

Empregando modelo de interação na representação de padrões de usabilidade

Macilon Araújo Costa Neto¹, Jair Cavalcanti Leite²

¹Universidade Federal do Acre
Caixa Postal 500, 69915-900, Rio Branco, AC, Brasil

²Universidade Federal do Rio Grande do Norte
Caixa Postal 1524, 59072-970, Natal, RN, Brasil

macilon@ufac.br, jair@dimap.ufrn.br

Resumo. *Padrões de projeto (design patterns) estão cada vez mais difundidos em desenvolvimento de software desde a proposta de um catálogo de soluções reutilizáveis em software orientado a objetos, conhecidos como padrões GoF [Gamma et al. 1995]. Padrões têm sido aplicados no processo de software e no design de interfaces de usuário com bastante sucesso. Uma das razões da popularidade dos padrões é a possibilidade de representar de forma sistemática o conhecimento tácito obtido da experiência dos envolvidos na construção do software. No entanto, ao contrário dos padrões GoF, os padrões para sistemas com Interfaces de Usuário Gráfica (GUI) e Web (WUI) são representados de uma forma que não possibilita uma visualização deles com diagramas que modelam e especificam a interação usuário-sistema de forma que se possa realizar análises de usabilidade e do impacto na arquitetura de software. Neste artigo, mostramos como padrões de design de interação podem ser representados na linguagem de modelagem de interação ALaDIM e discutimos os benefícios que esta integração pode proporcionar aos profissionais de engenharia de software e IHC envolvidos.*

Abstract. *Design patterns are widely used in software development since the original proposal of a catalog of reusable object-oriented solutions - the GoF patterns. They have been applied successfully also in software process and user interface and interaction design. One of the main reasons of its popularity is because it is possible to represent the tacit design knowledge that has been revealed in design practice. However, differently as in GoF patterns, GUI and WUI patterns are not represented using visual diagrams, which has been the lingua franca for most software development stakeholder. Beyond that, visual languages allow usability and architectural impact analysis at design time. In this paper, we show how design patterns can be represented in ALaDIM, an interaction design language. We also argue that the integration of interaction patterns and visual languages enhances the collaboration among HCI and software engineering professionals.*

1. Introdução

Tipicamente, os Sistemas de Informação baseados em computadores, ou simplesmente SI, têm na sua Interface de Usuário (IU, ou simplesmente interface, neste trabalho), um componente fundamental que permite que as pessoas, para quem os sistemas são construídos,

possam utilizá-los de forma eficaz, eficiente e com satisfação. Interfaces mal projetadas causam diversos problemas de usabilidade que podem causar desde insatisfação até erros graves que podem afetar diversos envolvidos.

Nas últimas décadas, a evolução tecnológica nas interfaces dos SI chega a ser espantosa. Contudo, o processo de design destas interfaces não tem evoluído como esperado. Embora em alguns aplicativos de escritório, como planilhas e editores de texto, as interfaces de usuário tenham usabilidade bastante satisfatória, muitos Sistemas de Informação apresentam problemas. Parte destes problemas ocorrem devido à falta de um bom processo de design de sua interface. Métodos de desenvolvimento como o Processo Unificado ou o Scrum não colocam o design como uma atividade de primeiro plano. Além disso, faltam aos desenvolvedores a formação adequada em um processo de design e o uso de técnicas e ferramentas apropriadas para essa tarefa.

Os padrões de design têm sido amplamente empregados no design de software orientado a objetos [Gamma et al. 1995] e no design de interfaces de usuário [Tidwell 2002]. Apesar de ter surgido há várias décadas na arquitetura urbana [Alexander et al. 1977], somente quase vinte anos depois, Gamma *et al* [Gamma et al. 1995] publicaram um trabalho sobre uso de padrões na engenharia de software. Desde então, eles têm sido cada vez mais enfatizados em outras áreas da ciência da computação, como, por exemplo, no design de interação e da interface com usuário, visando a melhoria da usabilidade dos sistemas interativos [Borchers 2001, van Welie 2001, Tidwell 2002].

