

Casos de (Re)Uso: Uma Abordagem para Reuso de Software Interativo Dirigida por Casos de Uso e Padrões Concretos de Interação

Augusto Abelin Moreira ^{1,2}

¹ *Companhia de Processamento de Dados
do Estado do Rio Grande do Sul
(PROCERGS)
Caixa Postal 236 – 90010-340
Porto Alegre – RS – Brasil
+55 51 3210-3100
augusto@procergs.rs.gov.br*

Marcelo Soares Pimenta ²

² *Instituto de Informática
Universidade Federal do Rio Grande do
Sul (UFRGS)
Caixa Postal 15.064 – 91501-970
Porto Alegre – RS – Brasil
+55 51 3316 6814
{aamoreira ,mpimenta}@inf.ufrgs.br*

Abstract

This paper aims to present an use case driven software reuse approach for interactive systems, integrating – by means of some aspects of use case life cycle (from modeling to implementation) - several well-known reuse concepts and techniques like use case patterns, interaction patterns and design patterns. The approach focuses on how to promote user interface reuse integrated to reuse of application-domain related software artifacts and by means of concrete interaction patterns (CIPs) usage. A CIP extends the usual interaction pattern documentation describing interactions and user interface behaviour through User Interface Description Language (UIDL), allowing reuse and implementation of an abstract user interface to different platforms and technologies.

Keywords: use case driven reuse, reified use case pattern, user interface reuse, software reuse, use case presentation, interaction pattern, concrete interaction pattern, User Interface Description Language, UIML.

Resumo

O objetivo deste artigo é apresentar uma abordagem de reuso de artefatos de software interativo dirigida por casos de uso, integrando – por meio de alguns aspectos do ciclo de vida de casos de uso (da modelagem à implementação) – vários conceitos e técnicas de reuso bem conhecidos como padrões de casos de uso, padrões de interação e padrões de projeto.

Um dos focos da abordagem é como promover o reuso de interface com o usuário integrado ao reuso dos artefatos orientados ao domínio da aplicação através da especificação e utilização de modelos de reificação de padrões de interação, os quais denominamos: padrões concretos de interação (concrete interaction patterns – CIPs). Um padrão concreto de interação estende a documentação de um padrão de interação com a descrição das suas interações em uma linguagem de descrição de interface com o usuário (UIDL). Desta forma, explicitam-se os elementos de IU necessários para realizar a interação bem como os seus comportamentos, o que torna possível reusar e derivar implementações para as mais diversas plataformas e tecnologias.

Palavras-chaves: reuso dirigido por casos de uso, padrão de caso de uso reificado, reuso de interface com o usuário, reuso de software, apresentação de caso de uso, padrão de interação, padrão concreto de interação, Linguagem de Descrição de Interface com o Usuário, UIML.

1. Introdução

Embora o reuso seja uma das práticas mais indicadas da Engenharia de Software, a sua aplicação em IHC é ainda pouco difundida, talvez pela crença muito presente entre os designers de que reuso de soluções de outrem é um sinal da diminuição da sua criatividade. No entanto, o reuso em software é visto por muitos autores como uma das mais prováveis “balas de prata” para solucionar os problemas do desenvolvimento de sistemas [8].

Sua adoção na prática de design de interação por profissionais de IHC potencialmente permitiria uma

maior qualidade deste design (reuso de uma solução bem sucedida em teoria diminui a probabilidade de ocorrência de erros) e uma maior produtividade (reuso de elementos libera os designers para tratar problemas para os quais ainda não há solução) da equipe.

Mas a adoção de práticas de reuso não é implantada facilmente em uma organização. Construir artefatos reusáveis requer informações de identificação, extração, organização e representação de uma maneira que seja fácil de entender e manipular. Encontrar os artefatos reusáveis que possam ser (re)utilizados para o desenvolvimento de um novo sistema pode ser muitas vezes mais difícil e trabalhoso do que o desenvolvimento deste novo sistema. De fato, software para ser reusável e reusado deve ser projetado, documentado e implementado para este fim de reuso [25].

Embora a tecnologia disponível já habilite os designers a praticar reuso, isto não implica que o reuso ocorrerá. É preciso uma estratégia de reuso (com conjunto de conceitos e ferramentas associados) que propicie o reuso em diferentes níveis de abstração. Tal estratégia deve focar em reuso durante todo o ciclo de desenvolvimento, prover suporte à reutilização de artefatos (development with reuse) e à produção de artefatos reusáveis (development for reuse), ser facilmente integrável a outros métodos e técnicas de desenvolvimento utilizados e ser implementada por meio de ferramentas (também integráveis a ferramentas correntemente utilizadas) [20].

