n \rangle$ número de parâmetros,
- [] indicam que os parâmetros são opcionais.

Como exemplo desta formação considere o objetivo “Efetuar login de funcionário” para acessar o Sistema de Caixa do Subdomínio Automação Bancária do domínio Área Bancária.

Tem-se identificados:

- Verbo Principal: Efetuar,
- Objeto Principal: login,
- Complemento: de funcionário.

Este objetivo pode ser considerado um objetivo genérico, pois aparecerá em vários sistemas de forma semelhante. Neste caso, um cenário poderá descrever o objetivo, ou seja, para que o funcionário acesse o sistema, este(o sistema) deverá:

- 1) Solicitar código de acesso
- 2) Validar código de acesso
- 3) Solicitar senha
- 4) Validar senha
- 5) Aplicar restrições ao usuário conforme código de acesso.

A reutilização deste objetivo poderá ocorrer nos níveis de Intersistemas(grande parte dos Sistemas Corporativos possuem este cenário) e de Interdomínios(o cenário ocorre em Sistemas Corporativos de vários domínios).

Nesta abordagem um objetivo genérico deve ser codificado, isto é, deve possuir um código que o identifique no documento no qual ele será inserido para posterior busca durante o processo de reutilização. Tal codificação pode ter o seguinte formato:

Obj_{GenN} = {Domínio, “Descrição”, Verbo Principal, Objeto Principal, <Parâmetro[0..n]>}

onde: N: número identificador do Objetivo,
Descrição: o Objetivo,

n: número de parâmetros contidos no Objetivo.

Dessa forma, os objetivos genéricos e seus respectivos cenários são listados de maneira menos informal em um repositório, que pode ser um documento escrito em um processador de texto.

O intuito de se manter este repositório é que, ao se mapear estruturas, pode-se consultar o documento do repositório para se verificar a possibilidade de reutilização de objetivos por parte da nova aplicação. Os demais objetivos desta aplicação são chamados de objetivos específicos. Estes são adicionados ao documento após o término da fase da reutilização.

O objetivo “Retirar valor de <conta corrente> em um <cash-dispenser>” contido no Sistema de *Cash-dispenser* do Subdomínio de Automação Bancária do domínio Área Bancária, pode ser apresentado como exemplo sob a forma:

$Obj_{Gen1} = \{ \text{Área Bancária, Retirar valor de <conta corrente> em um <cash-dispenser>, Retirar, valor, <conta corrente>, <cash-dispenser> }.$

- Cenário:

- 1) Solicitar número da <conta corrente>
- 2) Validar número da <conta corrente>
- 3) Solicitar senha da <conta corrente>
- 4) Validar senha da <conta corrente>
- 5) Solicitar valor a ser retirado
- 6) Se valor \leq (saldo <conta corrente> + limite) então efetivar o saque.

Senão, exibir mensagem

Tanto no objetivo quanto nos cenários, foram identificadas partes que irão variar de um objetivo para outro. Neste exemplo, o objetivo pode ser reutilizado para a retirada de valor em uma conta poupança, bastando para isso, que se use <conta poupança> como parâmetro.

Outros objetivos e cenários deverão ser trabalhados de forma semelhante para que se tenha um número razoável de requisitos parametrizados que facilitem a sua reutilização.

3.4 Reutilização

Nesta seção será descrita como efetivar a reutilização de forma não automatizada. A reutilização com o auxílio de uma ferramenta de software será apresentada em uma seção posterior.

Uma vez identificados, descritos e formatados os objetivos genéricos da aplicação em desenvolvimento, estes devem ser colocados em um repositório. O repositório consiste de um documento de texto que pode ser manipulado por um processador de texto. No caso do uso de uma ferramenta de software, o repositório é uma base de dados.

No caso da reutilização não automatizada, o repositório deve ser mantido pelo engenheiro de requisitos para propiciar a reutilização, pois é neste documento que os objetivos genéricos ficarão disponibilizados para serem consultados, selecionados e reutilizados. Os objetivos que forem reutilizados devem ser transportados para um segundo documento que irá conter todos os objetivos da aplicação em desenvolvimento. Após a transferência estes serão adaptados à nova aplicação.

Considere que o repositório contenha alguns objetivos genéricos já adicionados. A seguir serão descritos os passos para se efetuar a consulta, seleção e reutilização destes objetivos.