Um problema importante nos padrões de interface e interação é a forma como eles têm sido documentados. Para Alexander [Alexander 1979], “cada padrão é uma regra de três partes, que expressa uma relação entre um determinado contexto, um problema e uma solução”. Assim, a descrição de um padrão é organizada em várias seções, cuja granularidade pode variar bastante. Por exemplo, em Gamma *et al* [Gamma et al. 1995] é usada uma dúzia, já em Tidwell [Tidwell 2002] são usadas apenas cinco. Essa diferença na quantidade de seções não é um problema, contanto que elas cubram a descrição das três partes.

Em seu trabalho, Alexander *et al* [Alexander et al. 1977] advogam que na descrição da solução deve haver “um diagrama, que mostra a solução numa forma diagramática, com rótulos para indicar seus principais componentes”. A presença desta representação tem sido uma evidente diferença entre a descrição dos padrões de ES (Engenharia de Software), tais como [Gamma et al. 1995] e [Buschmann et al. 2001]; e de IHC (Interação Humano-Computador), tais como [Tidwell 2005] e [van Welie 2001]. Nos primeiros, são usados modelos para representar os principais componentes do padrão e suas relações, onde é apresentado um diagrama de classes (em alguns casos também há um diagrama de sequência/colaboração) na seção estrutura [Gamma et al. 1995]. Já no segundo grupo, os principais componentes do padrão e suas relações não são modelados e a principal forma de representação é feita por uso de *screenshots* de instâncias reais do padrão, apresentados na imagem que inicia a descrição e/ou na seção de exemplos [Tidwell 2002], como aparece na figura 1.

Dessa forma, neste trabalho, argumentamos que os padrões de interface e interação tenham seus aspectos interativos e funcionais descritos sob a forma de modelos de interação, e que estes modelos sejam criados usando uma linguagem diagramática, como

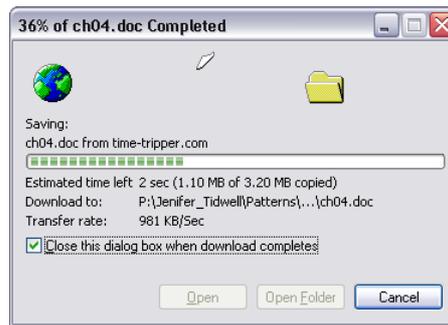


Figura 1. Screenshot do padrão Progress Indicator (extraído de [Tidwell 2002]).

ocorre com a UML (*Unified Modeling Language*) na descrição dos padrões de ES. No entanto, embora a UML tenha alcançado notório sucesso na descrição dos aspectos funcionais dos sistemas, ela não se mostra eficiente para descrever os aspectos de interação entre a interface e seu usuário.

Assim, nossa proposta é que uma nova seção chamada *estrutura* seja acrescentada à descrição dos padrões de IHC, como é advogado por Alexander [Alexander et al. 1977], para representar os elementos principais do padrão e suas relações. Nesta seção, deverá constar um modelo de interação para a porção da interface correspondente ao padrão, que deverá contemplar os principais elementos interativos e quais as ações o usuário deverá realizar sobre os mesmos para alcançar seus objetivos.

Nesse trabalho, mostraremos como representar padrões de interação da interface de usuário usando uma linguagem diagramática. Nosso objetivo é que este modelo seja usado como um artefato complementar aos modelos da ES e aos protótipos de interface. Para isso, utilizamos ALaDIM (*Abstract Language for Description of Interactive Message*), uma linguagem para modelar a interação do usuário com a interface de um SI e que permite a representação abstrata dos elementos interativos presentes nos padrões de IHC.

2. Trabalhos relacionados

O trabalho de Brown e Marshall [Brown and Marshall 1998], demonstra como é importante o compartilhamento de artefatos utilizados por profissionais de ES e IHC. Eles argumentam que a comunicação entre os profissionais destas áreas é dificultada porque os artefatos representam diferentes pontos de vistas e diferentes níveis de abstração. Para superar essas dificuldades, os autores apresentam uma ferramenta que possibilita aos designers navegar simultaneamente entre várias artefatos como, hierarquia de tarefas, especificações de diálogo, especificações da arquitetura da interface e fragmentos de programas. A produção desses artefatos é guiada pela composição de dois processos de software: um de IHC orientado a tarefas; e um incremental de ES.

Apesar do trabalho de Brown e Marshal enfatizar a integração entre as duas áreas, para os artefatos empregados isso ocorre na ligação entre a descrição do cenário e do cenário estruturado, que na verdade é um diagrama de sequência. Dessa forma, dos quatro artefatos empregados, dois (tarefas e cenários) representam uma perspectiva centrada no usuário e os outros (cenários estruturados e arquitetura da interface) representam uma perspectiva centrada no sistema.

