

Capítulo

5

Design da Interação Humano-Computador com MoLIC

Simone Diniz Junqueira Barbosa, Bruno Santana da Silva
simone@inf.puc-rio.br, bruno@imd.ufrn.br

Abstract

This chapter presents a Modeling Language for Interaction as Conversation, named MoLIC, which is grounded in semiotic engineering. We will explore several alternative design solutions to illustrate the use of the language in diverse scenarios.

Resumo

Este capítulo apresenta a linguagem de modelagem da interação como conversa, MoLIC, fundamentada na engenharia semiótica. Ele demonstra como utilizar a linguagem para explorar soluções de design alternativas em diferentes cenários.

5.1 Introdução

O design da interação [Rogers *et al.* 2011] é uma atividade complexa que pode se beneficiar de representações que auxiliam o designer a organizar e registrar seu aprendizado sobre o problema e suas ideias de solução. Algumas dessas representações são fundamentadas em teorias [Carroll 2003]. Dentre as mais utilizadas no design de interação figuram *cenários* [Rosson e Carroll 2001], frequentemente utilizados em conjunto com *personas* [Cooper *et al.* 2007] e, para concretizar o design, *wireframes*, *mock-ups* ou esboços de tela. Menos frequente é o uso de modelos, cujo tipo mais amplamente utilizado é o de tarefas [Diaper e Stanton 2003]. Este tipo de modelo, entretanto, tradicionalmente pressupõe o desempenho humano competente e livre de erros.

Ancorada na teoria da engenharia semiótica [de Souza 2005], a linguagem MoLIC (do inglês *Modeling Language for Interaction as Conversation*) vem sendo desenvolvida há mais de uma década como uma ferramenta epistêmica para apoiar o design da interação como uma conversa entre o usuário e o preposto do designer [Paula 2003, Silva 2005, Araujo 2008, Barbosa e Silva 2010]. Diferente de outras representações e modelos, a MoLIC enfatiza o papel do designer como interlocutor durante a interação, através de seu preposto (a interface de usuário). A consequência

desta perspectiva é um enfoque na comunicação, para o usuário, das situações de uso vislumbradas pelo designer, incluindo situações em que o designer antevê que possa haver problemas durante a interação. Através da MoLIC, o designer representa e reflete sobre as possibilidades de interação que estarão disponíveis para os usuários alcançarem seus diversos objetivos com apoio da solução computacional sendo projetada. Vale observar que a MoLIC não pretende substituir representações e modelos que já são utilizados no design da interação, mas sim complementá-los.

Este capítulo está organizado da seguinte maneira. A próxima seção faz uma breve introdução à engenharia semiótica, teoria de IHC na qual a MoLIC se baseia. As seções 5.3 e 5.4 apresentam um ciclo de design de IHC e algumas representações utilizadas durante o processo de design. A seção 5.5 apresenta a linguagem MoLIC e a seção 5.6 ilustra sua aplicação em um projeto de IHC. A seção 5.7 descreve possíveis mapeamentos entre a interação representada em diagramas MoLIC e a interface com usuário. A seção 5.8 discute um possível uso da MoLIC na avaliação de IHC. Finalmente, a seção 5.9 conclui este capítulo com alguns comentários e direções de pesquisa futura.

5.2 Introdução à Engenharia Semiótica

A interação entre pessoas geralmente ocorre através de processos de comunicação ou de modos que podem ser compreendidos como comunicação. A engenharia semiótica [de Souza 2005] é uma teoria que caracteriza a interação humano-computador como um caso particular de comunicação humana mediada por sistemas computacionais. Durante o uso do sistema, o designer faz da interface o seu representante (preposto do designer) para comunicar ao usuário sua lógica de design; ou seja, sua interpretação sobre as necessidades, objetivos, valores e preferências dos usuários e sobre como o sistema projetado pode apoiá-los a atingirem seus objetivos. Esta comunicação designer-usuário ocorre enquanto o usuário se comunica com o próprio sistema, também por meio da interface (Figura 5.1.).

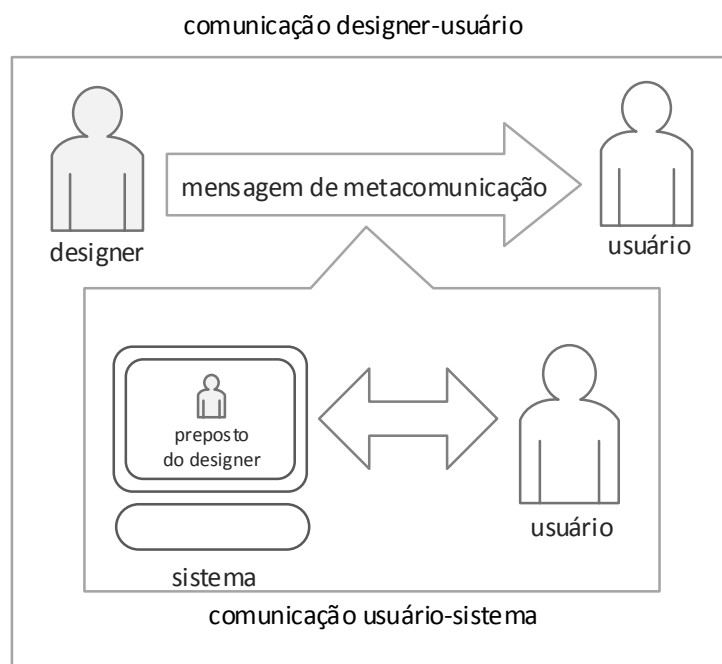


Figura 5.1. Processo de metacomunicação.

Como a comunicação designer-usuário busca explicar por que, onde, quando e como deveria ocorrer a comunicação usuário-sistema, dizemos que a comunicação designer-usuário é uma **metacomunicação** que transmite a **metamensagem** através da interface. O conteúdo da metamensagem pode ser parafraseada como:

“Esta é a minha interpretação sobre quem você é, o que eu entendi que você quer ou precisa fazer, de que formas prefere fazê-lo e por quê. Este é portanto o sistema que eu projetei para você, e esta é a forma que você pode ou deve usá-lo para atingir objetivos alinhados com a minha visão.” (de Souza 2005:84)

Para transmitir a metamensagem de forma efetiva e eficiente, o designer deve escolher bem os signos representados na interface. **Signos** são as palavras, imagens, símbolos, gestos ou qualquer outro recurso utilizado na comunicação. A cadeia de interpretações de um signo, denominada **semiose**, é um processo pelo qual alguém percebe sua representação e gera uma ideia ou pensamento. Esse processo é influenciado pela experiência e conhecimento prévio que as pessoas possuem, assim como pelo contexto e cultura nos quais estão inseridas no momento. Os signos escolhidos pelo designer devem motivar interpretações dos usuários compatíveis com as dele, para que os usuários compreendam a metamensagem e sejam capazes de fazer um uso efetivo e criativo do sistema.

Quando os signos são utilizados de forma sistemática, eles formam um sistema de signos ou uma linguagem que facilita seu emprego em processos de comunicação. Desse modo, o designer deve definir uma linguagem de interface composta pelos signos que o usuário poderá utilizar para se comunicar com o sistema. Os signos na interface e os respectivos comportamentos do sistema devem ser consistentes e coerentes em todo o sistema para que o usuário perceba, interprete e aprenda a usar adequadamente a linguagem de interface. Se uma representação tiver dois significados (e.g. “fechar” em um lugar na interface termina a atividade sem salvar e em outro lugar salva e termina a atividade) ou se dois significados forem representados da mesma forma (e.g. remover um arquivo de áudio da biblioteca ou do computador pode ser acionado pelo mesmo comando “remover”), o usuário provavelmente terá dúvida sobre o comportamento do sistema.

Mesmo que o designer faça um bom projeto, não é possível prever ou determinar os significados que os usuários darão a cada signo na interface. Além disso, a semiose é ilimitada, ou seja, enquanto a pessoa desejar e tiver condições, ela pode reinterpretar um signo, evoluindo seu significado na sua mente. Enquanto isso, o sistema continua com seus signos congelados conforme foram codificados. O processo de comunicação designer-usuário pode falhar, ou seja, a interpretação do usuário pode se distanciar daquela pretendida pelo designer e codificada no sistema. Essas rupturas de comunicação ocorrem quando a interface do sistema (preposto do designer) não consegue responder adequadamente as requisições do usuário ou quando os usuários cometem erros ou equívocos. O designer deve tentar antecipar situações em que rupturas de comunicação podem ocorrer e oferecer ao usuário mecanismos para se recuperar dessas rupturas e continuar a interação para atingir seus objetivos. Por exemplo, deve informar ao usuário quando este fornecer um dado no formato errado, ou quando tentar acessar a internet e a conexão não estiver disponível.

A linguagem MoLIC motiva o designer a se manter num papel comunicativo ativo durante o projeto da interação através do seu preposto. Ela permite que o designer represente todas as possíveis conversas que o usuário terá com seu preposto, ou seja, ele poderá descrever todos os possíveis rumos da conversa (ou caminhos de interação), incluindo rumos alternativos para atingir o mesmo objetivo e rumos para a recuperação de rupturas de comunicação.

5.3 Ciclo de Design de IHC

Em termos gerais, um processo de design envolve atividades de análise, síntese e avaliação [Lawson 2006]. No caso específico de IHC, Rogers, Sharp e Preece (2011) segmentam a atividade de síntese em duas atividades: (re)design conceitual e construção de uma versão interativa.

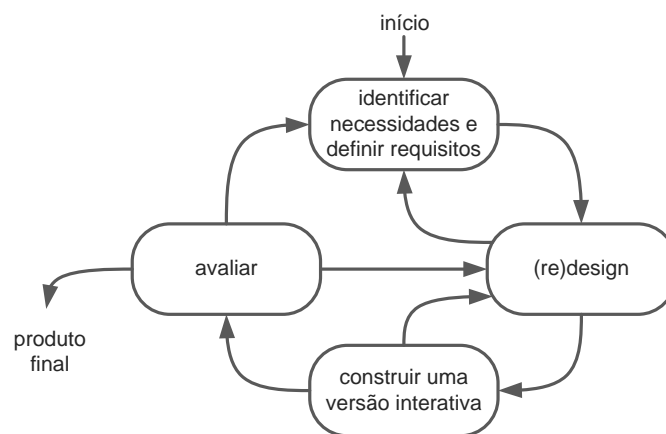


Figura 5.2. Modelo simples de processo de design de IHC (adaptado de Rogers et al., 2011).