▪ *Consulta e Seleção dos objetivos genéricos*

Ao se enumerar os objetivos da aplicação em desenvolvimento, o engenheiro de requisitos pode consultar o repositório para verificar a existência de requisitos genéricos que possam ser reutilizados pela aplicação.

A consulta pode ser agilizada através de uma busca pelos Verbo e Objeto Principais contidos nos objetivos.

▪ *Reutilização dos objetivos genéricos*

Uma vez encontrados os objetivos genéricos, estes devem ser analisados e avaliados com relação ao seu reuso pela nova aplicação.

Para que um objetivo genérico seja reutilizado por outro sistema, é suficiente que se utilize de uma cópia deste e ajuste os seus parâmetros para que o mesmo se adeque à aplicação em desenvolvimento. Outra forma de adaptar o objetivo à sua nova realidade é fazer a reutilização parcial dos cenários que o descreve. Isto não descarta, entretanto, a possibilidade de reutilização total dos cenários.

Considere como exemplo o objetivo “Retirar valor de uma <conta corrente> em um <terminal de caixa>” de Sistema de Caixa de um banco. Ele pode ser reutilizado pelo mesmo sistema como “Retirar valor de uma <conta poupança> em um <terminal de caixa>”, ou ainda como “Retirar valor de uma <carteira de crédito> em um <terminal de caixa>” por outro sistema de mesmo domínio, como um Sistema de Empréstimo.

Os cenários que descrevem o objetivo que será reutilizado podem e devem ser também parametrizados. Assim sendo, não há a necessidade de se descrever novamente todo o cenário, mas apenas substituir os parâmetros. No caso do objetivo acima, os cenários que o descrevem podem ser totalmente reutilizados nos sistemas do exemplo.

O seguinte trecho do documento com os objetivos do Sistema de Caixa ilustra a reutilização do objetivo do exemplo dado:

Objetivo: Retirar valor de <conta corrente> em um <terminal de caixa>

Obj_{Gen1} = {Área Bancária, “Retirar valor de <conta corrente> em um <terminal de caixa>”, Retirar, valor, <conta corrente>, <terminal de caixa>}

- Cenário:

- 1) Solicitar número da <conta corrente>
- 2) Validar número da <conta corrente>
- 3) Solicitar senha
- 4) Validar senha
- 5) Solicitar valor do saque
- 6) Se valor <= (saldo <conta corrente> + limite) então
Deduzir valor do saldo da <conta corrente>
Senão
Exibir mensagem e interromper a transação.

Objetivo: Retirar valor de uma <conta poupança> em um <terminal de caixa>

Reutilizar $\text{Obj}_{\text{Gen1}} = \{\text{Área Bancária, "Retirar valor de <conta poupança> em um <terminal de caixa>"}, \text{Retirar, valor, <conta poupança>, <terminal de caixa>}\}.$

Como pode ser notado, o objetivo genérico reutilizado precisa apenas ser adaptado à nova aplicação no documento, reduzindo o esforço do engenheiro. Percebe-se, no entanto, que a manutenção e manipulação do documento pode ser comprometida se ele for muito grande. Assim, foi proposta a maneira semi-automatizada de se efetivar o método através do protótipo FROBJER, descrita na seção 5.

Um exemplo que mostre de forma detalhada a aplicação do MROBJER é apresentado na seção que se segue.

3.5 Comparativo com outras abordagens

Verifica-se que, apesar das abordagens MRAM, FORE, RARE, *Scenario Method Chunks* e MROBJER se utilizarem de características comuns para efetivarem a reutilização, os artefatos a serem reutilizados mudam de uma proposta para outra. Como exemplo tem-se que o método RARE se propõe a reutilizar requisitos, o *Scenario Method Chunks* reutiliza blocos de cenários e o MROBJER objetivos.

Neste contexto, observa-se que o MROBJER permite a possibilidade da reutilização total ou parcial de cenários que descrevem os objetivos genéricos, além da parametrização. Isto flexibiliza o método no sentido de promover uma maior adaptabilidade do seu artefato de reuso à nova aplicação. Outra característica que deve ser ressaltada é a reutilização Interdomínios.

4. Exemplo

Tomando-se como exemplo o domínio da Área Bancária, tem-se a seguir a sua representação em forma de árvore conforme estrutura mapeada nas seções 3.1 e 3.2. Em seguida, serão descritos de forma textual alguns objetivos e cenários. Ressalta-se que a figura que apresenta o mapeamento exibe a numeração de referência aos objetivos listados no trecho do repositório, como por exemplo A, A.1, B, B.1, etc.