Na proposta, os autores usam o que eles chamam de cenários estruturados (diagrama de sequência) como um modelo de diálogo da interface. Contudo, ele é um artefato que representa a perspectiva centrada no sistema e não no usuário (como é confirmado no exemplo apresentado no trabalho), onde o diálogo enfatizado é aquele entre os componentes da interface e não entre o usuário e o sistema. Assim, não há uma descrição mais estruturada sobre o diálogo na perspectiva dos usuários que é de interesse dos profissionais de IHC, mantendo de fato os cenários como o artefato integrador. Nossa proposta diverge por colocarmos nosso modelo de diálogo, relativamente mais estruturado que os cenários, porém acessível por todos os envolvidos, como o artefato integrador.

O trabalho de da Silva *et al* [da Silva et al. 2005] apresenta um modelo de processo de prototipação apoiado por padrões, onde os autores apontam o protótipo como artefato que possibilita uma integração entre os profissionais de ES e IHC. Contudo, assim como nos cenários, as duas áreas estão distintamente focadas neste artefato e no processo de desenvolvimento do mesmo. Os autores também destacam que, além da prototipação, o uso de padrões é outra prática adotada por essas duas áreas.

Sobre os aspectos comunicativos, da Silva *et al* [da Silva et al. 2005] defendem que a comunicação entre os profissionais seja ampla o suficiente para que o protótipo em desenvolvimento permita explorar os aspectos focados pelas duas áreas, possibilitando o aprendizado interdisciplinar. Não significando que um especialista de uma área se torne especialista na outra, mas que tenha compreensão dos aspectos relevantes desta. Este trabalho assemelha-se ao de Brown e Marshall [Brown and Marshall 1998], apesar de destacar um artefato como integrador, eles acabam colocando os profissionais das duas áreas dentro de algum processo específico.

No trabalho de Bueno e Barbosa [Bueno and Barbosa 2007], as autoras apresentam uma forma de usar diagramas MoLIC [Barbosa and de Paula 2003], para representar sequências de interações associadas a tarefas, que formam padrões já conhecidos (ex.: *Login, Main navigation, etc.*). Além disso, são apresentados exemplos que demonstram a capacidade da linguagem MoLIC de “colar” os vários fragmentos (diagramas de vários padrões) formando uma solução de design de um sistema completo. Elas defendem o uso da MoLIC como uma linguagem de representação para documentar padrões de soluções de design de interação, argumentando que uma das principais dificuldades ao se analisar a literatura sobre padrões de IHC é a ausência de uma linguagem padronizada para representação da interação.

Nosso trabalho concorda com de Bueno e Barbosa no que se refere à ausência de padronização de uma forma de representar a estrutura abstrata dos padrões de IHC e defendemos o uso de modelos de interação. Além disso, propomos a construção de uma biblioteca de modelos de interação para um conjunto de padrões, que será disponibilizado junto à uma ferramenta [Lavor and Leite 2011] de modelagem que irá permitir que os diagramas (fragmentos) representando os padrões já modelados e incorporados à biblioteca sejam usados (no estilo *drag-and-drop*) dentro das soluções desenvolvidas.

3. Uma linguagem para modelagem de IHC

Modelar a interação humano-computador é uma das principais atividades realizadas pelos especialistas de IHC, durante o design de interação. Dela resulta um dos principais artefatos, o modelo de interação. Este modelo é importante para os especialistas de IHC

por vários fatores. O principal deles é a possibilidade de uma avaliação formativa da usabilidade da interface em desenvolvimento, através de análise e inspeção. Dessa forma, ainda durante o design é possível detectar falhas no projeto do processo interativo. Outra vantagem do uso de um modelo de interação é poder ter uma visão abstrata do processo de interação e navegação entre diferentes elementos da interface de usuário sem a necessidade de tratar detalhes de *layout*, cores e outros elementos estéticos. Assim, o foco do designer e do avaliador fica apenas no processo de interação.