Desenvolver interfaces usando um processo de reuso possui basicamente 3 etapas: 1) realizar buscas em algum repositório de elementos reusáveis (assets), 2) selecionar os elementos mais adequados e 3) adaptá-los ao novo contexto específico de uso. Hoje, os assets de IHC mais comuns presentes em um repositório são os objetos de interação - num nível de implementação - e os padrões de interação (interaction patterns) - num nível mais abstrato.

Um padrão de interação captura conhecimentos comprovados de projeto de interfaces e é descrito em termos de um problema, um contexto e uma solução, no mesmo estilo dos padrões GoF [12]. Em particular, um padrão de interação deve estar focado em soluções que melhorem a usabilidade do sistema em uso [29].

O objetivo deste artigo é apresentar uma abordagem para reuso de interfaces com o usuário (IUs) através da especificação e utilização de modelos de reificação de padrões de casos de uso e padrões de interação. O artigo está estruturado da seguinte forma: a seção 2 apresenta uma revisão sobre os conceitos fundamentais dos padrões de interação e a sua aplicação no projeto de sistemas interativos e a seção

3 resume os principais conceitos envolvidos na descrição de IUs através de linguagens baseadas em XML. A seção 4 apresenta o conceito de padrões concretos de interação, os ilustra através de um exemplo e discute como podem ser utilizados em práticas de reuso. O projeto de IU em um processo de desenvolvimento dirigido por casos de uso e o papel dos padrões de casos de uso neste processo são discutidos nas seções 5 e 6, respectivamente. Na seção 7, apresentamos o fio condutor da nossa abordagem de reuso que é o padrão de caso de uso reificado. O processo de identificação de artefatos de IU reusáveis integrado a um processo de desenvolvimento OO é apresentado na seção 8. Finalmente, a seção 9 é a conclusão.

2. Reusando padrões de interação no projeto de interfaces com o usuário

Os primeiros textos relacionados especificamente ao desenvolvimento de IU associados a padrões de interação (*interaction patterns* - *IP*) começaram a surgir a partir de 1994 [18]. Entretanto, somente a partir dos trabalhos de Borchers [3, 4], Sutcliffe e Carroll [21] e Tidwell [23] é que se deu um incremento da aplicação de padrões na área de Design de Interação.

Os padrões raramente existem de forma isolada. As linguagens de padrões (*pattern languages*) reúnem padrões que se relacionam e se complementam entre si, disponibilizando para o projetista de interfaces um acervo de idéias comprovadas de interação que podem ser aplicadas no seu projeto de forma consistente [24]. Entretanto, as linguagens de padrões ainda não são utilizadas de forma sistemática nos projetos de interface talvez por existir uma multiplicidade de linguagens (ver p.ex: 9, 13, 14, 23 e 28) muitas vezes incompatíveis entre si [10].

No estágio atual, os padrões de interação têm um potencial muito grande para serem úteis no reuso de projeto de interfaces. No entanto, para isto ocorrer de forma efetiva, algumas questões precisam ser endereçadas. Uma delas se refere à padronização do formato. Embora existam propostas de padronização do formato do padrão (entre elas [11] e [16]), seu objetivo é estruturar a documentação do padrão e não a descrição da IU que implementa a sua solução. Outra questão está relacionada à especificação das interfaces que implementam o padrão: não existem descrições precisas dos elementos de interação e como eles devem se comportar numa implementação da solução proposta pelo padrão de interação. De fato, pode haver diferentes implementações do mesmo padrão com

diferenças de comportamento entre elas. Para ilustrar isto, considere uma IU que implementa o padrão de interação “Parts Selector” da coleção de Van Welie [22] apresentada na Figura 1.

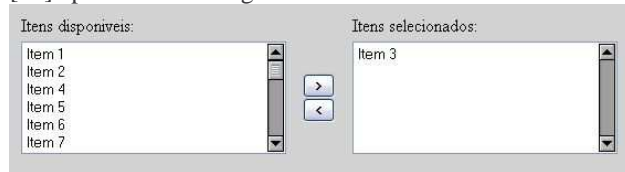


Figura 1. Uma IU do padrão de interação “Parts Selector”

A documentação deste padrão define claramente o trinômio problema-contexto-solução bem como apresenta exemplos de uso deste padrão. Entretanto, esta documentação não é suficiente para que um projetista de interface possa entender o funcionamento da interface e implementá-la de uma forma consistente com as demais implementações da mesma interface. Muitas questões ficam em aberto. Por exemplo: um item a ser adicionado na caixa de listagem “Itens selecionados” será inserido no final da lista ou no lugar que ocupará em uma lista ordenada? Os itens que são movimentados da caixa “Itens disponíveis” para a caixa “Itens selecionados” são removidos ou permanecem na caixa “Itens disponíveis”? Podem-se selecionar múltiplos itens ou somente um de cada vez? Como funciona a interface com o uso do teclado? Os botões de movimentação ficam habilitados quando não há item selecionado ou uma caixa de listagem fica vazia?