Como a maioria dos processos de design atuais, o modelo de processo proposto por Rogers *et al.* (2011) é altamente iterativo: ao realizar cada atividade, o designer pode identificar a necessidade de retornar a uma atividade anterior para ampliar, refinar ou retificar algum entendimento ou alguma parte do artefato produzido até então. Em geral, são utilizadas personas e cenários como produto da atividade de identificação de necessidades e definição de requisitos; cenários, modelos de tarefa e de interação e esboços de tela como produto da atividade de (re)design; e protótipos (em papel ou funcionais) como produto da atividade de construção de uma versão interativa. Essas representações são descritas na próxima seção.

5.4 Representações de design de interação e de interface

As subseções a seguir apresentam as seguintes representações utilizadas no design de interação: *cenários* [Rosson & Carroll 2001], *personas* [Cooper et al. 2007], modelos de tarefas [Diaper & Stanton 2003] e *mock-ups* ou esboços de tela.

5.4.1 Cenários

Um cenário é uma narrativa sobre pessoas realizando uma atividade [Rosson & Carroll 2002]. Cenários são utilizados principalmente para descrever situações, problemáticas ou não, que contribuam para o entendimento das atividades dos usuários, visando a apoiar a definição dos requisitos de um sistema; e para descrever situações alternativas

que podem vir a ocorrer após a introdução da tecnologia, visando a apoiar a tomada de decisões de design e a avaliar se um produto satisfará as necessidades e desejos dos usuários. Segundo Rosson e Carroll (2002), os elementos característicos de um cenário são:

- *ambiente ou contexto*: detalhes da situação que motivam ou explicam os objetivos,
- ações e reações dos atores do cenário;
- *atores*: pessoas (incluindo suas características relevantes) interagindo com o computador ou com o ambiente;
- *objetivos*: o que motiva o planejamento e as ações realizadas pelos atores;
- *planejamento*: atividade mental visando a transformar um objetivo em um conjunto de ações;
- *ações*: comportamento observável, ou seja, o que os atores fazem;
- *eventos*: ações externas ou reações produzidas pelo computador que sejam importantes para o cenário (mesmo que estejam ocultas ao ator);
- *avaliação*: atividade mental visando a interpretar a situação.

Veja a seguir um exemplo de cenário que evidencia um problema em um sistema bancário:¹

Transferência bancária – Qual é mesmo o número daquela conta?
 Dia 10 chegou *evento*, e Marta *ator* se lembra *evento* que precisa transferir o dinheiro do aluguel *objetivo* para Ana *ator*. Como está longe de uma agência bancária e não sabe se daria tempo de pegar uma agência aberta, pois já eram 15h30 *contexto*, Marta decide efetuar a transferência pela Internet *planejamento*. Ela entra no site do seu banco *ação*, mas ao iniciar o processo de transferência *ação*, percebe que não está com sua agenda em mãos, e como não memorizou os dados da conta de Ana, não pode fazer a transferência naquele momento *avaliação*. Angustitada pelo risco de ter que pagar uma multa *contexto*, Marta tenta ligar para Ana *ação*, mas o telefone dela está fora da área de alcance. Marta então abandona todos os seus planos para aquela tarde e corre para casa, na tentativa de pegar sua agenda e efetuar a transferência a tempo *planejamento*.

Para solucionar o “problema” apontado pelo cenário anterior, pode-se projetar um mecanismo de cadastramento de contas favoritas, como ilustrado no seguinte cenário:

¹ Os elementos citados por Rosson & Carroll estão marcados nos cenários por questões ilustrativas, mas devem ser omitidos ao apresentar os cenários para discussão com clientes, usuários e demais partes interessadas no sistema.

Transferência bancária para conta pré-cadastrada

Dia 10 chegou ^{evento} e Marta ^{ator} se lembra ^{evento} que precisa transferir o dinheiro do aluguel ^{objetivo} para Ana ^{ator}. Como está longe de uma agência bancária e não sabe se daria tempo de pegar uma agência aberta ^{contexto}, Marta decide efetuar a transferência pela Internet ^{planejamento}. Ela entra no site do seu banco ^{ação}, inicia o processo de transferência ^{ação}, indica a conta de Ana de uma lista de contas pré-cadastradas ^{ação}, informa a quantia a ser transferida ^{ação} e confirma a operação com sua senha ^{ação}. Ao conferir que a transferência foi efetuada ^{avaliação}, ela envia uma confirmação da transferência por e-mail para Ana e para ela própria ^{ação}, verifica o seu saldo para confirmar o quanto ainda resta de dinheiro na conta ^{avaliação} e sai do sistema ^{ação}.

5.4.2 Personas

Cooper trouxe o conceito de personas das artes cênicas para o desenvolvimento de software, visando a manter o foco nas características de cada grupo de usuários finais durante discussões de design [Cooper et al. 2007]. Cada grupo de usuário é representado por uma persona, um personagem fictício, arquétipo hipotético daquele grupo. Ao se utilizar personas durante o processo de design, evita-se considerar o usuário como um ser “elástico” e com características distintas a cada momento do design, o que pode causar inconsistências na interação e na interface sendo projetadas. Embora hipotética, uma persona é fruto de extensa e rigorosa análise dos usuários, seus objetivos e suas atividades.

Courage e Baxter (2005) argumentam que uma persona deve ser caracterizada pelos seguintes elementos:

- *identidade*: nome, sobrenome, idade, foto e outros dados demográficos que tornem a persona realista e memorável, incluindo, se possível, uma citação que a sumarie (por exemplo “Só aprendo fazendo”);
- *status*: indicação de usuário primário ou secundário do seu sistema, ou uma parte interessada (*stakeholder*), alguém que não vai utilizar o produto mas pode ser afetado por ele;
- *objetivos*: objetivos práticos, ou seja, aqueles que a persona deseja alcançar com o sistema; objetivos pessoais (por exemplo, não cometer erros, conseguir realizar uma quantidade de trabalho razoável); e objetivos corporativos, que são os do ambiente de trabalho da persona (para o caso de sistema corporativos);
- *habilidades*: especialidades da persona, incluindo educação, treinamento e competências específicas, fortemente relacionadas com o seu produto específico ou não;
- *tarefas*: desde as tarefas básicas até as tarefas críticas que a persona realiza, incluindo sua frequência, importância e duração;
- *relacionamentos*: com quem a persona precisa interagir, mediada ou não pelo sistema; quais as relações de poder, de dependência de informação, entre outras;
- *requisitos*: tudo de que a persona precisa, em termos de informações, atividades prévias, consultas a ou autorizações de outras personas, entre outros;
- *expectativas*: a forma como a persona acredita que o produto deveria funcionar e como as informações no seu domínio ou trabalho deveriam ser organizadas.

A seguir ilustramos duas personas que representam grupos de usuários distintos de um sistema de *e-banking*.



Marta Oliveira, vendedora, “eficiência é fundamental”

Marta Oliveira, 37 anos, é vendedora experiente da loja Vista-se Bem especializada em roupas, calçados e acessórios. Ela está acostumada a usar Internet em diferentes dispositivos para interagir com as pessoas. Na vida pessoal, não deixa de se comunicar com amigos e familiares sempre que encontra alguns minutos livres. Costuma usar diferentes aplicativos de comunicação e rede social, seja em trânsito no smartphone, seja no seu PC em casa. No trabalho, as redes sociais e e-mail são um canal importante para manter contato com seus clientes e construir um bom relacionamento. Ela aprendeu a usar essas TICs para apresentar seus produtos, ouvir as necessidades dos clientes e manter o valor da marca de sua loja. Sempre atarefada, evita utilizar qualquer TIC pouco eficiente, pois não tem receio de experimentar alternativas interessantes e gosta de aprender fazendo.

Na correria do dia-a-dia, Marta prefere usar os serviços bancários pela Internet. Suas prioridades são pagar suas contas em dia e investir o pouco que sobra todo mês. Algumas dessas contas mensais são pagas via boleto bancário; outras, no entanto, ocorrem por transferência bancária, como o aluguel, por exemplo. Sempre que possível, ela procura formas de gastar menos tempo com os serviços bancários para poder cuidar de outras coisas. Ajudaria muito se os sistemas lembrassem de dados importantes para ela, como o número do cartão de crédito e os dados bancários do dono do seu apartamento.



Pedro Santos, padeiro, “segurança, base da prestação de serviços”

Pedro Santos é um jovem padeiro muito preocupado em oferecer um produto de qualidade, que não prejudique a saúde dos seus clientes. Acostumado a interagir com os clientes pessoalmente na loja, ele busca identificar a reação a novos produtos, perceber a satisfação com produtos antigos, ou simplesmente bater um papo para ser mais conhecido pela clientela. Ele usa computador somente para o necessário, como ler notícias, enviar e-mail e arrisca alguma participação nas mídias sociais para interagir mais com familiares distantes. Sempre que pode, prefere ser atendido por uma pessoa no caixa do seu banco, apesar de ter usado o caixa eletrônico em algumas situações.

Ultimamente, Pedro tem encontrado dificuldades para ir até sua agência bancária. Familiares têm indicado o uso do site do banco para realizar algumas operações simples. Meio desconfiado, Pedro resolveu cadastrar sua senha de Internet para uma emergência, mas ainda com receio da segurança das operações. Nos dias mais atarefados, ele pretende arriscar apenas pagamento de algumas contas e talvez alguma transferência bancária para começar.

Personas e cenários podem ser utilizados no design segundo uma perspectiva da engenharia semiótica, pois ajudam a definir a primeira parte da metamensagem designer–usuário: “Este é o meu (designer) entendimento de quem você (usuário) é, do que aprendi que você quer ou precisa fazer, de que maneiras prefere fazer, e por quê.” [de Souza 2005:25]. Para alavancar ainda mais a natureza epistêmica de cenários, Paula (2003) propõe complementá-los com perguntas que revelem a intenção do designer ao elaborar os cenários, que pode envolver questões que o designer almeja descobrir, explorar ou ratificar junto aos usuários.

