

4

Integrando os Projetos de Interação e Interface

Hoover *et al.* (1991) apontaram que existem diferentes níveis de detalhe em design, onde os designers utilizam modelos incompletos para abstrair certos aspectos da solução separadamente (p.ex. um modelo cinético e um modelo geométrico de uma bicicleta), como forma de alcançar uma qualidade maior através de refinamentos sucessivos desses modelos. Nota-se que, em processos de design de uma forma geral, discutir aspectos pertencentes a um modelo durante a concepção de outro pode levar à negligência de detalhes em ambos.

Em IHC, quando se trata de design de interação e interface isso também pode ser verificado. Antecipar decisões de interface no momento do projeto da interação pode colocar uma importância desnecessária em detalhes que podem nem chegar a serem codificados na solução final, tornando o processo de design muito custoso. À medida que o projeto da interação progride, torna-se útil lançar mão de esboços de tela e elementos de interface para ajudar a visualizar a solução em um nível mais concreto.

Em consonância com a recomendação feita por Frederick Brooks, de que a interface deve ser projetada antes da programação do sistema [Brooks, 1975], acreditamos que a interação e a interface devem ser concebidas entre as fases de análise de requisitos e especificação da arquitetura do sistema, sempre com canais abertos permanentemente entre os membros da equipe de designers e desenvolvedores.

A MoLIC foi desenvolvida visando aproximar as utilizações mais comuns em IHC como cenários, modelos de tarefas e *storyboards*. Assim, ela pode ser utilizada para estruturar a interação descrita nos cenários e *storyboards* e ajudar na avaliação da consistência e compreensibilidade de um determinado design.

No entanto, em vez de apenas modelar a interação a partir dos possíveis cenários, acreditamos que o projeto da interação como um todo possa se beneficiar da associação de cenas e esboços de tela contendo as imagens de um *storyboard*. Dessa maneira, será possível retratar de forma concreta os

vários possíveis momentos da interação, dando a chance ao designer para se expressar num nível diferente daquele da modelagem da **estrutura** da conversa, permitindo possivelmente uma expressividade maior com relação ao **conteúdo**, por parte do designer.

A proposta de extensão da linguagem tira vantagem de uma representação mais estruturada como uma máquina de estados e a integra, como uma camada, no diagrama de interação. A linguagem MoLIC parece ser flexível ao ponto de absorver tal representação, já que ela representa os turnos usuário/sistema, claramente sequenciando a interação ao passo que permite que sejam especificados caminhos alternativos para um mesmo objetivo.

4.1

Proposta de extensão da MoLIC

Assim como apresentado na seção 3.4, onde cada elemento da MoLIC foi relacionado a elementos de interface, podemos concretizar a representação de cenas e ações dos usuários sobre estas cenas, utilizando esboços de tela e falas.

A Figura 4.1 conceitualiza a proposta de integrar o modelo da MoLIC com esboços de tela. A parte esquerda da figura mostra um diagrama “clássico” e a parte direita mostra o diagrama com esboços de tela inseridos nas cenas. Os esboços de tela são desenhados sem detalhes, apenas para ilustrar a ideia que o designer tem a respeito da representação do sistema.

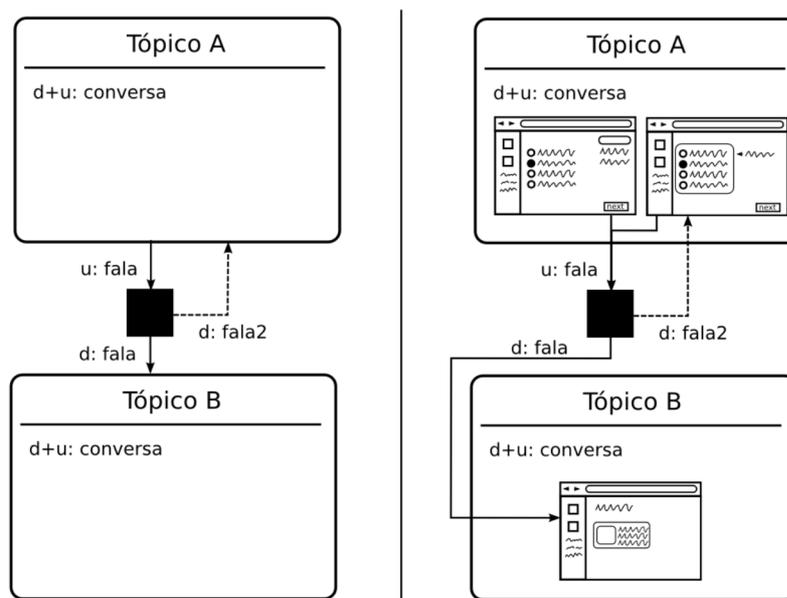


Figura 4.1: A MoLIC estendida com esboços de tela representando a busca do Google