Neste artigo, propomos que a documentação do padrão de interação seja estendida por um conjunto de descrições das IUs que o implementam. Desta forma, as dúvidas de funcionamento da IU são dirimidas e - o que é mais importante - dá consistência às IUs pelo fato de poderem ser implementadas da mesma maneira.

Com a descrição das IUs, vislumbram-se oportunidades de reuso não só das (boas) idéias do padrão de interação, mas também de inúmeros artefatos que vão desde reuso das descrições de IUs nos projetos de interface até o reuso direto de implementações destas IUs.

3. Descrevendo as interfaces com o usuário

Uma IU pode ser descrita através de hierarquias de elementos de interação e como eles se comportam. Estas hierarquias podem ser representadas em vários níveis de abstração que vão desde descrições concei-

tuais até as implementações da IU em uma determinada tecnologia.

Dentre as várias abordagens de descrição existentes, a nossa abordagem foi inspirada no modelo de desenvolvimento de IUs para aplicações interativas multi-contexto, o CRF - Cameleon Reference Framework [5]. O Cameleon Framework define modelos que suportam os vários níveis de descrições de IU (que no CRF são chamados de passos desenvolvimento), bem como os processos para a transformação de uma descrição de IU em outra mais concreta (processo de reificação); em outra mais abstrata (processo de abstração); ou em outra em um contexto de uso diferente num mesmo nível de abstração (processo de tradução). A Figura 2 ilustra os dois passos de desenvolvimento do CRF com os processos que serão usados na nossa abordagem.

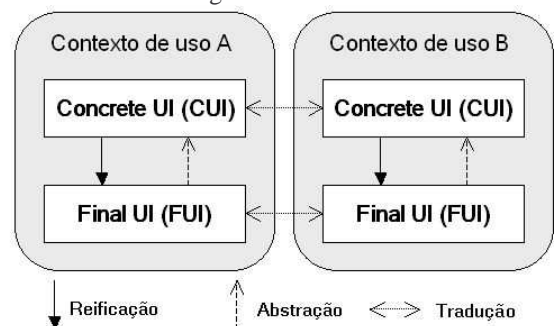


Figura 2. Os dois níveis mais “concretos” do Cameleon Framework.

Uma *Final UI (FUI)* é uma IU operacional, ou seja, qualquer IU executando em uma plataforma de computação específica. Os artefatos de uma FUI são os códigos-fonte ou os códigos-objeto que implementam a IU.

Uma *Concrete UI (CUI)* é uma representação abstrata de uma FUI de forma que seja independente de qualquer plataforma computacional ou de toolkit de desenvolvimento. Uma CUI consiste de uma decomposição hierárquica de CIOs de um dado contexto de uso. Um CIO (*Concrete Interaction Object*) é definido como qualquer entidade de IU que o usuário pode perceber (tais como texto, imagem e som) e/ou manipular (tais como botões, caixas de listagem ou menus) [27].

O artefato mais usado para descrever uma CUI é através de uma Linguagem de Descrição de IU (*User Interface Description Language - UIDL*). As UIDLs são linguagens baseadas em XML que permitem descrever as IUs de forma independente de tecnologia e de ambiente de desenvolvimento integrado (IDE). Atualmente, existem várias propostas de UIDL tais como: AUIML, UIML, UsiXML, XIML e XUL, para

citar algumas. Nós optamos por descrever as CUIs em UIML [26] por dois motivos básicos: porque é um dos padrões abertos de UIDL mais utilizados atualmente e porque tem grande flexibilidade para definir quaisquer conjuntos de elementos de interação (widgets) através dos *vocabulários*.