5.4.3 Modelos de Tarefa

A análise de tarefas investiga o trabalho dos usuários para compreender suas necessidades: O que eles precisam fazer? Como eles costumam fazer? Por quê? O escopo desta análise costuma se restringir às principais tarefas dos usuários, com foco nos fatores que influenciam seu desempenho. Em geral, a análise de tarefas inicia com a investigação dos objetivos dos usuários, e prossegue identificando como estes objetivos são atingidos através de decomposição de tarefas: uma tarefa maior pode ser realizada quando um conjunto de tarefas menores é concluído. Esse conhecimento sobre as tarefas permite projetar um sistema computacional que contribua para a satisfação das necessidades dos usuários. Existem vários métodos e modelos de análise de tarefas [Diaper & Stanton 2003], dos quais podemos destacar a Análise Hierárquica de Tarefas (AHT ou, em inglês, HTA – *Hierarchical Task Analysis* [Annett 2003; Annett & Duncan 1967]), o GOMS (*Goals, Operators, Methods, and Selection Rules* [Card, Moran & Newell 1983, Kieras 2003]) e o ConcurTaskTrees (CTT [Paternò, 1999]). Neste trabalho discutimos brevemente o primeiro. Boa parte dos métodos de análise de tarefas propõem modelos específicos para representar as tarefas sendo analisadas.

A Análise Hierárquica de Tarefas [Annett 2003; Annett & Duncan 1967] investiga a relação entre o que as pessoas fazem, por que e as consequências de não fazê-lo corretamente. Nesta proposta, uma **tarefa** começa sendo definida por um **objetivo** mais abstrato, que representa um objetivo do usuário. Este objetivo abstrato será decomposto em **subobjetivos** até chegar em **operações** que possam ser executadas diretamente no sistema. Um plano determina quais subobjetivos e a ordem em que eles devem ser alcançados para atender a um objetivo maior. Os subobjetivos podem ser sequenciais (o segundo só é atingido depois do primeiro), paralelos (mais de um subobjetivo pode ser atingido ao mesmo tempo) ou de seleção (o usuário pode atingir um dos subobjetivos dependendo das circunstâncias). A Figura 5.3 apresenta a representação gráfica os elementos da AHT.

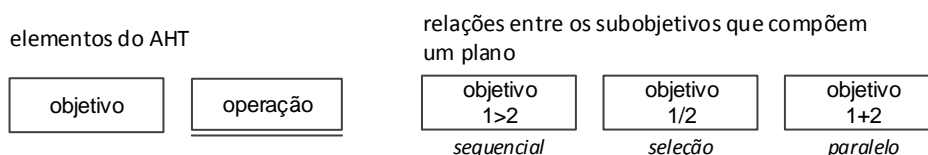


Figura 5.3. Elementos do modelo de Análise Hierárquica de Tarefas.

A tarefa de transferir dinheiro entre contas bancárias abordada anteriormente pode ser ilustrada pelo modelo de tarefas da Figura 5.4. Para transferir o dinheiro, o usuário precisa entrar no sistema do banco (se já não o tiver feito), informar os dados conta de origem, iniciar a transferência, depois informar a conta de destino e informar o valor a ser transferido (essas duas tarefas em qualquer ordem), para finalmente confirmar a

operação. Para informar a conta de destino, o usuário pode seguir uma de duas estratégias: informar o número da agência e da conta de destino, ou escolher uma conta pré-cadastrada. Para confirmar a operação, o usuário primeiro informa novamente a senha, depois confere o resultado da operação com saldo atualizado, e por fim ele envia a confirmação por e-mail para as pessoas envolvidas.

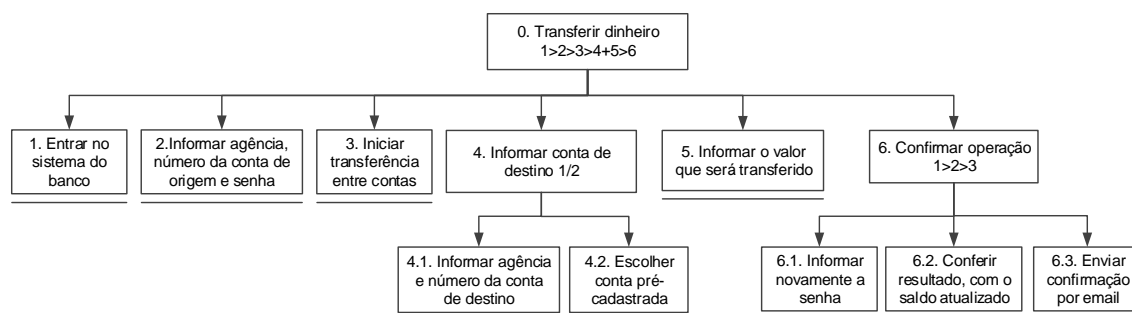


Figura 5.4. Exemplo de modelo de tarefa para transferir dinheiro entre contas bancárias.

Modelos de tarefas são muito úteis para analisar tarefas bem estruturadas e definidas. Entretanto, eles costumam apresentar limitações quando as tarefas não se encaixam bem em uma estrutura hierárquica. Por exemplo, ciclos, situações de erro e outros tipos de relação não hierárquica entre tarefas são complicados de representar nesses modelos.

5.4.4 Esboços de Tela

Uma representação bastante comum no design de IHC são os esboços de tela, também conhecidos como *wireframes*, *mock-ups*, protótipos ou maquetes. Eles representam elementos de interface com os quais o usuário é capaz de interagir (os *widgets*). Estes elementos são organizados de uma maneira lógica na interface nas dimensões de espaço e tempo, para permitir o usuário tirar proveito do sistema de acordo com seus objetivos. Em interfaces gráficas, os elementos de interface são organizados em telas que vão sendo substituídas conforme a interação progride. As transições entre telas também podem estar representadas nos esboços.

Os esboços de tela podem ser classificados pelo seu grau de fidelidade. Um esboço de baixa fidelidade representa (os elementos de) a interface sem muita preocupação com seus aspectos gráficos. Já um esboço de alta fidelidade representa a interface considerando os aspectos gráficos, como tamanho, posição, cor, fonte e outros detalhes visuais de cada elemento. Os dois tipos de esboços podem ser elaborados manualmente ou em alguma ferramenta computacional, como o Pencil Project,² Balsamiq Mockups,³ Microsoft Visio⁴ ou Axure.⁵ Todavia, os esboços de alta fidelidade geralmente são construídos com alguma ferramenta computacional.

A Figura 5.1 apresenta um exemplo de esboço de tela de baixa fidelidade para um sistema web que permite ao usuário realizar transferência de dinheiro entre contas bancárias. Em cada tela estão representados os elementos de interface mais relevantes para o objetivo de transferência bancária. Além disso, estão representadas algumas transições entre essas telas.

² <http://pencil.evolus.vn/>

³ <https://balsamiq.com/>

⁴ <http://office.microsoft.com/en-us/visio/e>

⁵ <http://www.axure.com/>

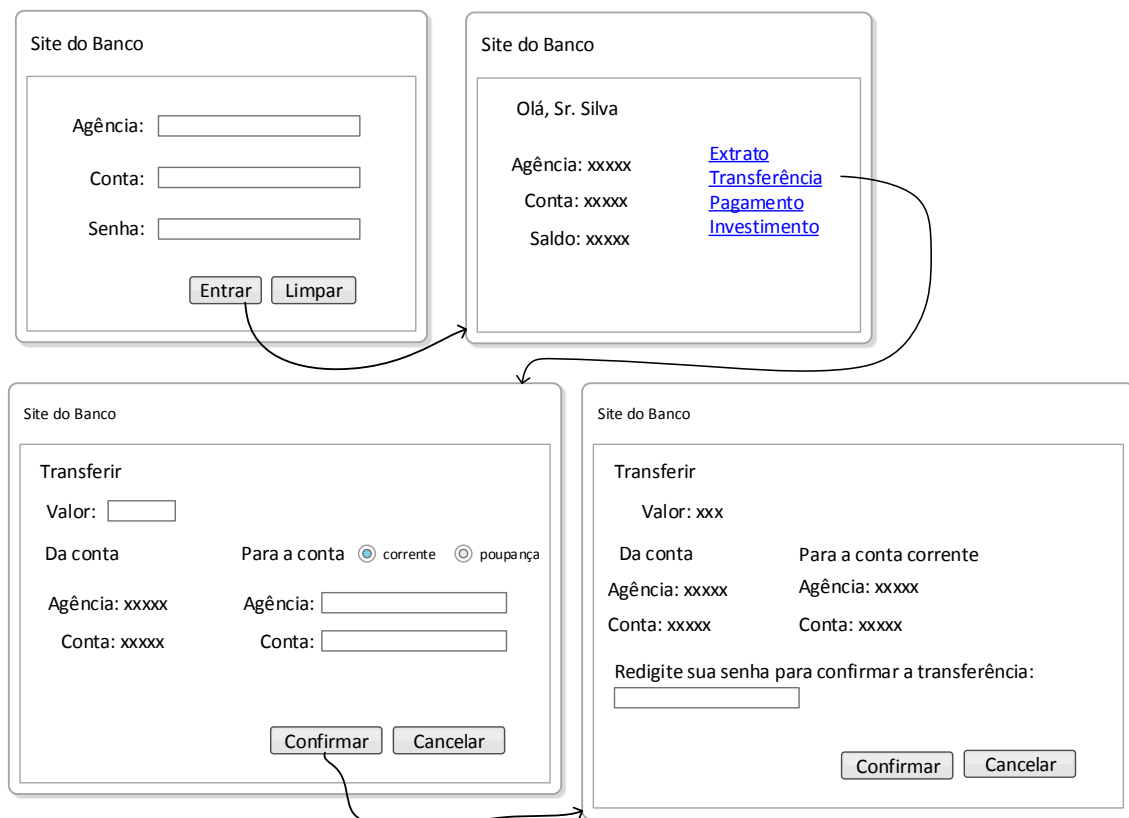


Figura 5.5. Exemplo de esboço de tela para transferir dinheiro entre contas.

Por ser uma representação da interface concreta, próxima daquilo com o que o usuário vai interagir, os esboços de tela são bastante utilizados com insumo de avaliações formativas junto a usuários e demais interessados no sistema.

5.5 Interação como uma conversa

A MoLIC adota uma perspectiva de interação como uma conversa entre o usuário e o preposto do designer. Nessa conversa, usuário e preposto do designer (representado pela interface do sistema) endereçam certos assuntos, trocando turnos em diálogos e falas, voltados para o alcance dos objetivos do usuário [Paula 2003, Silva 2005].