Os esboços de tela dentro de cada cena compartilham o tópico daquela cena. Sob uma ótica de máquina de estados, cada esboço representa um estado distinto da interação. Com esse acoplamento, podemos então criar uma associação entre a conversa definida na MoLIC e uma representação executável definida através de telas conectadas entre si. Cada tela pode ser conectada às falas que têm suas origens ou fins na cena em que estão inclusas.

A proposta de extensão da MoLIC assume que todos os esboços de tela dentro de uma cena compartilham o mesmo **assunto**, variando no **conteúdo**, perspectiva através da qual o assunto é endereçado. Assim, as cenas podem ser vistas como grupos de estados que compartilham um discurso comum, enquanto esboços vistos individualmente podem ser interpretados como estados da interface propriamente dita.

Assim, falamos de extensão do modelo no sentido de acrescentar uma camada “gráfica” ao diagrama de interação – que pode ser visualizado tanto sem os esboços de tela, se desejarmos colocar o foco apenas na **estrutura** da conversa entre usuário e designer; como com a adição dos esboços, sem que a estrutura básica do diagrama seja alterada.

Ou seja, a extensão é aplicada às **cenar**s, pois com a adição de esboços visamos enriquecer a expressão de uma cena. O conceito de fala se mantém o mesmo – sendo a expressão do usuário ou designer a respeito da intenção em mudar o foco da conversa.

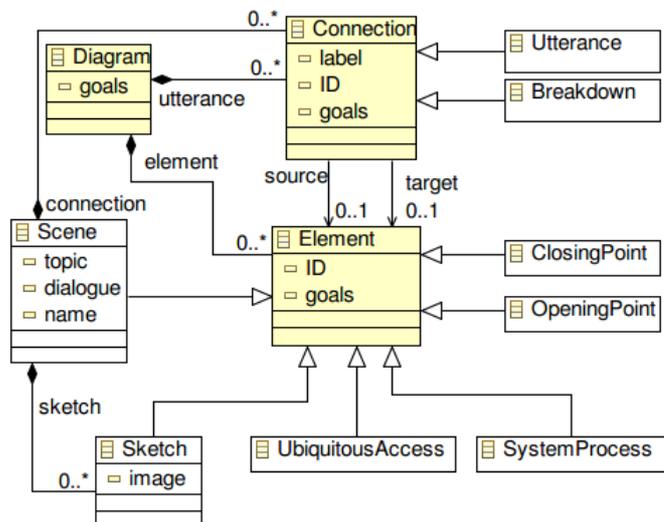


Figura 4.2: Metamodelo do diagrama MoLIC estendido

O metamodelo desenvolvido para a ferramenta *MoLIC Designer* (detalhada no capítulo 6) ilustra a extensão a nível de implementação. O metamodelo básico do diagrama é apresentado na Figura 4.2, que destaca uma

estrutura de grafo (*Diagram*) contendo nós (*Element*) e arcos (*Connection*). Dessa forma, os elementos básicos da linguagem são especializações do objeto *Element* e as falas, de *Connection*.

Desta forma, a extensão da MoLIC pode ser expressa pela adição do elemento *Sketch* (Esboço) agregado a *Scene*, sendo a MoLIC original representada por todos os outros elementos.

Um exemplo de aplicação da extensão pode ser construído com uma espécie de engenharia reversa da interação que acontece na busca do Google¹, tal como na Figura 4.3. Nele existem duas cenas, cada qual com estados diferentes: *Search*, onde usuário e preposto falam sobre os termos da busca (o que é representado pelo diálogo *d+u: search terms*) contendo os esboços de interface 1 e 2; e *View Results*, onde ambos falam sobre o resultado da busca proveniente dos termos, contendo os estados 3 e 4 que falam sobre “exibição de resultados”.