A figura 3.4 exibe a estrutura do domínio Área Bancária com três subdomínios ligados ao domínio: “Automação de Agências”, “Internet” e “Sistemas Corporativos”. O nível seguinte contém os sistemas direcionados a atividades de negócio mais específicas dentro destes. Percebe-se também a ocorrência de agregações funcionais e objetivos semelhantes sendo candidatos para o reuso.

Sabe-se que são vários os dispositivos existentes para que o cliente realize suas transações financeiras em uma agência bancária. Cada dispositivo contém uma gama de programas específicos com operações semelhantes que atendem ao cliente do banco.

Os sistemas que compõem o nível aplicações possuem agregações funcionais que contém operações com aspectos semelhantes, como por exemplo a agregação

funcional de Transações Financeiras. Nesta agregação funcional as operações de depósito e retirada são realizadas de maneiras diferentes.

Exemplificando melhor, uma operação de retirada no *cash-dispenser* é realizada pelo próprio cliente, isto é, o sistema espera receber uma solicitação de “login” do cliente, validando assim o número da conta. Em contrapartida, a mesma operação em um terminal de caixa é realizada por um funcionário do banco, o caixa. Neste caso é o caixa quem solicita o “login” do sistema informando seu código de acesso e sua senha. Somente após o “login” é que o saque poderá ser efetivado pelo caixa com a validação do número da conta e da senha do cliente. Esta senha é digitada pelo mesmo em um periférico denominado de *pin-pad*.

Os objetivos genéricos e seus respectivos cenários são listados de forma textual no repositório. Uma vez inseridos no repositório, os objetivos genéricos são associados em definitivo à nova aplicação sendo inseridos também no documento que irá conter os objetivos desta aplicação.

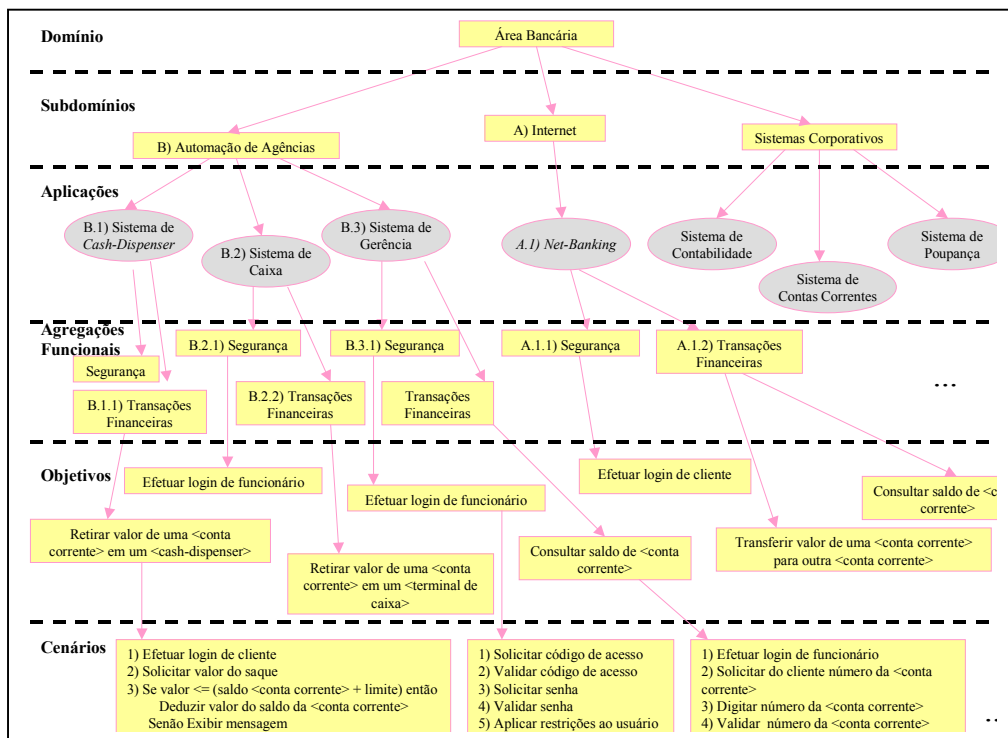


Figura 3.4 Mapeamento do domínio Área Bancária

Caso existam objetivos genéricos de aplicações pertencentes a outros domínio, o engenheiro de requisitos deve buscar outros objetivos para reutilização. Se a busca for bem sucedida, os objetivos encontrados devem ser analisados. Aqueles objetivos que forem selecionados devem ser inseridos no documento de objetivos da aplicação em desenvolvimento e adaptados.

