

# Reuso de Interfaces Através de Padrões Concretos de Interação

Augusto Abelin Moreira<sup>1,2</sup>

<sup>1</sup> PROCERGS (Companhia de Processamento de Dados do Estado do Rio Grande do Sul)  
Praça dos Açorianos, S/N  
Caixa Postal 236 – 90010-340  
Porto Alegre – RS – Brasil  
+55 51 3210-3100  
augusto@procergs.rs.gov.br

Marcelo Soares Pimenta<sup>2</sup>

<sup>2</sup> 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

The goal of this paper is to introduce an approach to user interface (UI) reuse 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.

**Categories and subject descriptors:** H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces---Standardization, Theory and methods; D.2.13 [Software Engineering]: Reusable Software--- Reusable libraries, Reuse models; D.2.2 [Software Engineering]: Design Tools and Techniques---User interfaces

**Keywords:** user interface reuse, interaction pattern, concrete interaction pattern, User Interface Description Language, 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 (4).

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 (17).

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) (13).

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 (8). Em particular, um padrão de interação deve estar focado em soluções que melhorem a usabilidade do sistema em uso (21).

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 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. Finalmente, a seção 5 é a conclusão.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission from authors.

IHC 2006 – VII Simpósio Sobre Fatores Humanos em Sistemas Computacionais. Novembro, 2006, Natal, Rio Grande do Norte, Brazil.  
ACM .2006 ISBN 1-59593-432-4/06/0011.

## 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 (12). Entretanto, somente a partir dos trabalhos de Borchers (1, 2), Sutcliffe e Carroll (14) e Tidwell (15) 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 (16). 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: 5, 9, 10, 15 e 20) muitas vezes incompatíveis entre si (6).

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 (7) e (11)), 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, podem 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.

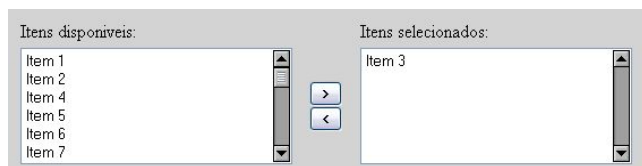


Figura 1. Uma IU do padrão de interação “Parts Selector”

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 conceituais 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 (3). 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 translação). A Figura 2 ilustra os dois passos de desenvolvimento do CRF com os processos que serão usados na nossa abordagem.

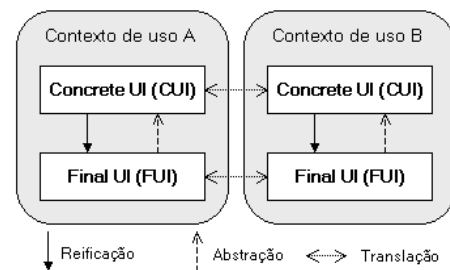


Figura 2. Os dois níveis mais “concretos” do Cameleon Framework.

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) (19).

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, pra citar algumas. Nós optamos por descrever as CUIs em UIML (18) 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.

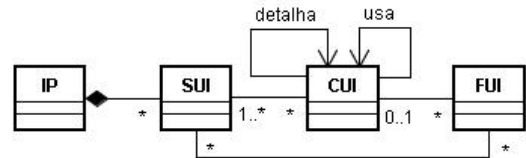


Figura 4. Estrutura de um padrão concreto de interação.

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 translação 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 translação, sugere-se descrever uma SUI em 3 CUIs com diferentes níveis de detalhamento (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.

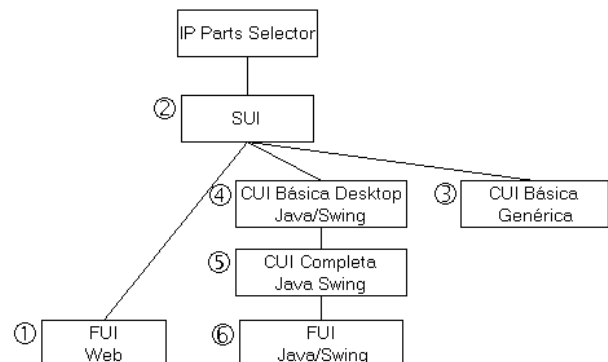


Figura 5. Exemplo de estrutura do CIP “Parts Selector”

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.

## 5. 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 um padrão 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.

A definição 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) a integração do modelo de interfaces descrito no CIP com modelos de domínio e modelos de tarefas, e b) definição da estrutura e funcionamento de um repositório de CIPs.

## 6. REFERÊNCIAS

- [1] Borchers, J. Designing Interactive Music Systems: A Pattern Approach, in 8th International Conference on Human-Computer Interaction. 1999. pp.276–280.
- [2] 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.
- [3] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers, vol. 15, no. 3, 289–308. 2003.
- [4] Cox, B. “What if there is a silver bullet... and the competition gets it first?” em Dr. Dobbs’s Journal, Oct 1992. Disponível em <http://virtualschool.edu/cox/pub/92ByteWhatIfSilverBullet/index.html> Acessado em abril 2006.
- [5] 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.
- [6] Gaffar, A. et al. Modeling Patterns for Task Models, em TAMODIA 2004, pp 99-104. 2004.
- [7] Gaffar, A.; Seffah, A.; Van de Poll, J. HCI Pattern Semantics in XML: a Pragmatic Approach. 2005.
- [8] Gamma, E. et al. Padrões de Projeto – Soluções reutilizáveis de Software Orientado a Objetos. Bookman, 2000.
- [9] Mahemoff, M.; Johnston, L. Pattern Languages for Usability: An Investigation of Alternative Approaches. In Tanaka, J. (Ed.), Asia-Pacific Conference on Human Computer Interaction (APCHI) 98 Proceed-ings , 25-31. Los Alamitos, CA: IEEE Computer Society. 1998.
- [10] Mahemoff, M.; Johnston, L. Usability Pattern Languages: the “Language” Aspect. em Human-Computer Interaction: Interact ’01. pgs 350-358. Disponível em <http://mahemoff.com/paper/> Acesso em: abril 2006.
- [11] Pattern Language Markup Language (PLML) Disponível em <http://www.hcipatterns.org>. Acessado em julho 2006.
- [12] Rijken, D. The Timeless Way... the design of meaning. SIGCHI Bulletin.Vol. 6, No. 3. PP. 70–79. 1994
- [13] Sindre G.1; Conradi R.; Karlsson E.-A The REBOOT Approach to Software Reuse Journal of Systems and Software, Volume 30, Number 3, September 1995, pp. 201-212(12)
- [14] 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.
- [15] Tidwell, J. Interaction Patterns In: Proceedings of Pattern Languages of Program Design. 1998.
- [16] Todd, E.; Kemp, E.; Phillips, C. What makes a good User Interface pattern language? In: Proceedings of the 5th Australasian User Interface Conference (AUIC2004). 2004.
- [17] Tracz, W. Software Reuse Myths, em ACM SIGSOFT Software Engineering Notes vol. 13 no. 1, Janeiro, págs. 17-21.
- [18] 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>. Acessado em julho 2006.
- [19] 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
- [20] Welie, M.; Trætterberg, 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 em: abril 2006.
- [21] 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.
- [22] The Parts Selector Interaction Pattern. Disponível em <http://www.welie.com/patterns/showPattern.php?patternID=parts-selector> . Acessado em julho 2006.