Neste trabalho, tal como os cenários e os protótipos, o modelo de interação também é um artefato de comunicação entre os profissionais de IHC e de ES. Nele é possível identificar as funcionalidades oferecidas pelo sistema, bem como as informações (ou dados) manipuladas por elas, sendo que estes elementos (conceitos) constituem a base do modelo de funcionalidades, tido como um dos principais artefatos produzidos pelos engenheiros de software [Trätteberg 2008]. Contudo, o modelo de interação apresenta-se mais estruturado e sistemático que os cenários e menos detalhado que os protótipos. Isto demonstra que estes artefatos não são excludentes e sim complementares.

O modelo de interação, aqui proposto, possibilita ao designer uma visualização que lhe permite planejar e avaliar a dinâmica (comportamento) do sistema face a interação do usuário. Ele expressa a relação comportamental entre o usuário e as funcionalidades oferecidas pelo sistema através de sua interface. Além disso, ele também fornece ao usuário o conhecimento sobre quais funcionalidades a aplicação lhe oferece e instruções sobre os passos necessários para usufruir dessas funcionalidades, ou seja, como interagir com sua interface. Dessa maneira, é notório que estes artefatos possuem uma estreita relação, que pode ser usada como ponte entre as atividades de IHC e ES.

Apresentamos os conceitos da linguagem ALaDIM usados para representar um modelo que descreve o processo de interação entre o usuário e o sistema, através de uma abordagem baseada nas ações do usuário e reações do sistema, onde o usuário interage através de espaços de interação. Os espaços de interações, por meio das ações do usuário, permitem acionar as funções dentro do sistema e comunicar ao usuário seus resultados, possibilitando, quando for o caso, novas interações dentro do mesmo espaço de interação ou através de outros espaços. Nas subseções a seguir, são apresentados os elementos da linguagem, pois a mesma encontra-se em desenvolvimento e avaliação. Eles são exemplificados no modelo de interação da figura 2 para o qual o sistema é descrito a seguir.

O banco Moneyless precisa criar uma rede de quiosques para seus clientes. Porém, com sérios problemas econômicos, conseguiu comprar um lote de equipamentos sem mecanismos para saque ou depósito, nem mesmo para impressão de documentos. Assim, seus quiosques permitirão aos clientes apenas realizar transações eletrônicas e com saídas em tela. Para isso, o sistema deverá permitir, ao cliente autenticado, realizar consultas de saldo e extrato de suas contas, bem como efetuar transferência para outras contas. Por motivos de segurança, o sistema deverá controlar o tempo de inatividade do cliente.

3.1. Espaço de interação

Espaços de interação representam os mecanismos para o usuário interagir e controlar as funções do sistema. Assim, é através deles que o usuário irá fornecer dados de entrada, ativar a execução do processo e perceber seus resultados. Um espaço de interação está

da funcionalidade associada a um espaço de interação. Na figura 2, temos exemplos dos operadores *sequence* e *join*. Os operadores são usados nas seguintes situações:

- *sequence*: quando o usuário deve executar as interações de maneira ordenada;
- *repeat*: quando o usuário precisa repetir várias vezes as interações;
- *choose*: quando o usuário precisa escolher uma dentre várias interações;
- *combine*: quando duas ou mais interações têm alguma dependência entre si;
- *join*: para agrupar interações que têm algum relacionamento, mas não requerem uma ordem de realização nem há dependência entre elas.

3.3. As interações básicas

As *interações básicas* representam as ações básicas que o usuário pode executar utilizando os elementos da interface de usuário. São abstrações para ações como *clicar num botão*, *marcar ou desmarcar um checkbox*, *selecionar uma opção num combo-box*, etc. Aquelas que requererem algum processamento do sistema terão setas, rotuladas com seus respectivos nomes, saindo do espaço de interação e chegando à função de domínio correspondente.

A função de domínio, após sua execução provocada por uma ação do usuário, deverá retornar a algum espaço de interação para apresentar ao usuário, caso necessário, o resultado do seu processamento. Dessa forma, o usuário está percebendo quais são as reações do sistema face às suas ações. As interações básicas são dos seguintes tipos:

- *perceiveInformation*: usado para especificar uma interação básica de percepção de informação, como um texto ou figura, por exemplo;
- *enterInformation*: usado para especificar uma interação básica de entrada de informação, como uma caixa de texto, por exemplo;
- *selectInformation*: usado para especificar uma interação básica de seleção de informação, como uma lista, por exemplo;
- *activate*: usado para especificar uma interação básica de ativação de um controle, como um botão, por exemplo, que irá disparar a execução de algum processo na aplicação e/ou transitar para outro espaço de interação.