A seguir será verificada a reutilização intersistemas através da relação de alguns dos objetivos genéricos do domínio da Área Bancária. Na relação contida no repositório encontra-se aplicada a formatação dos objetivos proposta na seção 3.3. Assim sendo, é visualizada a maneira na qual os objetivos estarão no repositório que os disponibilizará para a reutilização.

Destaca-se no trecho do repositório listado a seguir o fato das similaridades ocorridas nas agregações funcionais das aplicações relacionadas.

- *Domínio: Área Bancária*

A) Internet(Subdomínio)

A.1) Net-Banking(Aplicação)

A aplicação está disponível para clientes do banco em qualquer ponto da internet. O usuário poderá efetuar pagamentos, consultas e transferências de valores de sua conta corrente ou poupança para outra conta corrente ou poupança deste ou de outro banco. A transferência será efetivada se houver saldo, incluindo limite em caso de conta especial.

Os serviços seguintes estão geralmente disponíveis neste tipo de aplicação:

A.1.1) Segurança(Agregação Funcional)

a) Objetivo: Efetuar login de cliente

Obj_{Gen1} = {Área Bancária, “Efetuar login de cliente”, Efetuar, login}

- Cenário:

- 1) Solicitar número da <conta corrente>
- 2) Validar número da <conta corrente>
- 3) Solicitar senha da <conta corrente>
- 4) Validar senha da <conta corrente>
- 5) Aplicar restrições ao usuário

A ação “Solicitar número da <conta corrente>” consiste de ações mais específicas, que podem ser listadas como cenário primitivo:

- a. Exibir mensagem solicitando número da <conta corrente>;
- b. Aguardar evento de Ok ou enter;

A.1.2) Transações Financeiras(Agregação Funcional)

Observa-se que, para realizar as operações desta agregação funcional, o cliente deve estar previamente “logado” no sistema. Neste caso o objetivo “Efetuar ‘login’ de cliente” poderia ser considerado como sendo parte integrante dos cenários que descrevem os objetivos desta agregação funcional.

a) Objetivo: Transferir valor de uma <conta corrente> para outra <conta corrente>

Obj_{Gen2} = {Área Bancária, “Transferir valor de uma <conta corrente> para outra <conta corrente>”, Transferir, valor, <conta corrente>, <conta corrente>}

- Cenário:

- 1) Solicitar Banco, Agência e número da <conta corrente> Destino
 - 2) Solicitar valor a ser transferido
 - 3) Se valor <= (saldo <conta corrente> + limite) então
Deduzir valor do saldo da <conta corrente>
Criar a transação de depósito na <conta corrente> a ser favorecida
- Senão
Exibir mensagem e interromper a transação.

b) Objetivo: Consultar saldo de <conta corrente>

Obj_{Gen3} = {Área Bancária, “Consultar saldo de <conta corrente>”, Consultar, saldo, <conta corrente>}

- Cenário:

- 1) Solicitar número da <conta corrente>
- 2) Exibir informações solicitadas

B) Automação de Agências(Subdomínio)

O sistemas da Automação de Agências podem ser utilizados por mais de um tipo de usuário: clientes do banco e funcionários. No segundo caso há diversos níveis de autorização dependendo do cargo: caixas, gerentes, etc.

O cliente obtém informações e realiza suas operações de formas diferenciadas, seja por um terminal de caixa(operado pelo caixa), *cash-dispenser*(operado pelo próprio cliente) ou terminais de consultas(operados por atendentes e gerentes).

B.1) Sistema de Cash-Dispenser(Aplicação)

Observa-se novamente o fato de que o objetivo “Efetuar ‘login’ de cliente” poderia ser considerado como sendo parte integrante dos cenários que descrevem os objetivos desta agregação funcional.

B.1.1) Transações Financeiras(Agregação Funcional)

a) Objetivo: Retirar valor de <conta corrente> em um <cash-dispenser>.

Obj_{Gen4} = {Área Bancária, “Retirar valor de <conta corrente> em um <cash-dispenser>”, Retirar, valor, <conta corrente>, <cash-dispenser>}

- Cenário:

- 1) Solicitar valor do saque
- 2) Se valor <= (saldo <conta corrente> + limite) então
Deduzir valor do saldo da <conta corrente>
Entregar o valor para o cliente

Senão

Exibir mensagem e interromper a transação.