A Figura 3 apresenta uma CUI escrita em UIML para a interface apresentada na Figura 1.

```
<?xml version="1.0"?>
<!DOCTYPE uiml PUBLIC "-//Harmonia//DTD UIML 3.0a
Draft/EN" http://uiml.org/dtds/UIML3_0a.dtd>
<uiml>
<peers> <presentation
base="GenericJH_1.2_Harmonia_1.0"/></peers>
<interface>
<structure>
<part class="G:TopContainer" id="top">
<style> ... </style>
<part class="G:TextBox" id="textDisp">
<style>
<property name="g:textboxtype">textarea</property>
<property name="g:text">Itens disponiveis:</property>
<property name="g:editable">>false</property>
</style>
</part>
<part class="G:List" id="listaDisp"> ... </part>
<part class="G:Button" id="botaoAdiciona" >
<style>
<property name="g:text">&gt;</property>
<property name="g:buttontype">push</property>
...
</style>
</part>
<part class="G:Button" id="botaoRemove" > ... </part>
<part class="G:TextBox" id="textSelec"> ... </part>
<part class="G:List" id="listaSelec"> ... </part>
</part>
</structure>
<behavior> ... </behavior>
</interface>
</uiml>
```

Figura 3. Extrato de uma CUI para o padrão “Parts Selector”

4. Ligando padrões de interação a linguagens de descrição de interfaces com o usuário

Na prática, um padrão de interação precisa ser implementado utilizando algum conjunto de objetos de interação disponível em alguma plataforma e descrito em alguma linguagem. Obviamente, um mesmo padrão de interação pode ser mapeado para diferentes formas de interação (modalidades) e de implementação (tecnologias). A nossa proposta busca uma abordagem de reuso de IHC ao aglutinar os conceitos de padrão de interação com os de descrição de IUs no que chamamos de *padrão concreto de interação*.

Um padrão concreto de interação (*concrete interaction pattern – CIP*) é um conjunto de artefatos que

descrevem e implementam uma IU de um padrão de interação. O CIP estende a documentação do padrão de interação com a agregação de modelos notacionais para a reificação das IUs. Estes modelos estão estruturados em um CIP conforme apresentado na Figura 4.

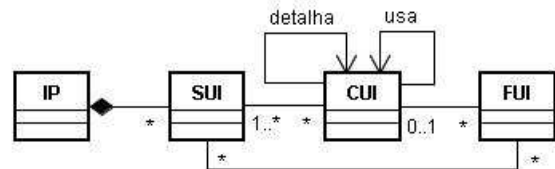


Figura 4. Estrutura de um padrão concreto de interação.

Um padrão de interação (IP) pode conter várias SUIs (Sketched UIs). As SUIs nada mais são do que um esboço (desenho, screenshot, etc) e uma descrição textual da IU que será descrita e/ou implementada pelos demais artefatos do CIP. O screenshot da Figura 1 ilustra uma SUI do padrão de interação “Parts Selector”.

Cada SUI pode conter um conjunto de CUIs que a descrevam em diferentes níveis de detalhamento e para diferentes modalidades de interface (conjunto de widgets) e tecnologias de implementação (plataformas/linguagens). Uma CUI pode encapsular (usar) CUIs de outros CIPs. Com isso, além da prática de reuso que o encapsulamento proporciona, consegue-se descrever como vários padrões de interação funcionam de forma integrada num mesmo contexto de IU. Esta prática reforçaria o desenvolvimento de uma *linguagem de padrões* na coleção de padrões de interação sendo utilizada.

Finalmente, uma SUI pode estar associada a várias FUIs que a implementam em tecnologias, linguagens ou frameworks distintos.

Todos os artefatos que constituem o CIP são obtidos através de processos de reificação (abordagem top-down), abstração (abordagem bottom-up) ou tradução descritos no CRF. A nossa abordagem não define se e em qual ordem devem ser construídos. Entretanto, os que são construídos num mesmo “caminho” de reificação/abstração devem ser compatíveis entre si.

Quanto mais alta a abstração de um CIP, maior a sua possibilidade de reuso. Para aumentar as possibilidades de reuso do CIP e facilitar os processos de abstração, reificação e tradução, sugere-se descrever uma SUI em 3 CUIs com diferentes níveis de detalhamento (CUI básica, intermediária e completa). A CUI básica contém apenas a discriminação dos elementos de interação, o layout e descrições textuais de

comportamentos relevantes da IU. A CUI completa contém uma especificação detalhada o suficiente para se gerar uma FUI a partir dela. Obviamente, a CUI intermediária, contém descrições no meio termo entre as duas.

Para ilustrar o processo de construção de um CIP, considere o diagrama da Figura 5 que apresenta a composição atual dos artefatos do CIP “Parts Selector”. Algumas seqüências possíveis de construção destes artefatos seriam: 1-2-3-4-5-6, 2-4-5-6-3-1 e 1-6-2-5-4-3.