As interações básicas podem ser do tipo que manipulam informações e controlam o diálogo, como *enterInformation* e *selectInformation*, ou simplesmente que controlam o diálogo, como *activate*. Os tipos de transição são detalhadas na seção 3.5. Na figura 2, temos várias interações básicas distribuídas em vários espaços de interação.

3.4. As funções de domínio

Funções de domínio são usadas para representar os processos que desempenham as regras (ou lógica) de negócios da aplicação, ou seja, um processo executado pelo computador, capaz de mudar o estado da aplicação, através da manipulação das informações passada por meio dos espaços de interação. Uma função de domínio poderá produzir novas informações e retornar aos espaços de interação para informar o usuário dos resultados do processo. São representadas por retângulos com dois compartimentos, sendo o primeiro sua identificação e o segundo para aspectos da função que não serão detalhados neste trabalho. Na figura 2, temos várias funções de domínio, como *ConsultarSaldo*, *ConsultarExtrato* e *EfetuarTransferência*.

3.5. Controlando as transições no diálogo

As transições de um espaço de interação para outro podem ser disparadas por qualquer interação básica, devendo ser representada por uma linha sólida e possuir como rótulo o nome da interação básica que a disparou; ou por inatividade do usuário, neste caso, a seta deverá possuir rótulo com um valor temporal (ex. 50s, 3min) indicando que após aquele tempo de inatividade do usuário a transição irá ocorrer.

Existem duas situações de transição entre espaços de interação. Na primeira não há demanda por funcionalidades da aplicação, neste caso, temos apenas uma *navegação*. Esta situação é comum em espaços de interação que concentram chamadas a outros espaços, como janela com menus, por exemplo. A segunda é quando a transição demandar a execução de uma função de domínio, sendo tratada de duas maneiras. Quando partir do espaço de interação para a função de domínio, onde é considerada uma *ação* do usuário solicitando alguma funcionalidade do sistema, representada por uma linha sólida.

Quando a transição parte da função de domínio para o espaço de interação, é considerada uma *reação* do sistema à ação do usuário, podendo ser representada de duas formas: (a) se a execução da função de domínio terminar normalmente será representada por uma linha sólida, cujo rótulo poderá ser alguma informação retornada pela função (formato não detalhado neste trabalho) ou uma mensagem explícita do designer para o usuário; (b) no caso da execução terminar de maneira anormal, a linha será tracejada e seu rótulo poderá ser alguma indicação do motivo da falha, através de uma mensagem explícita do designer e/ou alguma informação do domínio ou da aplicação.

3.6. Sincronismo entre espaço de interação e função de domínio

Além das transições apresentadas na seção 3.5, existem excepcionalmente as setas de duplo sentido que são as linhas de sincronismo. Elas são destinadas exclusivamente para representar uma ligação síncrona entre um espaço de interação e uma função de domínio. Isto significa que, mesmo quando o processo está em execução, o usuário poderá observar seu progresso e/ou intervir sobre o mesmo. Neste caso, a seta não deve possuir rótulo, como exemplificado na figura 3-b, onde depois de ter iniciado o processo de cópia em *Copiar arquivos*, o usuário, além de acompanhar o progresso da cópia, poderá suspender, reiniciar ou cancelar o processo através do espaço de interação *Copiando arquivos*.

4. Modelando padrões

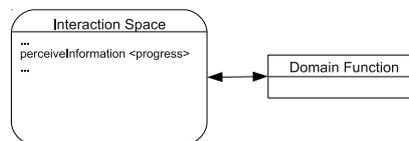
Nossa perspectiva é produzir uma biblioteca com modelos de interação para um conjunto de padrões, de maneira que seja possível fazer o reuso dessas estruturas na construção de modelos de interação para aplicações nas quais o designer busca garantir usabilidade da interface de usuário destes sistemas. Para isso, optamos por selecionar, inicialmente, aqueles que são identificados por Folmer [Folmer 2005] como *sensíveis à arquitetura*.