B.2) Sistema de Caixa(Aplicação)

B.2.1) Segurança(Agregação Funcional)

a) Objetivo: Efetuar login de funcionário

Obj_{Gen5} = {Área Bancária, “Efetuar login de funcionário”, Efetuar, login}

- Cenário:

- 1) Solicitar código de acesso
- 2) Validar código de acesso
- 3) Solicitar senha
- 4) Validar senha
- 5) Aplicar restrições ao usuário conforme código de acesso.

Obs.: O funcionário terá habilitadas funções pertinentes ao seu cargo dentro da agência bancária. Ex.: Caixa, Supervisor, Gerente, etc.

B.2.2) Transações Financeiras(Agregação Funcional)

a) Objetivo: Retirar valor de <conta corrente> em um <terminal de caixa>

Obj_{Gen6} = {Área Bancária, “Retirar valor de <conta corrente> em um <terminal de caixa>”, Retirar, valor, <conta corrente>, <terminal de caixa>}

- Cenário:

- 1) Solicitar número da <conta corrente>
- 2) Validar número da <conta corrente>
- 3) Solicitar senha
- 4) Validar senha
- 5) Solicitar valor do saque
- 6) Se valor <= (saldo <conta corrente> + limite) então
Deduzir valor do saldo da <conta corrente>
Senão
Exibir mensagem e interromper a transação.

B.3) Sistema de Gerência(Aplicação)

B.3.1) Segurança(Agregação Funcional)

a) Objetivo: Efetuar login de funcionário

Reutilizar **Obj_{Gen5}** = {Área Bancária, “Efetuar login de funcionário”, Efetuar, login}.

Uma vez listados objetivos assinalados no exemplo do domínio da Área Bancária, observa-se a reutilização do Objetivo Genérico Obj_{Gen5} em sistemas de um mesmo domínio. Trata-se da reutilização Intersistemas. Nestes casos não se fez necessário descrever novamente o cenário, mas apenas referenciar o objetivo genérico reutilizado no repositório.

Para se listar os objetivos genéricos a serem reutilizados no documento que contém os objetivos da aplicação em desenvolvimento, basta copiá-los e efetivar-se as adaptações necessárias.

5. FROBJER

A reutilização agora será realizada com o auxílio da ferramenta de software. Considere a que algumas estruturas na base de dados da ferramenta servirão de apoio para a reutilização de objetivos por novas aplicações.