Por exemplo, uma conversa para efetuar a transferência bancária representada pelo cenário da seção 5.4.1 poderia ser descrita como a seguir:

- U:⁶ Bom dia.
 D: Qual conta deseja manipular?
 U: Conta 9876-5 da agência 345-6, senha *****.
 D: O que deseja fazer?
 U: Quero efetuar uma transferência.
 D: Para qual conta?
 U: Para a conta pré-cadastrada de Ana Barros.
 D: Qual é o valor?
 U: R\$ 2.500,00.

⁶ U representa uma fala do usuário, e D representa uma fala do preposto do designer.

D: Por favor confirme a operação com sua senha.

U: *****.

D: Sua transferência de R\$ 2.500,00 para a conta de Ana Barros (agência 8352-4 conta corrente 41632-9) foi efetuada com sucesso.

Esta conversa pode ser segmentada de diferentes formas. Após a identificação da conta a ser manipulada, por exemplo, a transferência pode ser considerada como tendo apenas um tópico, Efetuar transferência, ou dois tópicos, Definir dados para a transferência e Confirmar a transferência. Nesse último caso, o sistema pode verificar, entre a definição dos dados e a confirmação da transferência, se os dados da conta bancária de destino são válidos e se o saldo do usuário é suficiente para efetuar a transferência, por exemplo. A Figura 5.6 ilustra essas duas situações.

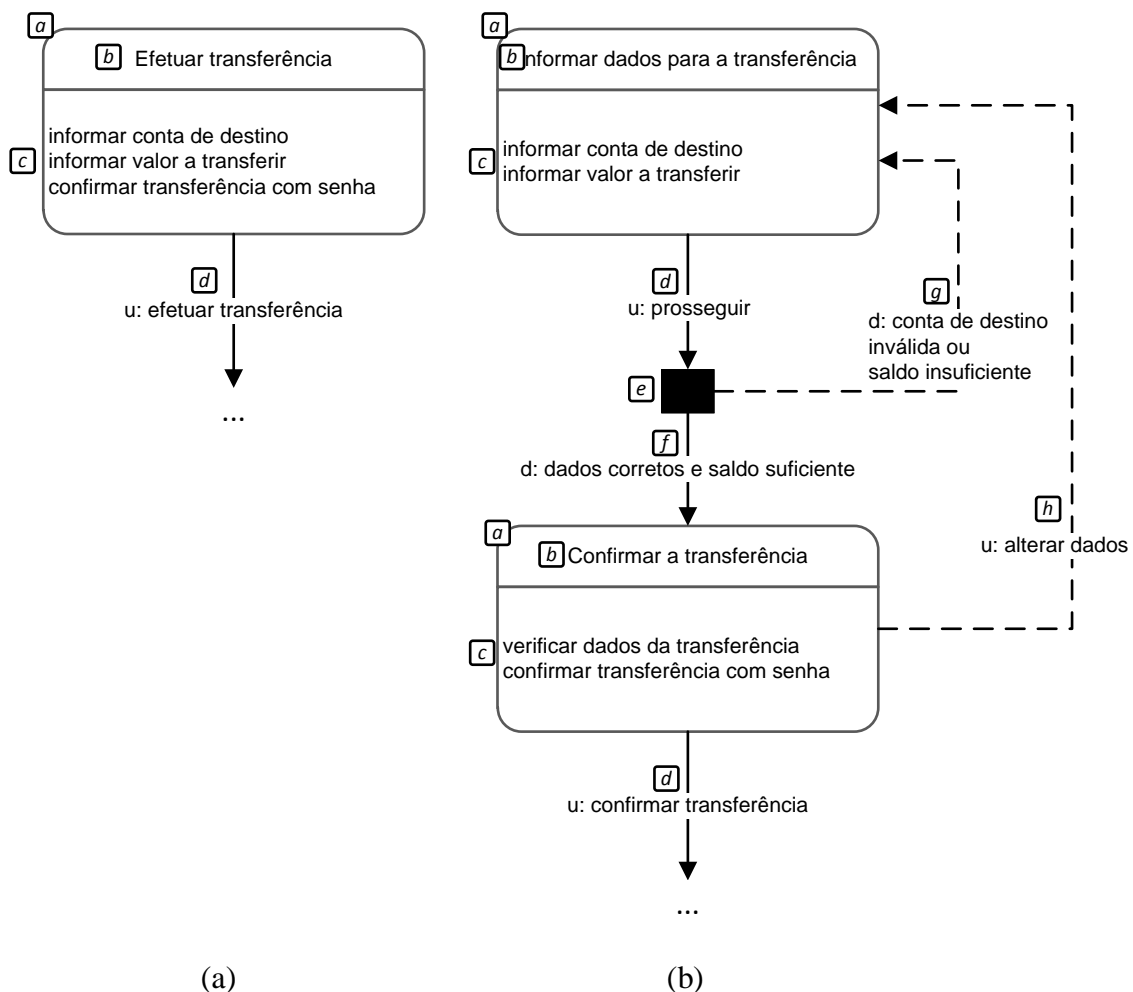


Figura 5.6. Duas possibilidades de segmentação de uma conversa para efetuar transferência bancária.

A Figura 5.6 apresenta os seguintes elementos da MoLIC:

- a) **cena**: palco da conversa entre usuário e preposto do designer sobre um assunto (tópico) específico
- representação*: retângulo com bordas arredondadas

- b) **tópico da cena**: assunto da conversa travada na cena, representado por uma frase no infinitivo, que pode ser lida como uma fala do preposto do designer para o usuário, no formato “Você agora deve <tópico>.”
exemplos: Efetuar transferência, Informar dados para a transferência, Confirmar a transferência
representação: texto no primeiro compartimento da cena
- c) **diálogos da cena**: unidades da conversa que focam diferentes aspectos do tópico da cena
exemplos: informar conta de destino, informar valor, confirmar transferência com senha, verificar dados da transferência
representação: texto no segundo compartimento da cena
- d) **falas do usuário para troca de turno (falas de transição do usuário)**: troca de turno, em que o usuário passa o controle da conversa para o preposto do designer
exemplos: u: efetuar transferência, u: prosseguir, u: confirmar transferência
representação: linhas direcionadas sólidas com rótulo u:
- e) **processamento do sistema**: momento oculto para o usuário, onde ele espera que o sistema esteja realizando a operação que o usuário solicitou ao preposto
representação: quadrado com fundo preto, indicando que, do ponto de vista do usuário, se trata de uma caixa-preta, ou seja, ele não sabe o que está ocorrendo durante o processamento
- f) **falas do preposto do designer para troca de turno (falas de transição do preposto do designer)**: troca de turno, em que o preposto do designer informa ao usuário sobre o resultado de um processamento, e pode devolver o controle da conversa para o usuário – se o destino da fala for uma cena – ou prosseguir para outros processamentos
exemplo: d: dados corretos e saldo suficiente
representação: linhas direcionadas sólidas com rótulo d:
- g) **falas do preposto do designer para recuperação de ruptura**: troca de turno, em que o preposto do designer informa o usuário sobre um resultado inesperado de um processamento do sistema, e pode devolver o controle da conversa para o usuário – se o destino da fala for uma cena
exemplo: d: conta de destino inválida ou saldo insuficiente
representação: linhas direcionadas tracejadas com rótulo d:
- h) **falas do usuário para recuperação de ruptura**: troca de turno, em que o usuário percebe que se enganou e muda de ideia, retornando para uma cena em que possa retificar parte da conversa travada previamente
exemplo: u: alterar dados
representação: linhas direcionadas tracejadas com rótulo u:

Em linhas gerais, a conversa se inicia quando o usuário entra no sistema, e se encerra quando o usuário sai do sistema. Para indicar esses momentos, utilizamos **pontos de abertura** (Figura 5.7a) e **de encerramento** (Figura 5.7b) da conversa.

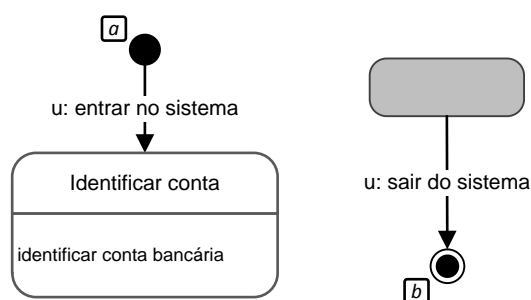


Figura 5.7 Pontos de abertura (a) e encerramento (b) da conversa.

Em alguns sistemas, pode-se projetar mais de um ponto de abertura. No caso do sistema bancário, por exemplo, pode-se imaginar um segundo ponto de abertura em que, utilizando um leitor de cartões, o usuário não precisa identificar sua conta, apenas informar sua senha.

Os diagramas na Figura 5.6 e na Figura 5.7 não representam a forma como, uma vez iniciada a conversa com o sistema, o usuário pode iniciar cada subconversa para atingir um objetivo específico, como o objetivo de efetuar a transferência. Em outras palavras, os diagramas naquelas figuras ainda não representam como o usuário leva a conversa para a cena Efetuar transferência. Para esse fim, utilizamos um **acesso ubíquo** (Figura 5.8), indicado por uma representação degenerada de cena, sem tópico e com cor de fundo cinza, conectada por uma fala do usuário à cena de destino para iniciar a conversa em direção ao objetivo representado pela fala. Um acesso ubíquo pode ser lido como “em qualquer cena onde o usuário esteja, ele pode enunciar <fala>”. Na Figura 5.8, a fala do acesso ubíquo é u: efetuar transferência. Uma vez iniciada essa conversa, o usuário pode mudar de ideia e desistir da transferência, retornando para a cena onde estava quando enunciou a fala do acesso ubíquo.

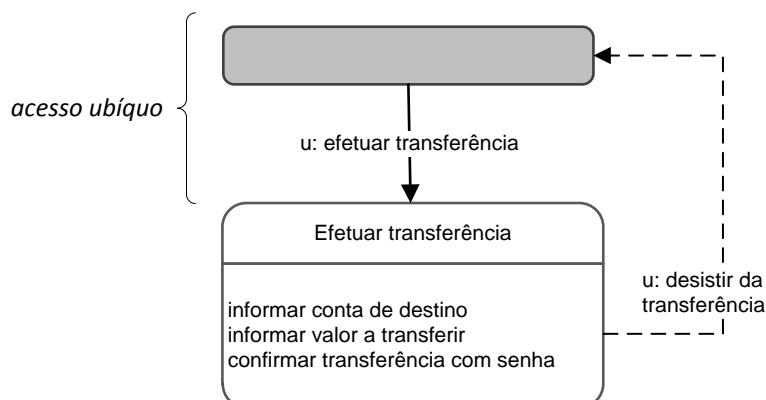


Figura 5.8. Acesso ubíquo para a cena Efetuar transferência.