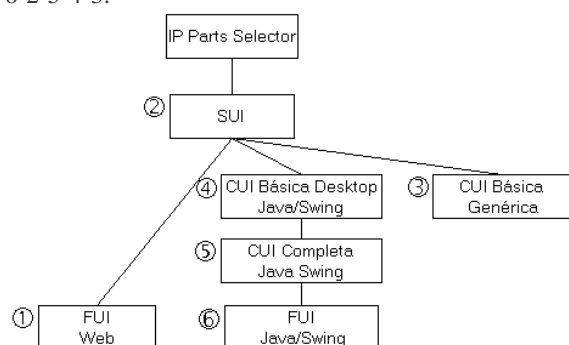


Figura 5. Exemplo de estrutura do CIP “Parts Selector”

5. Projetando interfaces a partir dos casos de uso

Casos de uso, hoje, são consensualmente aceitos como muito importantes para a modelagem de software, sobretudo orientado a objetos, e que está no núcleo de abordagens como UP (e RUP) e em notações como UML. Por definição, um caso de uso contém uma narrativa que descreve a interação que se dá pela troca (envio e/ou recebimento) de informações entre um ator e um sistema [6].

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. Atualmente, é senso comum de que estas descrições devem estar no grau de abstração essencial [7, 6]. Na narrativa essencial, são explicitadas as interações entre o ator e o sistema sem descrever detalhes de como e onde estas interações serão realizadas.

As seqüências de interações (ou ações) que compõem a narrativa do caso de uso não são identificadas como um elemento no metamodelo de casos de uso da UML. Entretanto, para aplicar a nossa abordagem de reuso, é necessário explicitar estas interações através da identificação dos fluxos e passos que compõem a narrativa do caso de uso. Desta forma, propomos que o metamodelo de casos de uso seja estendido conforme apresentado na Figura 6.

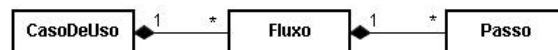


Figura 6. Metamodelo de Caso de Uso

A partir da modelagem dos casos de uso essenciais, constrói-se um protótipo da interface com o usuário. O protótipo é uma representação limitada (um croqui ou esboço) do projeto da interface que permite aos usuários ter uma visão concreta e visual da interface do sistema e explorar a sua conveniência [17].

Os protótipos constituem um conjunto de artefatos que chamamos de casos de uso apresentados. Os casos de uso apresentados são a forma visual e perceptível da descrição dos fluxos dos casos de uso essenciais. Eles ainda não fazem referência aos objetos do sistema, pois representam um nível intermediário entre a narrativa conceitual definida no modelo de casos de uso e os objetos e suas colaborações definidos na fase de realização dos casos de uso.

Somente após a conclusão da fase de apresentação dos casos de uso é que se pode começar com a atividade de projeto OO propriamente dito, onde os elementos da IU, juntamente com os conceitos, são traduzidos em termos de objetos e classes e define-se como esses objetos e classes colaboram entre si para a realização do caso de uso.

6. Padrões de casos de uso

Considerando que os casos de uso servem como base para o projeto de IU, parece uma abordagem natural considerar o reuso de casos de uso como uma estratégia para reuso de interfaces. Para isso, é preciso definir o que torna um caso de uso reusável. Dado que a probabilidade de algo ser reusado está diretamente relacionada com a forma de sua descrição, alguns autores [2] sugerem que casos de uso essenciais são mais reusáveis do que os casos de uso concretos ou do que cenários.

Na descrição dos casos de uso essenciais, é possível identificar padrões de casos de uso (use case patterns) [1, 19, 15] que podem ser documentados e catalogados de forma semelhante a padrões de projeto (design patterns) [12].

Logo, um padrão de caso de uso é aquele que pode ser reusado em função de representar uma tarefa (orientado a tarefa) ou um domínio (orientado a domínio) recorrente a vários sistemas.

Estes padrões de casos de uso representam soluções para problemas específicos e podem ser reusados em contextos similares. Procura-se, sempre que possível, definir padrões de casos de uso parametrizados, ou seja, a parametrização do texto da narrativa

do caso de uso de forma a aumentar as chances de reuso. Parâmetros tornam localizadas as referências específicas ao sistema ou ao domínio da aplicação para o qual ele foi criado e possibilita que o mesmo caso de uso possa ser usado em situações diferentes.

De uma forma geral, os padrões de casos de uso orientados a tarefas são mais facilmente parametrizáveis e reusáveis do que padrões de casos de uso orientados a domínios.