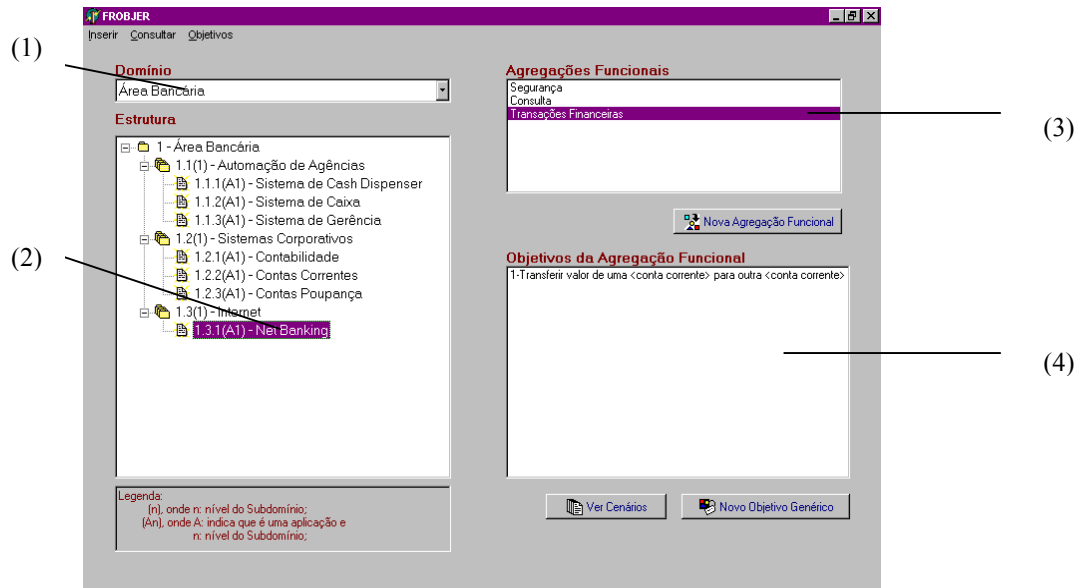


Figura 5.1 Visualização dos Domínios Mapeados

A figura 5.1 mostra como a estrutura é visualizada na ferramenta. O domínio desejado poderá ser selecionado através de um *combobox*(1) e a sua estrutura de árvore até o nível de Aplicações será mostrada pelo componente *TreeView*(2) da tela. Ao se selecionar uma aplicação, suas agregações funcionais poderão ser visualizados em uma lista(3). Da mesma forma, ao se selecionar uma agregação funcional, os objetivos associados a esta agregação funcional serão listados em(4).

Considere também que o engenheiro de requisitos tenha mapeado alguns objetivos e cenários genéricos e os tenha acrescentados à base de dados da ferramenta. Neste caso, ele poderá consultar se já existe na base algum objetivo genérico que possa ser reutilizado pela aplicação em desenvolvimento.

A consulta poderá ser feita na tela da figura 5.2. O usuário da ferramenta pode consultar os objetivos genéricos por domínio marcando a opção “Por Domínio” e selecionando o domínio desejado(1). Pode-se ainda se restringir a consulta pelos Verbo e Objeto Principais(2).

Os objetivos encontrados na consulta serão exibidos em (3) para que o engenheiro os analise e decida se irá ou não aproveitá-los. Ao se selecionar um objetivo, os seus cenários serão exibidos em (5). Ao se selecionar um cenário composto os seus cenários primitivos serão exibidos em (6).