Nem todo resultado de processamento do sistema requer uma fala explícita do preposto do designer. Em algumas situações, é possível que o preposto simplesmente conduza o usuário a uma cena, que por sua vez deixará claro o resultado do processamento. Nesse caso, é possível representar o resultado do processamento pela palavra-chave precond, em vez de uma fala explícita do preposto utilizando d:. Voltando ao exemplo de transferência bancária (Figura 5.6b), não é necessário informar ao usuário explicitamente que os dados da conta de destino estão corretos e que há saldo suficiente. A própria mudança de rumo para a cena Confirmar a transferência já reflete esse resultado

do processamento, ou seja, constitui a fala do preposto do designer. Nesse caso, basta indicar através da palavra-chave *precond* o que ocasionou aquela mudança de rumo da conversa (Figura 5.9). Cabe ao designer, portanto, decidir quais mudanças de rumo da conversa se beneficiam ou não de uma fala explícita do preposto do designer, que não seja apenas enunciar o tópico da próxima cena. Em linhas gerais, não se deve omitir a fala do preposto do usuário em casos de recuperação de ruptura. Observa-se na Figura 5.9 que a fala d: conta de destino inválida ou saldo insuficiente foi preservada.

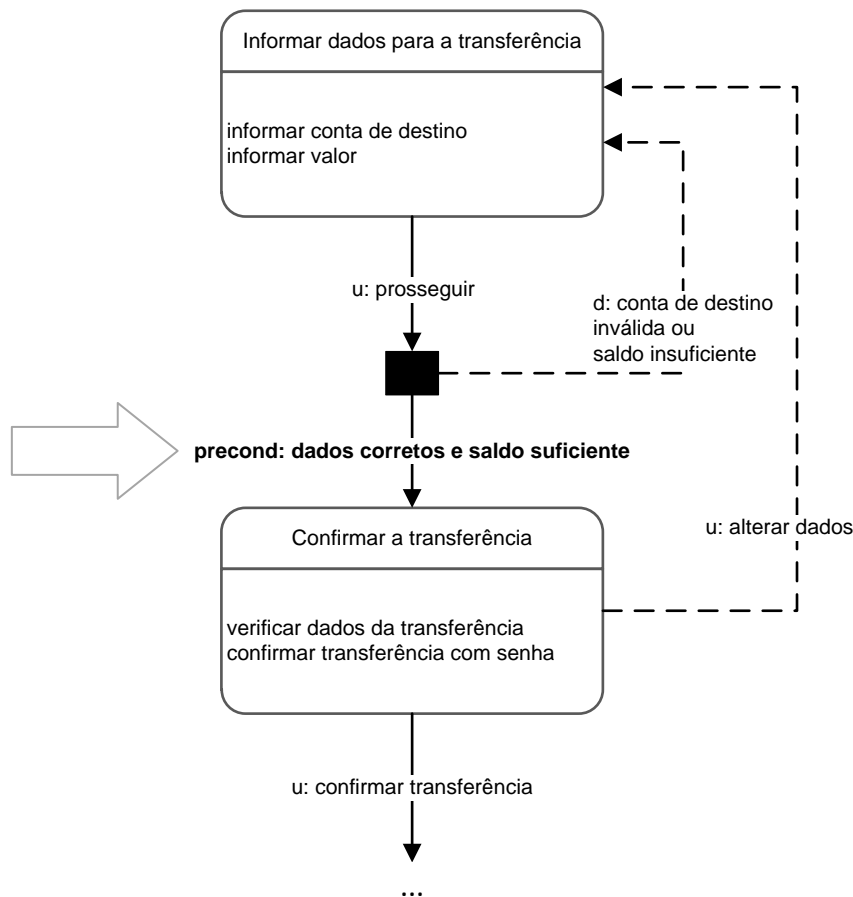


Figura 5.9. Uso da palavra-chave *precond* para representar um resultado de processamento sem fala explícita do preposto do designer.

De fato, a *precond* indica uma pré-condição para uma mudança de turno ser realizada, e pode ser utilizada tanto para mudanças de rumo iniciadas pelo designer quanto pelo usuário. Prosseguindo com o exemplo da transferência bancária, caso se queira impedir que o usuário enuncie a fala u: confirmar transferência até que a senha tenha sido informada, é possível incluir uma *precond* para esse fim, como ilustrado na Figura 5.10. A interface final deverá indicar essa pré-condição, possivelmente mantendo o botão de confirmação desativado até que a senha seja informada.

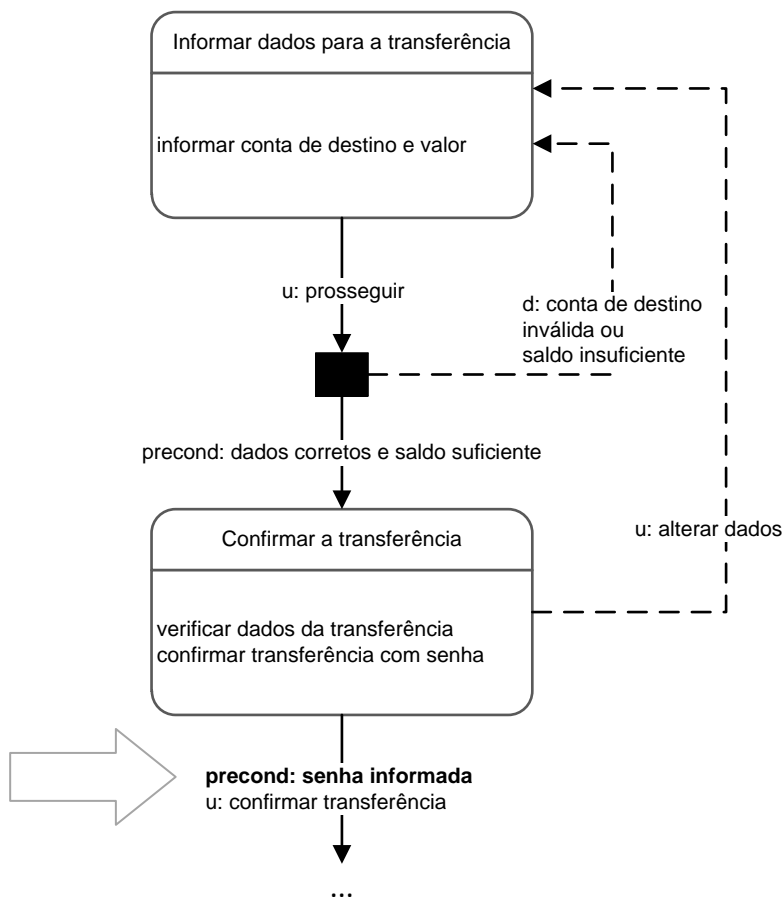


Figura 5.10 Uso da palavra-chave *precond* para impedir que o usuário enuncie uma fala de mudança de rumo da conversa.

Existem casos em que o designer considera certas conversas como podendo acarretar consequências negativas, como por exemplo a confirmação de uma transferência bancária que, por saldo insuficiente, resulte em o usuário fazer uso do seu limite de crédito. Nesses casos, a cena correspondente deve ser representada com linha tracejada, para indicar ao designer que os elementos de interface correspondentes à tal cena deverão ser devidamente diferenciados das demais. A Figura 5.11 ilustra essa situação.

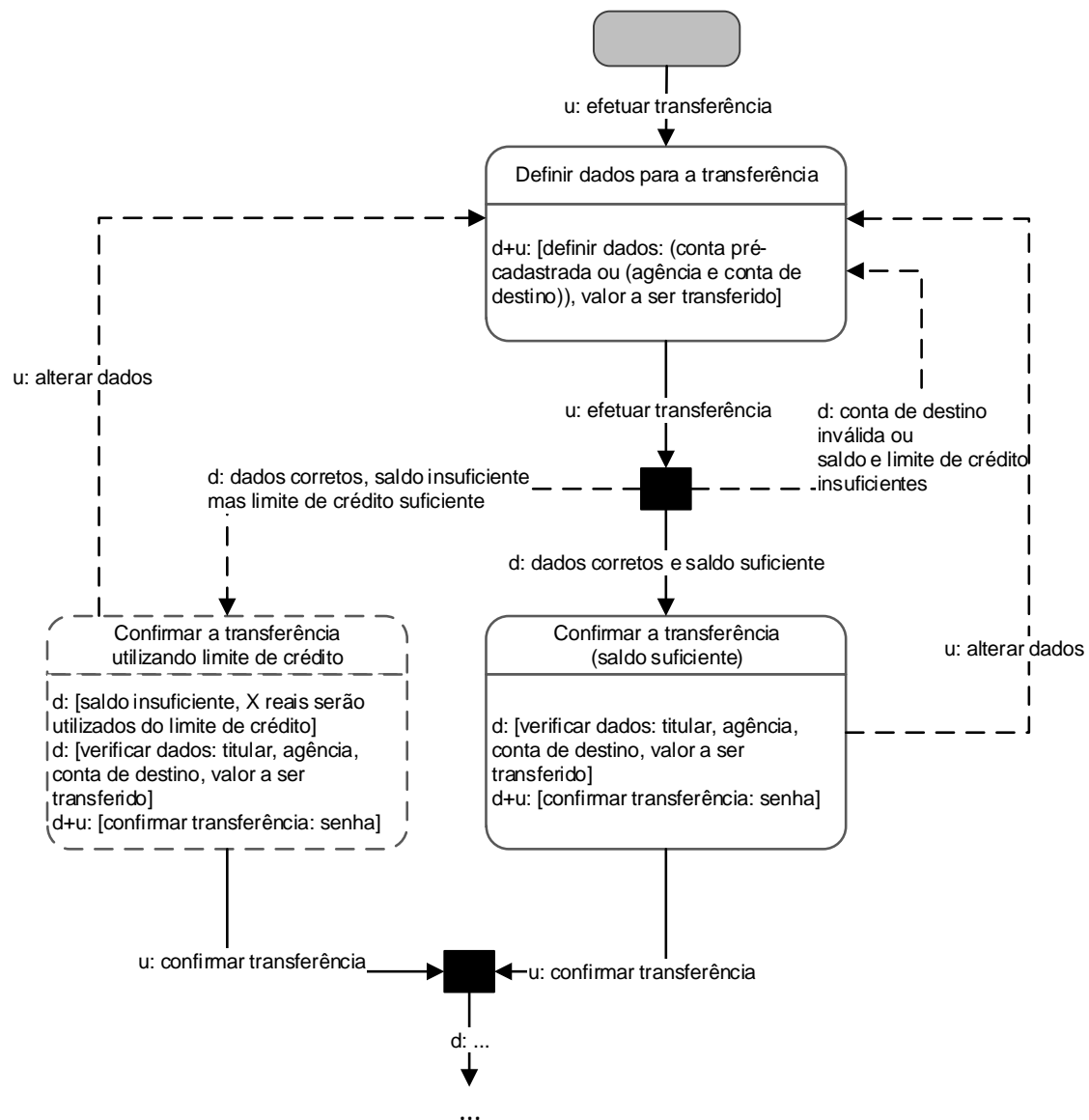


Figura 5.11. Cena de alerta indicando conversa que pode acarretar consequências negativas para o usuário (Confirmar a transferência utilizando limite de crédito).