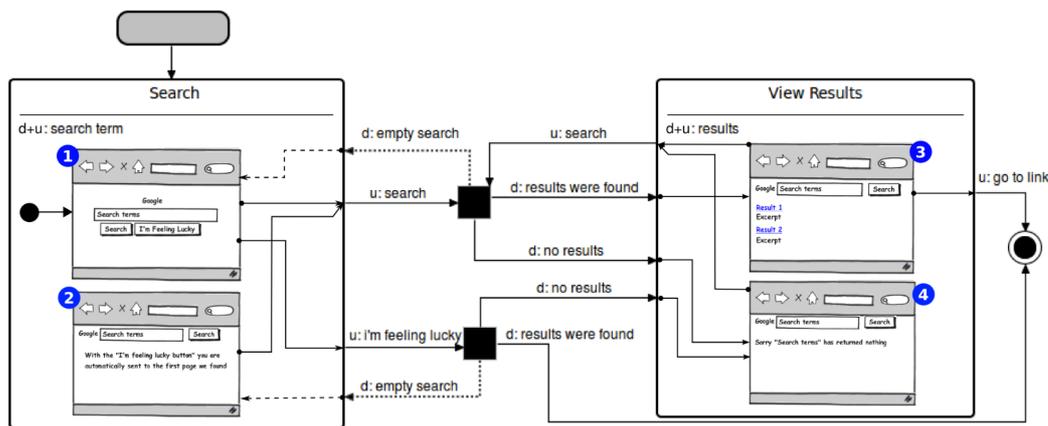


Figura 4.3: Um exemplo feito com a MoLIC estendida, representando a busca do *Google*

Da primeira cena no estado 1, o usuário pode dizer *search* ou ir direto ao primeiro resultado da busca, que é a funcionalidade *I'm Feeling Lucky*. As duas falas levam a processos e a partir daí, uma possível interpretação do que acontece pode ser:

- O usuário não digitou nenhum termo e falou “search” (busca vazia)** – Nesse caso, o usuário é levado à mesma página em que ele se encontra, sem nenhuma informação adicional sobre o que aconteceu. Uma interpretação possível sobre esse comportamento pode ser a de que

¹Esta engenharia reversa representa uma visão particular da conversa entre designer e usuário, já que não foi feita pelo designer responsável pelo sistema de busca do Google

o designer interpreta a “busca vazia” como uma ruptura. Ele pode ter imaginado que o usuário gostaria de buscar por algo (ou ele não estaria em um site de busca), mas que equivocadamente falou *search*. Assim, a escolha do designer é a de somente levar o usuário de volta para a mesma página;

- b. **O usuário não escreveu nenhum termo e falou “i’m feeling lucky”** – Nesse caso, a ruptura é mais explícita, pois o usuário é levado a uma explicação sobre o que significa exatamente “i’m feeling lucky”. É importante notar que esta funcionalidade só está disponível no estado 1;
- c. **O usuário digitou alguns termos e falou “search”, e uma ou mais páginas foram encontradas** – Nesse caso, os resultados são mostrados. Na perspectiva da MoLIC, isso é representado pelo preposto dizendo *d:results were found*, e exibindo os resultados na cena *View Results*, no esboço de tela respectivo. A partir desse estado, o usuário pode ir a um determinado resultado (falando *u: go to link*), ou buscar novamente (*u: search*);
- d. **O usuário digitou alguns termos, falou “search” ou “feeling lucky” e nenhum resultado foi encontrado** – Nesse caso, o resultado é o mesmo para ambos os casos: o usuário vê uma página com uma mensagem informando “nada foi encontrado”. Para nós, o designer não parece tratar essa situação como uma ruptura, e sim como um resultado esperado de uma busca.

A funcionalidade “I’m feeling lucky” se torna clara nesse exemplo, perpassando a cena de exibição dos resultados e indo direto para o fim da interação, onde o papel do *Google* de fato termina. Na nossa visão, a fala que o sistema usa para dizer que algo foi encontrado é justamente ir direto ao resultado.