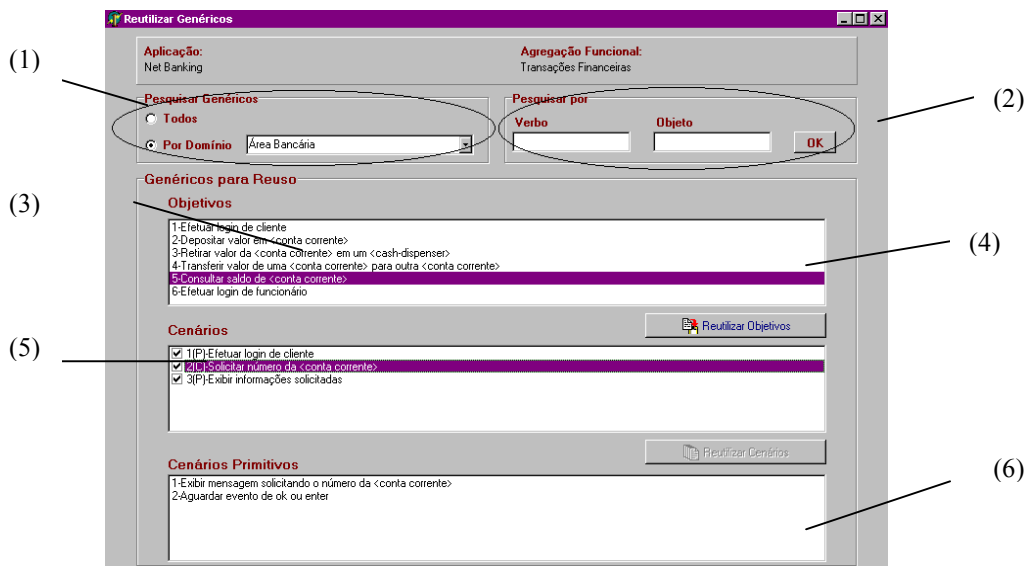


Figura 5.2. Tela usada para Reutilização

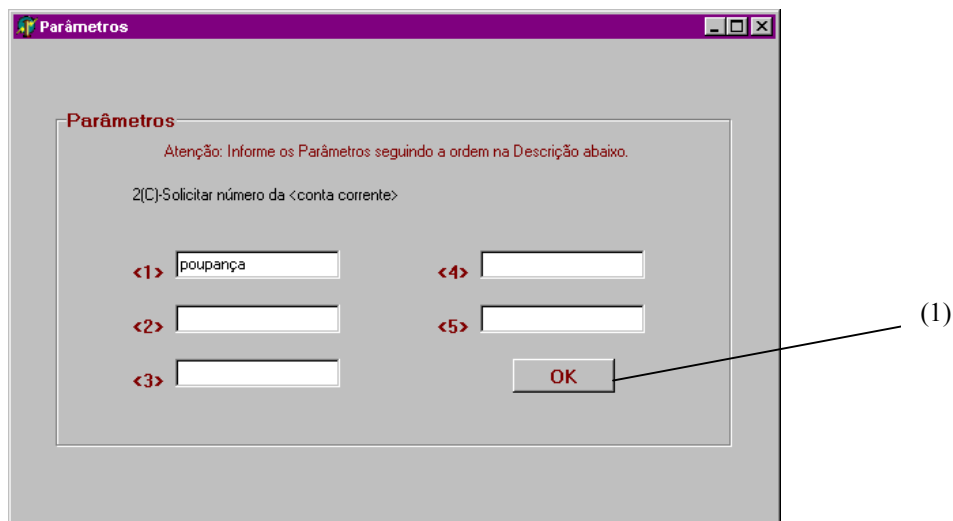


Figura 5.3 Tela para passagem de parâmetros

Para a reutilização do objetivo selecionado, deve-se marcar os cenários a serem reutilizados e clicar-se no botão “Reutilizar Objetivos”(4). Quando houverem parâmetros, uma tela será exibida para que estes sejam informados.

A figura 5.3 mostra a tela na qual os parâmetros são informados. Nela, o usuário da ferramenta só precisa informar os parâmetros e clicar no botão “OK”(1).

Uma vez efetuada a passagem de parâmetros, irá ocorrer uma reutilização estática, isto é, tudo o que foi reutilizado será associado à nova aplicação com os parâmetros condizentes a ela.

A figura 5.4 mostra a tela na qual o usuário da ferramenta poderá visualizar os objetivos que estão associados às agregações funcionais de uma aplicação.

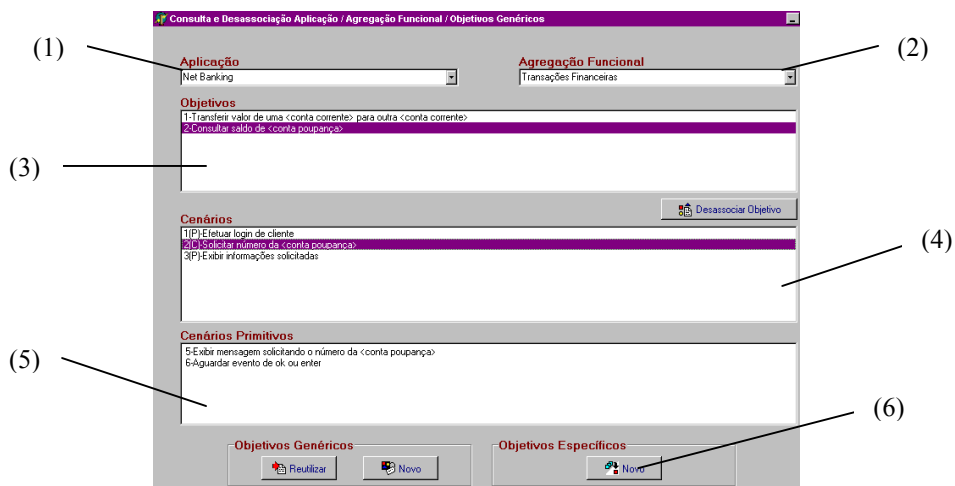


Figura 5.4 – Visualização dos Objetivos

Nesta tela, o engenheiro poderá visualizar os objetivos(3) e cenários da aplicação selecionada em (1) conforme a agregação funcional selecionada em (2). Ao se selecionar um objetivo, seus cenários são exibidos em (4). Caso o cenário selecionado seja composto, os seus cenários primitivos são visualizados em (5).

Os objetivos mais específicos também podem ser associados à aplicação, através de (6), para complementar o Documento contendo os Objetivos da Aplicação em Desenvolvimento que será gerado sempre que for necessário.

6. Conclusão

O artigo abordou a reutilização de objetivos como forma de promover a melhora da produtividade e o aumento da qualidade no desenvolvimento de software. A princípio, foram citadas algumas abordagens de reutilização que serviram de base para a elaboração de alguns passos que propõem a reutilização de objetivos na Engenharia de Requisitos.

O Método para Reutilização de Objetivos na Engenharia de Requisitos(MROBJER) foi apresentado no artigo.

Inicialmente foi mostrada a estrutura que será mapeada para propiciar a identificação de objetivos para posterior reutilização. Os objetivos são descritos através de cenários e poderão ser reutilizados sempre que a aplicação em desenvolvimento os contiver em alguma ou algumas de suas agregações funcionais.