Em sua tese de doutorado, Folmer verificou que em processos cuja análise de usabilidade era realizada depois que a arquitetura estava definida, e que padrões de usabilidade eram empregados para solucionar os problemas encontrados, um conjunto de padrões traziam impactos diretos sobre a arquitetura. Dentre os impactos, estavam a necessidade de novas funcionalidades e, conseqüentemente, novos elementos arquiteturais e o rearranjo de uma quantidade considerável de elementos arquiteturais. A esse conjunto de padrões de usabilidade, Folmer deu o nome de padrões de usabilidade sensíveis à arquitetura [Folmer 2005].

Neste trabalho, por questões de espaço, apresentaremos nossa proposta de descrição apenas para dois padrões de usabilidade do catálogo de Tidwell [Tidwell 2005], ambos classificados como sensíveis à arquitetura [Folmer 2005]. A descrição de cada padrão terá cinco seções, sendo as quatro primeiras um resumo de Tidwell e a última, chamada *estrutura*, que irá descrever os principais elementos do modelo de interação do padrão e suas relações, usando a notação ALaDIM.

a) Indicador de progresso

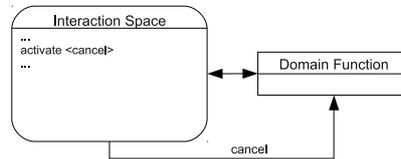
- **o que:** Mostrar ao usuário o progresso de uma operação demorada.
- **quando usar:** Uma operação demorada interrompe a interface de usuário por mais de dois segundos ou roda em segundo plano, ou algo semelhante.
- **por que:** Os usuários ficam impacientes quando a interface simplesmente pára. Não adianta apenas mudar o ponteiro do mouse para um relógio (ou ampulheta, por exemplo) se o restante da interface está travada, impossibilitando o usuário de saber quanto tempo terá que esperar.
- **como:** Mostre um indicador animado de quanto progresso tem sido feito. Usando uma notação textual ou gráfica (ou ambas) informe ao usuário: o que está acontecendo atualmente, qual a proporção da operação tem sido completada e quanto tempo resta.
- **estrutura:** Estrutura com um espaço de interação ligado por sincronismo à função de domínio em execução, sendo que o espaço de interação deve prover uma interação básica para o usuário perceber o progresso.



b) Cancelabilidade

- **o que:** Prover uma forma para cancelar imediatamente uma operação demorada, sem efeitos colaterais.
- **quando usar:** Uma operação demorada interrompe a interface de usuário por mais que dois segundos ou roda em segundo plano, ou algo semelhante. Também quando o usuário engajar numa operação que aparente ou literalmente interrompe a maioria das outras interações, como numa caixa de diálogo modal.
- **por que:** Os usuários mudam de ideia. Uma vez que uma operação demorada tiver iniciado, ele pode desejar pará-la, especialmente se souber o quão aquilo irá demorar. Ou mesmo se usuário tiver iniciado algo acidentalmente. Isto ajudará na prevenção e recuperação de erros. Fazendo o usuário se sentir mais à vontade em explorar a interface e experimentar coisas se ele souber que são canceláveis.
- **como:** Coloque um botão para cancelar diretamente na interface, próximo ao local onde aparecem os resultados da operação, rotule com as palavras “Parar” ou “Cancelar” e/ou coloque algum símbolo internacionalmente reconhecido. Quando usuário clicar no botão, cancele a operação imediatamente, se demorar mais que dois segundos o usuário pode duvidar se cancelou ou não a operação.

- **estrutura:** Estrutura com um espaço de interação ligado por sincronismo à função de domínio em execução, sendo que o espaço de interação deve prover uma interação básica para o usuário cancelar a execução.



Uma possível combinação entre os padrões *Indicador de Progresso* e *Cancelabilidade* é facilmente identificada. Um indicador de progresso informa ao usuário o quanto um determinado processo irá demorar para ser concluído, sendo essa informação determinante para sua decisão de interromper (ou não) o processo. Na figura 3-b, podemos perceber um espaço de interação ligado por sincronismo com uma função de domínio e que ele contém uma interação básica para o usuário perceber o progresso e outras para o usuário suspender, reiniciar e até cancelar o processo, demonstrando a aplicabilidade simultânea desses dois padrões no modelo de interação.

4.1. Aplicando padrões em modelos interação

Para exemplificar nossa proposta, observe os modelos de interação (figura 3) para interface de um sistema interativo, cujos requisitos foram retirados de um cenário onde “um usuário deseja copiar arquivos, sendo que ele deve primeiro informar os locais de origem e destino da cópia, para depois iniciar o processo”. Numa perspectiva centrada no sistema, o modelo da figura 3-a é suficiente para o design de uma solução para problema.