Os padrões de casos de uso são muito mais úteis se puderem ser relacionados uns aos outros e/ou agrupados por tarefas ou domínios em comum. Constantine e Lockwood propõem que, além dos relacionamentos de inclusão, extensão e generalização, os casos de uso sejam relacionados por afinidade (semelhança e equivalência) [7]. Além destes relacionamentos, os casos de uso que participam na solução de um determinado problema podem compor uma linguagem de padrões de casos de uso a qual chamamos de *padrão de domínio*.

Um padrão de domínio descreve o conjunto de casos de uso que são re-correntes na modelagem de casos de uso de sistemas de um mesmo domínio de problema e que se relacionam entre si e colaboram para dar suporte à prestação de uma funcionalidade de valor para os usuários. Um padrão de domínio de comércio eletrônico, por exemplo, pode conter os casos de uso: “Consulta um produto”, “Busca por palavra-chave”, “Busca por categoria”, “Coloca no carrinho de compras”, “Efetua compra”, entre outros, pois estes casos de uso são comuns a todos os sistemas de comércio eletrônico. Da mesma forma, o conjunto de casos de uso que suportam o serviço de autenticação e de permissões de usuários, e o conjunto de casos de uso que suportam o serviço de publicação de notícias em um site na Web, são exemplos de padrões de domínio.

7. Ligando padrões de casos de uso a linguagens de descrição de interfaces com o usuário

Da mesma forma que um padrão concreto de interação, um padrão de caso de uso pode conter vários níveis de reificação de IU, formando uma hierarquia de SUIs, CUIs e FUIs que modelam e implementam a IU do padrão de caso de uso. Ao conjunto de artefatos que compõem a hierarquia juntamente com o padrão de caso de uso, chamamos de *padrão de caso de uso reificado* (*reified use case pattern - RUCP*).

De forma semelhante à estrutura de um CIP, a estrutura de um RUCP é apresentada na Figura 7. No RUCP, uma SUI é o protótipo que constitui a apre-

sentação do padrão de caso de uso. Todos os demais elementos constituem os vários níveis de reificação deste protótipo.

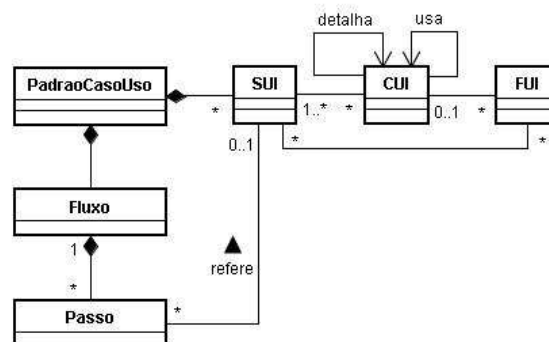


Figura 7. Estrutura de um padrão de caso de uso reificado.

Como o padrão de caso de uso deve ser escrito na forma essencial (independente de tecnologia), podem-se associar a este padrão hierarquias de IUs nos mais diversos tipos de ambiente de interação e de tecnologias de implementação. Um único padrão de caso de uso pode, por exemplo, conter SUIs e CUIs que o apresentem em um ambiente de interação Web, em um ambiente gráfico desktop, em um ambiente de um telefone celular, e assim por diante. Para cada um destes ambientes, podem existir FUIs que os implementem nas mais variadas linguagens de programação e/ou arquiteturas.

Cada passo do padrão de caso de uso pode ser associado com a respectiva IU que suporta a execução do passo. Isso permite “fracionar” a IU do caso de uso, restringindo-a ao conjunto de espaços de interação e de elementos de interação necessários à realização do passo.

É neste nível de granularidade da IU que se apresentam a maioria das soluções propostas pelos padrões de interação, permitindo - como veremos a seguir - o uso integrado e sistemático destes padrões no projeto e reuso de IHC.

8. Casos de reuso

Um caso de reuso existe quando são encontrados e aplicados artefatos para serem reusados a partir da especificação de um caso de uso. Na nossa abordagem, o principal elemento de ligação entre o caso de uso a ser desenvolvido e os possíveis artefatos a serem reusados é o padrão de caso de uso.

A abordagem utiliza a hierarquia de reificação do padrão de caso de uso como fio condutor do reuso. A identificação dos possíveis artefatos a serem reusados segue uma orientação top-down, ou seja, são analisa-

das as possibilidades de reuso a partir de especificações mais conceituais dos casos de uso (níveis mais altos da hierarquia de reificação) até as mais concretas (níveis mais baixos da hierarquia de reificação). A quantidade de artefatos reusados dependerá do acervo de artefatos disponível e da decisão tomada sobre qual o caminho a seguir no nível mais baixo.