Os objetivos e seus cenários são apresentados de forma codificada para propiciar melhor busca no repositório. Alguns objetivos são genéricos, isto é, podem constar de mais de uma aplicação. Parâmetros são identificados nestes objetivos e cenários para propiciar uma maior flexibilidade durante a sua reutilização por novas aplicações.

Um exemplo prático foi mostrado com o intuito de se visualizar o mapeamento da estrutura e a reutilização de alguns objetivos. Para tanto foi também apresentado o FROBJER, um protótipo de uma ferramenta que viabiliza a reutilização dos objetivos genéricos e cenários dentro e fora do Domínio exemplificado. Com o FROBJER o processo do MROBJER torna-se semi-automático e, conseqüentemente, mais fácil e prático.

A reutilização de objetivos é uma poderosa ferramenta para o processo de desenvolvimento de software se for aplicada de forma apropriada e com a cautela necessária no seu processo. Critérios que objetivem o sucesso da reutilização sempre devem ser considerados[6].

7. Referências

- [1] ANTÓN, Annie I. Goal Identification and Refinement in the Specification of Information Systems. Ph.D. Thesis. Georgia Institute of Technology. June 1997.
- [2] Ariane 5, Flight 501 Failure, Inquiry Board report, www.esrin.esa.it/tidc/Press/Press96/ariane5rep.html.
- [3] CYBULSKI, Jacob L. *RARE IDIOM: An Overview*. <http://www.dis.unimelb.edu.au/staff/jacob/>. 1995.
- [4] DARDENE, A. LAMSWEERDE, A. and FICKAS, S. Goal-Directed Requirements Acquisition” in *The Science of Computing Programming*, 20, p.3-50. 1993.
- [5] LAM, W. A Case-Study of Requirements Reuse Through Product Families. Rolls-Royce Technology Centre, 1997.
- [6] LAM, Wing, McDERMID, J. A., VICKERS, A. J. Ten Steps Towards Systematic Requirements Reuse. *Requirements Eng.* 1997. Springer-Verlag London Limited.
- [7] LAM, W. Achieving Requirements Reuse: A Domain-Specific Approach from Avionics. Rolls-Royce Technology Centre. The University of York. Heslington. York, UK. 1997.
- [8] LAM, Wing; JONES, S. And BRITTON, C. Technology Transfer for Reuse: A Management Model and Process Improvement Framework. *Proceedings of the 1998 International Conference on Requirements Engineering(ICRE'98)*. Published by the IEEE Computer Society.
- [9] MAIDEN, Neil. Analogy as a paradigm for specification reuse. *Software Engineering journal*, January 1991.
- [10] MAIDEN, N.A.M. and SUTCLIFFE, A.G. Analogical retrieval in reuse-oriented requirements engineering. *Software Engineering journal*, September 1996.
- [11] MANNION, Mike, KAINDL, Hermann, Wheadon, Joe. Reusing Single System Requirements from Application Family Requirements. 1998.

- [12] ROLLAND, C., PLIHON, V., RALYTÉ, J. Specifying the Reuse Context of Scenario Method Chunks. CREWS Report Series. Appeared in the Proceedings of the 10th International Conference CaiSE'98, B. Lecture Notes in Computer Science 1413, Pernici. C. Thanos(Eds), Springer. Pisa, Italy, June 1998.
- [13] ROLLAND, C., SOUVEYET, C., ACHOUR, C. B. Guiding Goal Modelling Using Scenarios. CREWS Report Series 98 - 27. Submitted to TSE special issue on **scenario management**, 1998.
- [14] SOMMERVILLE, Ian and KOTONYA, Gerald. Requirements Engineering: Processes and Techniques. John Wiley & Sons, 1997.
- [15] SOMMERVILLE, Ian and SAWYER, Pete. Requirements Engineering: A good practice guide. John Wiley & Sons, 1998.
- [16] TAWBI, M.; SOUVEYET, C.; ROLLAND, C. L'ECRITOIRE a tool to support a goal-scenario based approach to requirements engineering. CREWS Report Series. Submitted to Information and Software Technology Journal (ed. M.Shepperd) Elsevier Science Publishers. 1998.
- [17] ZANLORENCI, Edna Pacheco, BURNETT, Robert Carlisle. Engenharia de Requisitos: Processos e técnicas no contexto organizacional. BateByte Edição 77/98.