A partir da cena *View Results*, o usuário pode iniciar uma conversa direcionada a outra busca, em vez de voltar para a cena *Search* anterior, o que parece razoável do ponto de vista da qualidade de IHC. Isto é representado pela fala *u: search* partindo da cena *View Results*.

É importante observar que o designer pode escolher, na ocorrência de uma ruptura, direcionar o usuário a diferentes caminhos de interação. Este é um dos aspectos interessantes da linguagem que pode vir a beneficiar a geração de protótipos, ainda que não-funcionais, mas detalhados no que diz respeito à interação.

Assim, o desdobramento mais evidente da extensão da MoLIC com a notação de esboços é a possibilidade de geração de *storylines* alternativas, derivadas do modelo, para um mesmo objetivo. Como exemplo, um possível cenário pode ser gerado, formando o *storyline* da Figura 4.4:

Chico quer fazer uma busca por “Michael Jackson” no site. Chico abre o site e digita equivocadamente “Miguel Jetson” (1) e efetua a busca, sem resultados (4). Corrige seus termos e efetua a busca com “Michael Jackson” (4). Desta vez recebe resultados (3), e então clica em um deles.

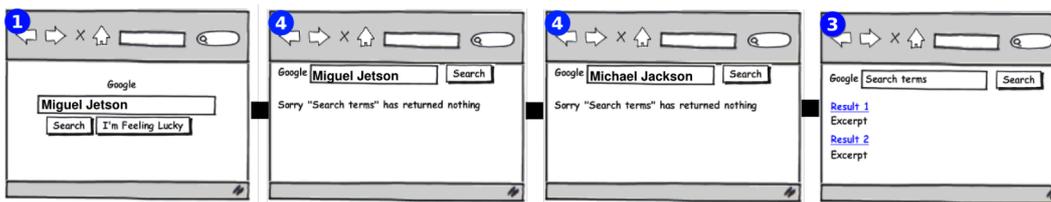


Figura 4.4: Um possível *storyline* para o cenário descrito

Partindo do conjunto de telas já definidas, seria interessante então permitir que o designer execute, de certa forma, os diferentes caminhos de interação através da ferramenta percebendo, talvez com mais clareza, caminhos que poderiam ser melhor explorados ou que são feitos de forma repetitiva – aspectos que emergem normalmente em simulações com protótipos.

Desta forma, espera-se dar a oportunidade ao designer para melhorar a sua solução interativa em um estágio muito inicial do projeto – a MoLIC estendida com esboços propicia tanto a percepção da visão geral da interação quanto permite testar como a interação poderá se dar em uma situação de uso legítima.

4.2

Discussão

Escolher entre a exibição de um diálogo de aviso ou simplesmente um texto destacado em uma página é o tipo de decisão de design que este trabalho quer apoiar com a junção entre a MoLIC e esboços de interface. Para tanto, procuramos abrir uma discussão sobre quão integrados devem (ou podem) estar os projetos de **interação** e **interface**, ao permitir a modelagem do comportamento do sistema seja feita em paralelo com o esboço da mensagem do designer, que se efetiva na interface.

Com o desenvolvimento da linguagem talvez sejamos capazes de identificar padrões de utilização e boas práticas no modelo de interação, e assim posamos reusá-los para dar aos usuários uma experiência interativa consistente [Bueno e Barbosa, 2006]. Se trata de apoiar computacionalmente as **decisões** de design.

Por exemplo, ao considerarmos o cenário *Busca na Web* baseada no *Google*, observamos ser uma prática comum às ferramentas de busca mais populares (como *Yahoo!*², *Ask*³ and *Bing*⁴) levar o usuário à mesma página que originou a busca, quando ele efetua uma busca sem termos – esse comportamento é facilmente verificável, basta que cliquemos *Search* nos sites de busca sem informar nenhum termo, ou simplesmente teclamos *Enter*.

Nos demais sistemas que encontramos no dia-a-dia, é comum verificamos exatamente o oposto: os usuários são explicitamente informados de que erraram, e (preferivelmente) são também informados como devem proceder para corrigir o erro, como por exemplo no *site Buscape*⁵, que exibe um alerta informando “*Por favor, digite um produto ou serviço para realizar a busca*”.