As atividades para a identificação de artefatos reusáveis ocorrem de forma integrada com o processo de desenvolvimento OO. Estas atividades são chamadas de *análise de reuso de casos de uso* e *análise de reuso de interações*. A análise de reuso de casos de uso acontece na fase de análise de requisitos do SsD em paralelo às atividades de modelagem de casos de uso e de modelagem conceitual. Posteriormente, na fase de apresentação do caso de uso, ocorre a análise de reuso de interações em paralelo às atividades de projeto e de prototipação da IU. Estas atividades serão detalhadas a seguir.

8.1 Análise de reuso de casos de uso

Na análise de reuso de casos de uso procura-se identificar os padrões de domínio e padrões de casos de uso que sejam aderentes aos requisitos funcionais do SsD (Sistema sendo Desenvolvido). A pesquisa é feita com o enfoque tanto na busca de casos de uso com recorrência de domínio do problema quanto nos casos de uso com recorrência de tarefas padrão. O resultado desta atividade é um conjunto de casos de uso candidatos a reuso no SsD.

A seguir, para cada caso de uso candidato, procura-se identificar, na hierarquia de reificação desse caso de uso, até que nível de artefatos será possível reusar. Obviamente, poderão ser encontrados mais artefatos para reuso se, no momento da realização desta atividade, o ambiente de interação e a tecnologia de implementação do SsD já tiverem sido definidos. Caso contrário, o reuso poderá se restringir em nível de narrativa do caso de uso ou, no máximo, em nível de SUI.

8.2 Análise de reuso de interações

Durante a fase de apresentação dos casos de uso ocorre a segunda análise de reuso. Nesta fase, o foco é explorar as possibilidades de reuso de interações para todos os casos de uso do SsD em que não foi possível identificar, no mínimo, reuso em nível de SUI na análise de reuso anterior.

A análise de reuso de interações ocorre simultaneamente com as atividades de projeto e de prototipação da IU. À medida que o projetista de interface faz

a leitura da narrativa do caso de uso e tenta esboçar uma IU que o suporte, também deve fazer uma pesquisa no repositório em busca de padrões concretos de interação (CIPs) que sejam adequados à execução das tarefas descritas nos passos do caso de uso. A busca no repositório deve ser feita de forma recursiva e incremental ao longo do projeto e prototipação da IU e deve levar em conta a natureza da tarefa, os requisitos de usabilidade e as informações necessárias para um passo ser executado.

Para cada CIP identificado como adequado para a IU que está sendo projetada verifica-se, na hierarquia de reificação deste CIP, até que nível e quais os artefatos que poderão ser reusados. O critério de escolha destes artefatos deve se basear na existência de um ramo na hierarquia do CIP que seja aderente ao ambiente de interação, tecnologia de implementação e compatibilidade com a arquitetura do SsD.

9. Conclusão

Neste artigo, foi apresentada uma abordagem de reuso que pode contribuir para o aumento do reuso de IUs durante o processo de desenvolvimento de um software. Nossa abordagem prevê o reuso de IUs a partir da especificação das interfaces de padrões de caso de uso, padrões de interação e dos artefatos de software que o implementam.

Desta forma, pretende-se atingir os seguintes objetivos: 1) propiciar o reuso de interfaces de usuário em vários níveis e 2) explicitar o comportamento das IUs como forma de garantir implementações consistentes e com alto grau de usabilidade e a construção das interfaces com o usuário ainda carece de abordagens efetivas e sistemáticas de reuso se comparadas com as práticas já existentes de reuso dos demais artefatos de software. Além disso, deve-se fazer um esforço para que a prática de reuso esteja adaptada e integrada com as demais atividades de projeto do software de forma que seja praticado naturalmente como parte integrante do processo de desenvolvimento. Como perspectivas de continuidade deste trabalho vislumbram-se: a) definição da estrutura e funcionamento de um repositório de CIPs e RUCPs, e b) a integração deste repositório com ferramentas CASE de modelagem da UML e ambientes integrados de desenvolvimento (IDEs).