Como dito anteriormente, o processamento do sistema é, do ponto de vista do usuário, uma caixa-preta. O usuário só sabe o que aconteceu através de falas do preposto do designer. Nas figuras apresentadas até o momento, as falas do preposto eram emitidas ao final do processamento. No entanto, há ocasiões em que se deseja fornecer informações sobre o estado do processamento enquanto ele ainda está em andamento. Um caso típico ocorre em operações de *download* de arquivos, como ilustra a Figura 5.12. Nesses casos, dizemos que há uma **comunicação síncrona** sobre o processamento do sistema.

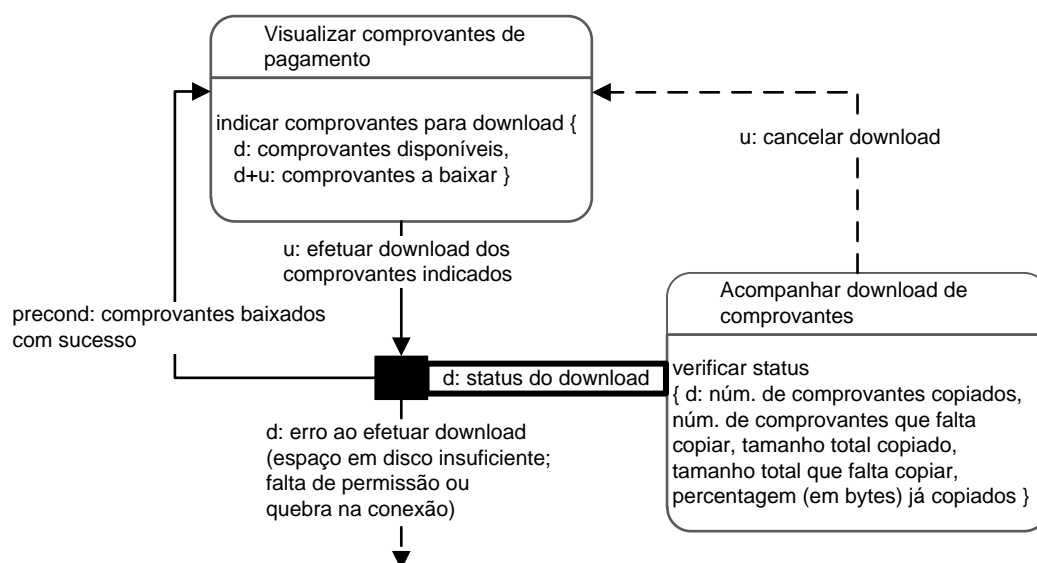


Figura 5.12. Comunicação síncrona sobre o processamento do sistema correspondente a uma operação de cópia.

Na Figura 5.12, observa-se a presença de um detalhamento sobre o enunciador das falas em cada diálogo, juntamente com os signos que compõem cada diálogo. A fala d: comprovantes disponíveis indica que o preposto do designer informa ao usuário quais são os comprovantes disponíveis, e d+u: comprovantes a baixar indica que usuário e preposto do designer dialogam sobre quais comprovantes devem ser baixados (ou seja, o usuário indica o que quer baixar e o preposto do designer indica que entendeu o pedido, possivelmente destacando os itens indicados pelo usuário). Como esse tipo de situação ocorre com frequência, costuma-se juntar as duas falas em uma única fala, como por exemplo: d+u: comprovantes disponíveis/a baixar.

Na cena Acompanhar download de comprovantes, há somente uma fala do preposto do designer, com diversos signos indicando o *status* do download. Nesse caso, trata-se rigorosamente de um monólogo, no qual o usuário deve interpretar a fala (composta de signos) emitida pelo designer, mas por sua vez não precisa responder diretamente com nenhuma fala. Nesse exemplo, a única fala que o usuário pode enunciar se refere a uma mudança de rumo na conversa, para indicar que não deseja prosseguir com o download (u: cancelar o download).

Em geral, a maioria dos diálogos é marcada pelo prefixo d+u:, indicando que se tratam de pares conversacionais, nos quais o preposto do designer fornece alguma instrução e solicita uma informação, fornecida então pelo usuário.

Além do detalhamento dos signos e seus enunciadores, os diálogos podem ser estruturados de diferentes formas, utilizando as palavras-chave SEQ (falas que precisam ser enunciadas em uma ordem predeterminada), OR (conjunto de falas no qual uma ou mais podem ser enunciadas) e XOR (falas mutuamente exclusivas, ou seja, conjunto de falas no qual somente uma fala pode ser enunciada).

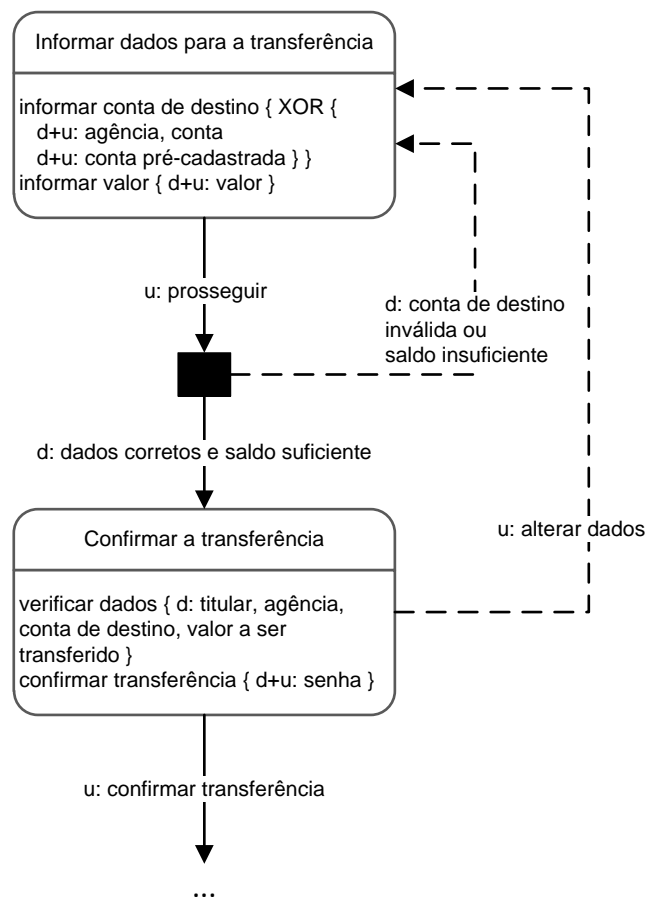


Figura 5.13 Exemplo de falas mutuamente exclusivas (XOR).

Até este momento, os diagramas representam o que poderá ocorrer no momento da interação, mas não incluem rastros de decisões do designer que levaram à solução neles representada. A MoLIC oferece a palavra-chave *presup* para indicar as pressuposições (em inglês, *presuppositions*) do designer sobre aspectos tenham levado a uma certa decisão de design, como por exemplo alguma característica do usuário, do seu objetivo, da sua atividade ou do contexto em que se espera que o sistema seja utilizado.

A *presup* permite aumentar o caráter epistêmico da MoLIC, pois ajuda ao designer registrar pontos de reflexão sobre o design e *tradeoffs* considerados. Por exemplo, a Figura 5.14 apresenta o uso de *presup* para indicar o motivo pelo qual o período *default* para a consulta de extrato abrange os últimos sete dias, com base em um conhecimento sobre o uso que os usuários fazem do sistema, provavelmente obtido através de instrumentação de software (*log* de uso do sistema).

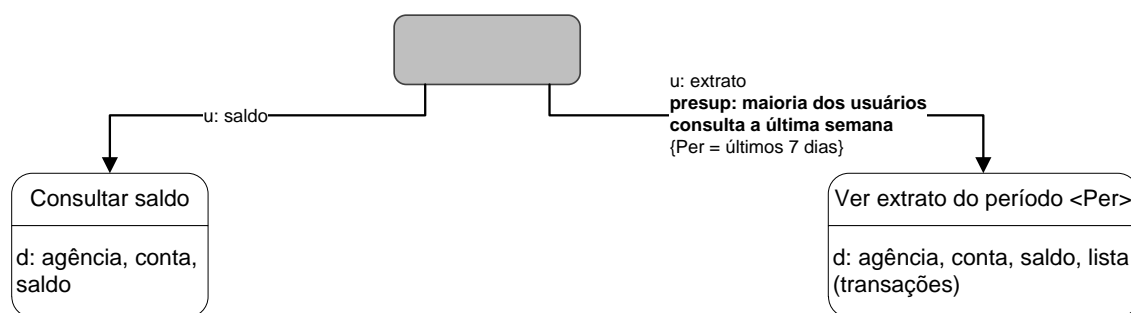


Figura 5.14 Exemplo de uso de *presup* para indicar o *rationale* de uma decisão de design.⁷

Existem principalmente duas situações para se utilizar *presup*. A primeira está relacionada com aspectos de natureza temporária, que podem mudar com o uso continuado do software ou com a evolução do seu contexto de uso. No caso da Figura 5.14, é possível que o acompanhamento dos *logs* de uso do sistema revele, no futuro, que a consulta mais frequente de extrato mudou dos últimos sete para os últimos três dias, para as últimas *n* transações, ou ainda que não há uma opção *default* clara, ou seja, que há uma variedade de opções utilizadas por diferentes grupos de usuário, e que portanto deveriam ser configuráveis.