Um pouco mais revelador acerca do real construtor da mensagem são os *sites QueBarato*⁶, que também exibe um alerta: “*Por favor, seja um pouco mais específico. Digite ao menos 2 caracteres e tente novamente*”; e o Portal de Serviços e Informações do Governo do Brasil⁷, que exibe um alerta dizendo “*Você deve digitar pelo menos um parâmetro para pesquisa!*”.

Por último, também podemos observar *sites* que retornam **todos** os resultados possíveis, como é o caso da *ACM Digital Library*⁸, que exibe: “*Searching within The Guide for: no search specified. Found 1,394,228 of 1,394,228*” ou do *eBay*⁹, que leva o usuário para uma página com todas as categorias nas quais estão os **produtos** do site. Parece se tratar, nesse caso, de uma interpretação de que o usuário queira saber o quê pode vir a procurar.

Qualquer que seja o caso, o ponto é que todos os cenários citados são decisões **arbitrárias** de design para responder à mesma ação (uma busca vazia, sem termos). Estas decisões serão tomadas em algum momento, seja no projeto ou na codificação do sistema. Se não estão explícitas no projeto de alguma maneira, ficará a cargo finalmente do programador, quem por último

²<http://www.yahoo.com>

³<http://www.ask.com>

⁴<http://www.bing.com>

⁵<http://www.buscape.com.br> em Dezembro/2009

⁶<http://www.quebarato.com.br>

⁷<http://www.e.gov.br/> em Dezembro/2009

⁸<http://portal.acm.org/> em Dezembro/2009

⁹<http://www.ebay.com> em Dezembro/2009

implementará a comunicação com os usuários, decidir como (ou se) irá informar o usuário, ou mesmo se tratará a ação como um erro. Em outras palavras, se considerarmos que o foco do programador não é projetar a comunicação com os usuários, é possível que a solução implementada funcione nos casos para os quais ela foi projetada, mas confunda o usuário em situações não antecipadas pelo projetista.

Provavelmente restrições tão específicas, como a anteriormente citada que alerta para exatamente 2 caracteres na busca, são restrições de software enfrentadas pelo **programador**, que parece repassar imediatamente seu problema ao usuário, utilizando o seu próprio vocabulário (“caracteres” e “parâmetros”), talvez por julgar que será entendido por todos os usuários. Essa suposição é especialmente perigosa quando se trata de um portal governamental, que idealmente deveria falar a todas as pessoas do Brasil.

Para o designer, segundo a Engenharia Semiótica, a questão central no projeto de uma solução interativa gira em torno de transmitir a mensagem para o usuário sobre **como** ele pode interagir com esta solução para fazer o que quer/precisa/imagina, utilizando um conjunto de elementos que faça sentido para ele [de Souza, 2005].

Além disso, passar a mensagem de uma forma consistente é fundamental não apenas para “contar” ao usuário como o software deve ser usado segundo imaginado/preferido pelo designer, mas também para apoiar as maneiras possíveis pelas quais o artefato possa vir a ser utilizado por seus usuários. As mensagens de erro são, sem dúvida, importantes para “guiar” ou aproximar o usuário para estes caminhos preferenciais, uma vez que o erro faz parte da conversa entre o usuário e o sistema [Norman, 1988].

A linguagem MoLIC permite a representação destes caminhos preferenciais e permite que as situações de erro sejam bem definidas e contextualizadas. Porém, apesar de a linguagem permitir a modelagem de todas as conversas possíveis entre usuário e designer, a sua notação não possui a expressividade necessária para representar **tudo** o que é transmitido através da interface durante a interação, ou seja, ainda há um espaço muito grande entre a MoLIC e o produto interativo final.

Para os trabalhos anteriores, esse espaço não representou um problema muito grande, já que a MoLIC foi projetada para que o designer pense na interação de forma abstrata, suscitando a reflexão-em-ação [Schön, 1983] oriunda da modelagem da conversa entre ele e o usuário, e a consequente “troca de turnos” entre eles nessa conversa. O valor da MoLIC está no fato de

que o designer ganha conhecimento sobre o artefato sendo produzido, ou seja, o valor da MoLIC é inicialmente epistêmico.

No entanto, quando propomos seguir na direção de aproximar a **interação** da **interface**, tentando cruzar o golfo entre a modelagem abstrata da conversa e a representação do artefato, damos um enfoque mais **pragmático**.