10. Referências

- [1] Biddle, R.; Noble, J.; Tempero, E. Essential Use Cases and Responsibility in Object-Oriented Development. 2001. Disponível em: <

- <http://www.foruse.com/articles/euc-responsibility.pdf> >. Acesso em: abril 2006.
- [2] Biddle, R.; Noble, J.; Tempero, E. Supporting Reusable Use Cases In Proceedings of the 7th International Conference on Software Reuse: Methods, Techniques, and Tools. pp. 210-226. 2002.
 - [3] Borchers, J. A Pattern Approach to Interaction Design em Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques. Pp. 369-378. 2000.
 - [4] Borchers, J. Designing Interactive Music Systems: A Pattern Approach, in 8th International Conference on HCI. 1999. pp.276-280.
 - [5] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouil-Lon, L., Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers, vol. 15, no. 3, 289-308. 2003.
 - [6] Cockburn, A. Writing Effective Use Cases [S.l.]: Addison-Wesley, 2001.
 - [7] Constantine, L.; Lockwood, L. Software for Use: A Practical Guide to The Models and Methods of Usage-centered Design. [S.l.]: Addison-Wesley, 1999.
 - [8] Cox, B. "What if there is a silver bullet... and the competition gets it first?" em Dr. Dobb's Journal, Oct 1992.
 - [9] Duyne, D.; Landay J.; Hong, J. The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience. Boston: Addison-Wesley, 2002.
 - [10] Gaffar, A. et al. Modeling Patterns for Task Models, em TAMODIA 2004, pp 99-104. 2004.
 - [11] Gaffar, A.; Seffah, A.; Van de Poll, J. HCI Pattern Semantics in XML: a Pragmatic Approach. 2005.
 - [12] Gamma, E. et al. Padrões de Projeto – Soluções reutilizáveis de Software Orientado a Objetos. Bookman, 2000.
 - [13] Mahemoff, M.; Johnston, L. Pattern Languages for Usability: An Investigation of Alternative Approaches. In Tanaka, J. (Ed.), in APCHI 98 Proceedings , 25-31. 1998.
 - [14] Mahemoff, M.; Johnston, L. Usability Pattern Languages: the "Language" Aspect. em HCI: Interact '01. pgs 350-358. Disponível em <http://mahemoff.com/paper/> Acesso em 04/2006.
 - [15] Overgaard, G.; Palmkvist, K. Use Cases Patterns and Blueprints. Indianapolis: Addison-Wesley, 2005.
 - [16] Pattern Language Markup Language (PLML) Disponível em www.hcipatterns.org. nov/2006.
 - [17] Preece, J.; Rogers, Y.; Sharp, H. Design de interação: além da interação homem-computador. Porto Alegre: Bookman, 2005
 - [18] Rijken, D. The Timeless Way... the design of meaning. SIGCHI Bulletin.Vol. 6, No. 3. PP. 70-79. 1994
 - [19] Saeki, M. Reusing Use Case Descriptions for Requirements Specification: Towards Use Case Patterns in Proceedings of the Sixth Asia Pacific Software Engineering Conference. 1999.
 - [20] Sindre G.1; Conradi R.; Karlsson E.-A The REBOOT Approach to Software Reuse Journal of Systems an Software, Volume 30, Number 3, September 1995, pp. 201-212(12)
 - [21] Sutcliffe, A. & Carroll, J. Designing Claims for Reuse in Interactive Systems Design, in International Journal of Human-Computer Studies, /50(3), pp 213-242. 1999.
 - [22] The Parts Selector Interaction Pattern. Disponível em www.welie.com/patterns/showPattern.php?patternID=parts-selector . Acessado 07/2006.
 - [23] Tidwell, J. Interaction Patterns In: Proceedings of Pattern Languages of Program Design. 1998.
 - [24] Todd, E.; Kemp, E.; Phillips, C. What makes a good User Interface pattern language? In: Proceedings of the 5th AUIC2004. 2004.
 - [25] Tracz, W. Software Reuse Myths, em ACM SIGSOFT Software Engineering Notes vol. 13 no. 1, Janeiro, págs. 17-21.
 - [26] UIML User Interface Markup Language Specification Working Draft 3.1. March 2004. Disponível em <http://www.oasis-open.org/committees/download.php/5937/uiml-core-3.1-draft-01-20040311.pdf>. Acesso 07/2006
 - [27] Vanderdonckt, J.; Bodart, F. Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection. In Proc. of ACM Conf. on Human Aspects in Computing Systems InterCHI'93 (Amsterdam, April 24-28, 1993). ACM Press, New York, 424-429. 1993
 - [28] Welie, M.; Trættemberg, H. Interaction Patterns In User Interfaces in Proceedings of the Pattern Languages of Programming PLoP' 2000. Disponível em: <http://www.idi.ntnu.no/~hal/publications/design-patterns/PLoP2k-Welie.pdf> . Acesso 04/2006.
 - [29] Welie, M.; Veer, G.; Eliëns, A. Patterns as Tools for User Interface Design. In International Workshop on Tools for Working with Guidelines. pp. 313-324. 2000.