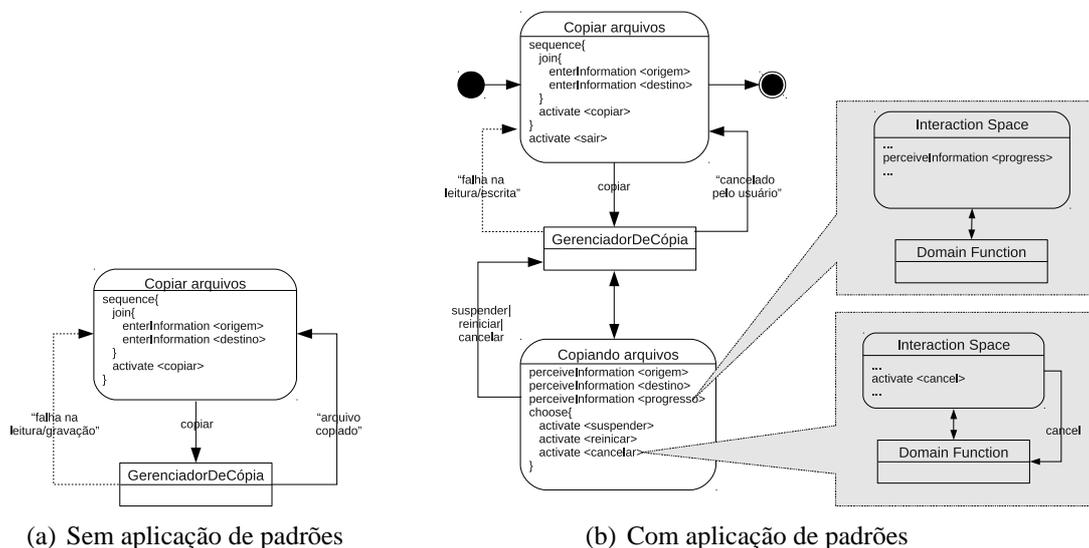


Figura 3. Modelo de interação para copiar arquivos.

Considerando que o processo de copiar arquivos poderá ser demorado, um especialista de IHC percebe que o modelo de interação da figura 3-a, pode ser melhorado através do uso dos padrões aplicáveis àquela situação. Então ele resolve inicialmente aplicar o padrão *Indicador de Progresso*. Como já visto na seção 4, este padrão tem uma forte ligação com o padrão *Cancelabilidade*, sendo finalmente aplicados os dois na solução proposta.

Estas ações, contribuem diretamente para um produto melhor, oferecendo maior grau de usabilidade, visto que aplicação agora oferece rico feedback e controle ao usuário.

Usando-se da biblioteca de modelos de padrões IHC, o designer da interação, acrescenta facilmente ao modelo os elementos que compõem os padrões aplicados, como pode ser observado na figura 3-b, onde temos várias interações básicas distribuídas nos espaços de interação *Copiar arquivos* e *Copiando arquivos*. No espaço de interação *Copiar arquivos*, o usuário deve primeiro informar origem e destino (em qualquer ordem), depois ativar o processo de cópia. Já no espaço de interação *Copiando arquivos*, o usuário poderá acompanhar e controlar o processo de cópia. Para isso ele percebe de e para onde está sendo copiado o arquivo, bem como o progresso da cópia e, se desejar, poderá ativar ações de suspender, reiniciar ou até cancelar o processo. Ainda neste exemplo, é possível perceber, como ALADIM, permite construir um modelo de interação representando os aspectos interativos e funcionais. Além de facilitar a inclusão dos componentes principais de padrões de usabilidade que estejam representados sob essa notação.

5. Considerações finais

Consideramos o modelo de interação um artefato integrador entre os profissionais das duas áreas, de maneira que ele deverá representar (sob a perspectiva da IHC) o diálogo entre o usuário e o sistema e (sob a perspectiva da ES) as informações e funcionalidades do sistema. Além disso, acreditamos que ele é mais expressivo para usuário entender o que o sistema lhe permite fazer, como ele (usuário) deverá interagir para fazer e como o sistema irá reagir diante de suas ações.