A segunda situação em que *presup* é bastante útil ocorre quando o designer visa a comparar diferentes soluções de design, para registrar o que as diferencia. A Figura 5.15 apresenta uma solução alternativa para a consulta de extratos, que permite ao usuário atingir esse objetivo de duas maneiras: uma através de um acesso ubíquo, como na Figura 5.14, e outra a partir da cena Consultar saldo. Conforme registrado com a palavra-chave *presup*, essa segunda alternativa contempla o caso frequente em que, dada qualquer quebra de expectativa do usuário quanto ao saldo, ele formulará um novo objetivo de consultar o extrato, e o designer deseja criar esse atalho para evitar que o usuário tenha que buscar um acesso ubíquo (em meio a tantos outros acessos ubíquos). A seção 5.7 descreve como essas alternativas podem se concretizar na interface com o usuário.

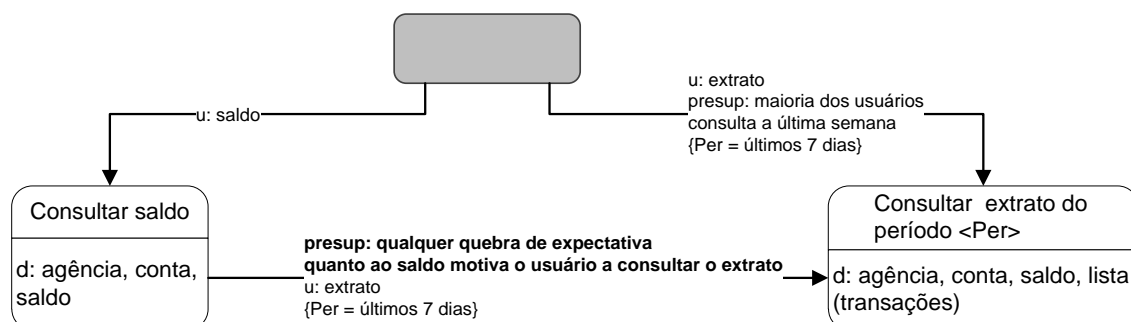


Figura 5.15 Exemplo de uso de *presup* para caracterizar uma solução alternativa.

Em conversas entre pessoas, tudo aquilo que foi dito em um momento pode ser recuperado em outro, ou seja, o histórico da conversa é mantido na memória dos

⁷ Observa-se na figura que os diálogos estão implícitos; somente os signos e seus enunciadores são representados. Essa representação pode ser adotada quando o tópico da cena e os signos forem suficientes para comunicar o teor da conversa naquele momento.

interlocutores. No caso de sistemas computacionais, caso alguma informação precise ser preservada para influenciar a conversa em momentos posteriores, é necessário registrar isso explicitamente. Para isso, a MoLIC oferece a palavra-chave *perloc*, que indica um efeito perlocutório de um trecho de conversa. A Figura 5.16 apresenta um exemplo de uso de *perloc* para indicar que, quando o usuário altera o período de consulta do extrato, o novo período deve ser “lembrado” em futuras consultas ao extrato durante a mesma sessão de uso, se sobrepondo ao período *default*. Para isso, faz-se uso de um signo dito articulado [Araujo 2008], como *sessão.período*. Tal recurso pode ser utilizado para efeitos de configuração com diferentes escopos conforme a intenção do designer, como por exemplo:

- **sessão:** *sessão.período* indica o período de extrato válido durante a sessão corrente, conforme o exemplo;
- **objeto** manipulado pelo sistema: *conta.período* indicaria o período de extrato *default* para a conta atual, independentemente de sessão;
- **usuário:** *usuário.período* indicaria o período de extrato *default* para o usuário corrente, para todas suas contas e em todas as sessões de uso do sistema).

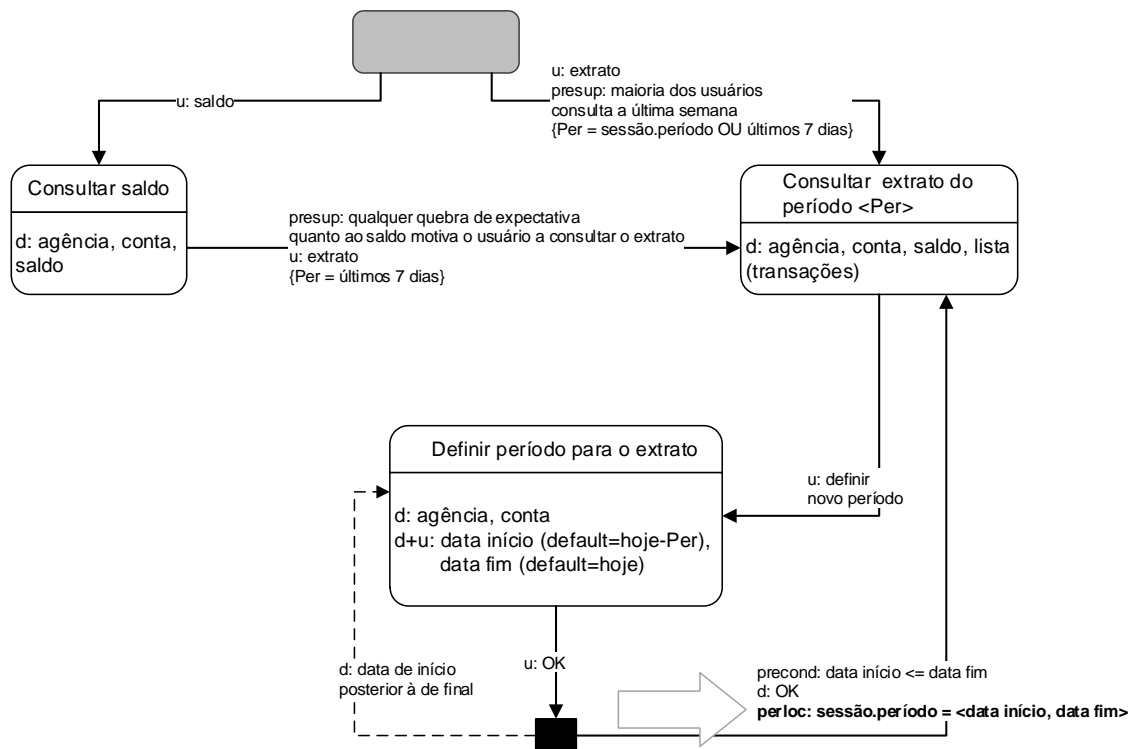


Figura 5.16 Exemplo de uso de *perloc* para “lembrar” parte de uma conversa travada anteriormente.

A próxima seção ilustra, passo a passo, um projeto de interação a partir dos objetivos dos usuários, discutindo algumas decisões de design que são tomadas ao longo do processo.

5.6 Projetando a interação

Não há uma ordem estrita para a construção de diagramas MoLIC. Entretanto, esta seção apresenta uma possível abordagem para essa construção. O design da interação com MoLIC costuma começar com a definição dos objetivos dos usuários que o sistema deve contemplar. No caso do sistema bancário, vamos considerar uma versão simplificada cujos objetivos são apenas os seguintes: consultar saldo, consultar extrato e efetuar transferência.

Em um primeiro momento, pode-se utilizar uma representação bem simplificada da MoLIC, ainda sem diálogos, e na qual cada objetivo é representado por uma cena (Figura 5.17).

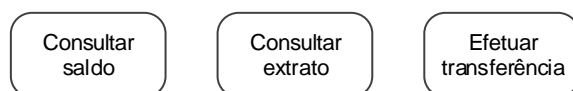


Figura 5.17 Início de construção de um diagrama MoLIC, contendo apenas cenas representando os objetivos dos usuários.

O segundo passo é identificar relações entre os objetivos dos usuários. Uma pergunta a fazer nesse momento é se a conclusão de um objetivo motiva o usuário a formular outro. Por exemplo, ao concluir uma transferência bancária, o usuário possivelmente gostaria de consultar o seu saldo. Para esses casos, inclui-se uma fala para a cena correspondente ao segundo objetivo (Figura 5.18).

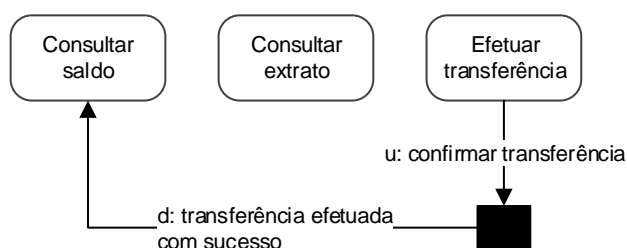


Figura 5.18 Diagrama MoLIC em construção, incluindo falas que levam de um objetivo a outro.

Em seguida, pode-se decidir para quais cenas haverá acessos ubíquos (Figura 5.19). Em geral, projeta-se acessos ubíquos para os principais objetivos do usuário. Nesse exemplo simplificado, são projetados acessos ubíquos para todos os objetivos.

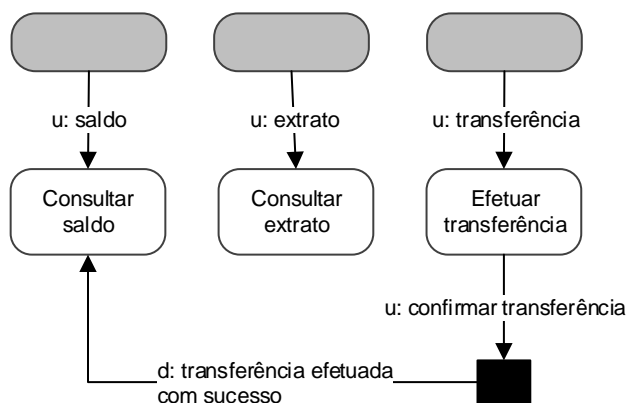


Figura 5.19 Acessos ubíquos para os objetivos dos usuários.