Nossa proposta de ter o modelo de interação como artefato de integração visa possibilitar aos profissionais de IHC, realizar (junto ao usuário) uma avaliação dos aspectos de usabilidade de maneira antecipada, deixando todo o processo menos oneroso e buscando garantir que as decisões de design sobre a interface e seu processo interativo sejam tomadas pelos profissionais de IHC, permitindo aos profissionais de engenharia de software manterem o foco em suas preocupações com os aspectos funcionais do sistema.

Aqui apresentamos uma notação para modelar os aspectos de interação entre usuário e um sistema interativo. Esta notação também foi usada para modelar estes aspectos presentes nos padrões de usabilidade, demonstrando a possibilidade de usar estes modelos como uma biblioteca de padrões, possibilitando melhorar reuso de seus componentes no design de soluções, através da composição de novos modelos.

Colocamos o modelo de interação como um meio termo entre cenários e protótipos, o que possibilita, quando necessário, que possam ser usados de maneira complementar. Este raciocínio também é válido quando usado para modelar a *estrutura* dos padrões de IHC, complementando as descrições textuais distribuídas nas outras seções. Aqui usamos o termo complementar por ele estar numa seção *extra* na descrição. Apesar disso, essa seção não substitui nenhuma das demais, pois como já visto, uma representação diagramática é considerada essencial na descrição de um padrão [Alexander et al. 1977].

Estamos trabalhando em outras frentes para sistematizar essa proposta. Atualmente está sendo desenvolvida uma ferramenta para suportar a construção do modelo usando a linguagem, além de manter integrada uma biblioteca de modelos para padrões de IHC, na perspectiva de oferecer ao designer uma forma fácil (*drag-and-drop*) de cons-

truir seus modelos, lançando mão de uma biblioteca de soluções comprovadas para os problemas semelhantes aos quais ele está propondo solução.

Referências

- Alexander, C. (1979). *The Timeless Way of Building*. Oxford University Press, New York.
- Alexander, C., Ishikawa, S., and Silverstein, M. (1977). *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press.
- Barbosa, S. D. J. and de Paula, M. G. (2003). Designing and evaluating interaction as conversation: A modeling language based on semiotic engineering. In Jorge, J. A., Nunes, N. J., and Falcão e Cunha, J., editors, *Interactive Systems. Design, Specification, and Verification, 10th International Workshop (DSV-IS'2003)*, volume 2844 of *Lecture Notes in Computer Science*, pages 16–33. Springer.
- Borchers, J. (2001). *A Pattern Approach to Interaction Design*. John Wiley & Sons, Inc., New York, NY, USA.
- Brown, J. and Marshall, S. (1998). Sharing human-computer interaction and software engineering design artifacts. In *OZCHI'98: Proceedings of the Australasian Conference on Computer Human Interaction*, page 53, Washington, DC, USA. IEEE Computer Society.
- Bueno, A. M. and Barbosa, S. D. J. (2007). Using an interaction-as-conversation diagram as a glue language for hci design patterns on the web. In *Proceedings of the 5th international conference on Task models and diagrams for users interface design, TAMODIA'06*, pages 122–136, Berlin, Heidelberg. Springer-Verlag.
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Michael, S. (2001). *Pattern-Oriented Software Architecture: A System of Patterns*, volume 1. John Wiley and Sons.
- da Silva, A. C., Silva, J. C. A., Penteadó, R. A. D., and da Silva, S. R. P. (2005). Integrando visões de ihc e de es por padrões no desenvolvimento por prototipação descartável. In *Proceedings of the 2005 Latin American conference on Human-computer interaction, CLIHC '05*, pages 223–234, New York, NY, USA. ACM.
- Folmer, E. (2005). *Software Architecture Analysis of Usability*. PhD thesis, University of Groningen.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. M. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley.
- Lavor, R. B. and Leite, J. C. (2011). Ferramenta para modelagem de interação e interface de usuário. In *XIV Congresso Ibero-americano em Software Engineering (CIbSE'11)*.
- Tidwell, J. (2002). <http://www.designinginterfaces.com/>. Acessado em setembro de 2010.
- Tidwell, J. (2005). *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly.
- Trøttemberg, H. (2008). Integrating dialog modeling and domain modeling - the case of diamodl and the eclipse modeling framework. *J. UCS*, 14(19):3265–3278.
- van Welie, M. (2001). <http://www.welie.com/patterns/>. Acessado em setembro de 2010.