O início da conversa com o sistema é então projetado através de pontos de abertura, indicando se o usuário, ao entrar no sistema, já começa a conversa em direção a um objetivo, ou se é levado a uma cena de início como uma *home* ou *dashboard*, a partir da qual deve decidir qual objetivo deseja alcançar. No caso de uma aplicação bancária, o usuário precisa inicialmente identificar a conta que deseja manipular, e caso o designer assuma que o usuário deseja primeiramente consultar seu saldo, pode projetar uma solução semelhante à da Figura 5.20. Dependendo da plataforma computacional em que o sistema deva ser projetado, no entanto, é possível que questões de segurança inviabilizem essa solução.

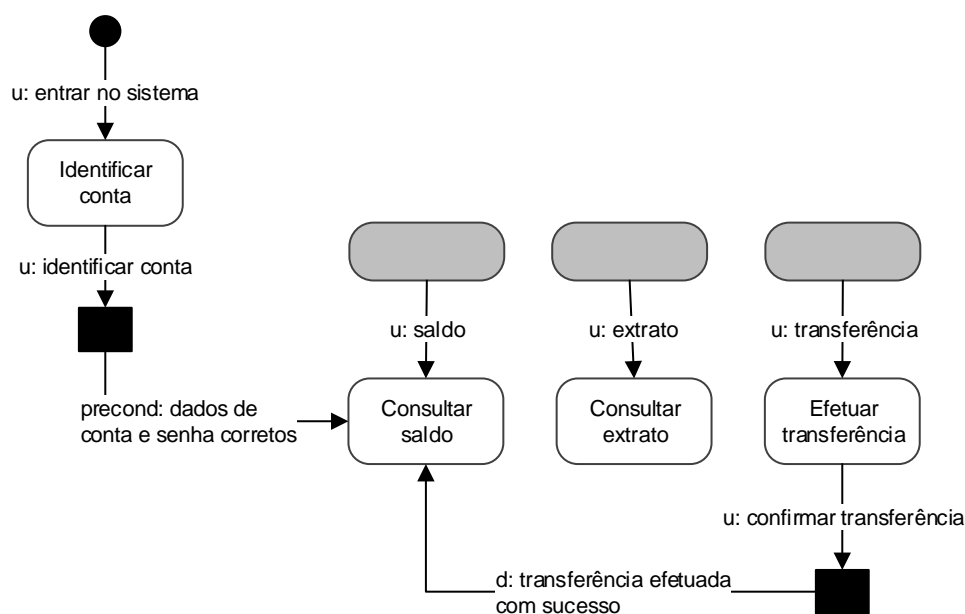


Figura 5.20 Ponto de abertura do sistema.

As cenas podem então ser decompostas e interligadas pelas falas de troca de turno e processamento do sistema (Figura 5.21).

Finalmente, são detalhados os diálogos e signos, tal como exemplificado dos diagramas da Figura 5.12 à Figura 5.16.

No caso do sistema bancário de exemplo, além da questão de iniciar com a consulta ao saldo ou não, existem outras decisões de design que afetam o projeto de interação, como por exemplo as seguintes:

- Ao solicitar o extrato, deve-se assumir um período de tempo fixo, assumir um número fixo de últimos lançamentos ou requerer que o usuário sempre informe o período de tempo desejado?
- Ao concluir uma transferência, deve-se apresentar uma mensagem informando sobre o resultado da operação, levar o usuário à consulta ao saldo, ou à consulta ao extrato?

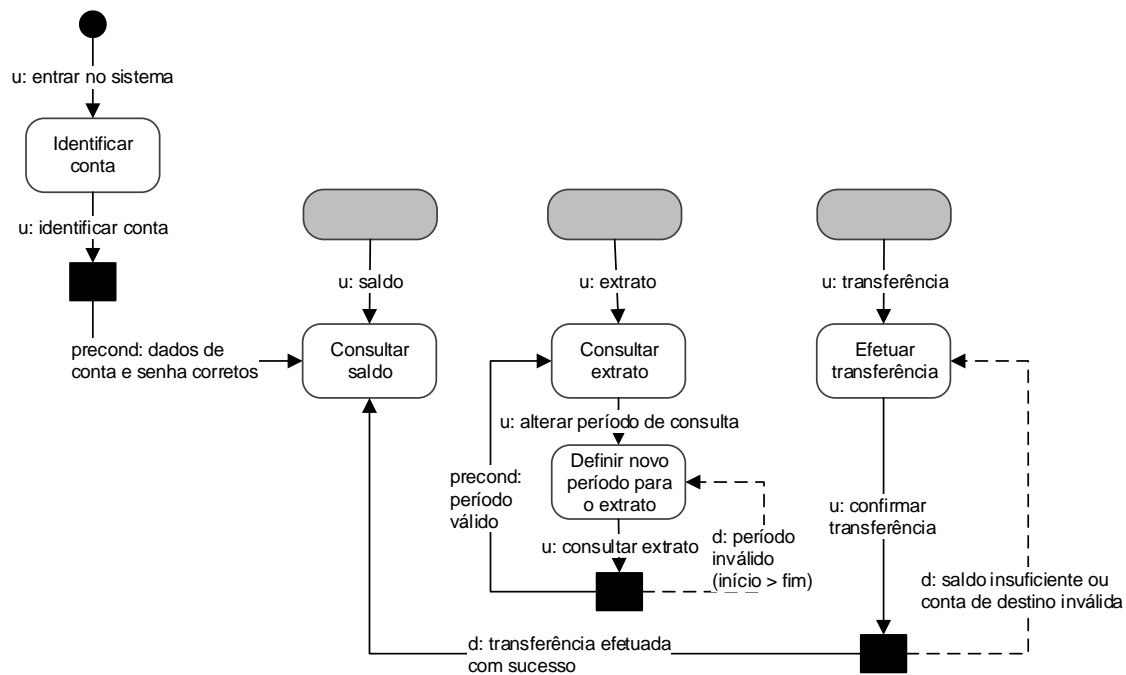


Figura 5.21 Exemplo de diagrama MoLIC de aplicação bancária, ainda utilizando uma notação simplificada (sem diálogos e signos).

Como visto na seção anterior, essas e outras decisões podem ser registradas em diagramas alternativos, utilizando o elemento *presup* para indicar em que situações o designer acredita que cada solução de design de interação seria mais recomendada, ou seja, quando cada estratégia de uso traria mais benefícios ao usuário.

5.7 Mapeamentos entre o Design da Interação e o Design da Interface

Cada diagrama de interação MoLIC pode ser concretizado por diferentes interfaces. No entanto, há alguns padrões de mapeamento que podem guiar as decisões de design envolvidas no projeto da interface a partir do projeto da interação. Vale observar que não se propõe que o processo seja estritamente sequencial, ou seja, que se tenha especificado completamente a interação em MoLIC antes de se definir a interface propriamente dita.

À medida que a interação é projetada, cabe aos designers transitarem pelo design da interação e da interface alternadamente, para melhor vislumbrar as consequências de suas decisões de design considerando as duas perspectivas (interação e interface). Alguns dos mapeamentos comumente utilizados em interfaces gráficas e Web são apresentados na Tabela 5.1.

Tabela 5.1 Mapeamento usual entre elementos da MoLIC e elementos de interface.⁸

elemento da MoLIC	elemento de interface	exemplo
cena	▪ janela ou página	Figura 5.22a
tópico da cena	▪ título da janela ou página	Figura 5.22b
diálogos da cena	▪ regiões da tela ou estruturas de painéis e agrupamentos de outros elementos de interface	Figura 5.22c
signos	▪ elementos de interface interativos (no caso de d+u:) ▪ elementos de interface não interativos (no caso de d:)	rótulos e campos de agência, conta e senha na Figura 5.22
acesso ubíquo	▪ item de menu ou barra de navegação	
fala do usuário (seja para troca de turno ou para recuperação de ruptura)	▪ botão (principalmente se envolve um processamento do sistema (ou um cancelamento de um processamento) ou ▪ link (principalmente se leva diretamente a uma cena, como uma navegação simples, que não envolva processamento)	botão OK (Figura 5.22d)
fala do preposto do designer para troca de turno ou mudança de rumo da conversa	▪ janela de mensagem (principalmente no caso de mudança de assunto) e/ou ▪ informações na janela (cena) de destino (principalmente no caso de simples troca de turno, sem mudança de assunto)	
fala do preposto para recuperação de ruptura	▪ janela de mensagem de erro (principalmente no caso de erros sem recuperação) e/ou ▪ informações na cena de destino (principalmente com indicações de rupturas, como por exemplo instruções para o preenchimento correto de um formulário e de cada campo (signo) que tiver causado a ruptura)	mensagem no topo da Figura 5.23a e destaque nos campos com problema (Figura 5.23b)

A Figura 5.22 apresenta um exemplo de mapeamento da cena Identificar conta e seus elementos.

⁸ Os mapeamentos aqui apresentados não devem ser considerados prescritivos, e sim como tendo surgido espontaneamente em diagramas MoLIC e esboços de tela construídos por diferentes alunos e profissionais [Silva et al. 2005].

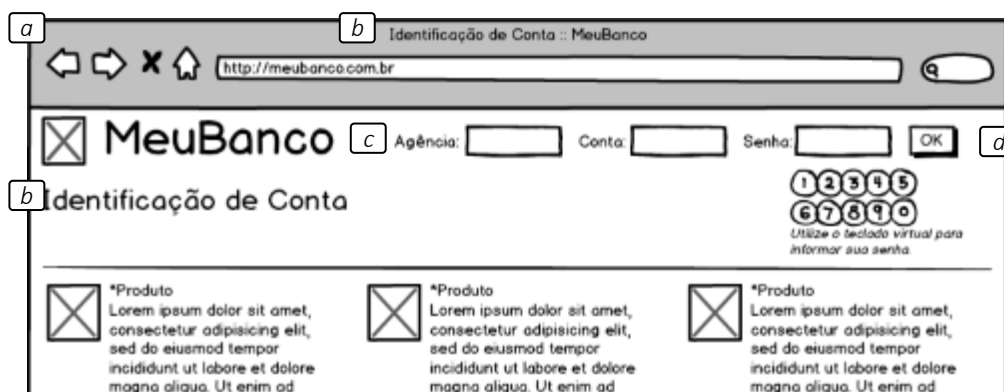


Figura 5.22 Exemplo de página mapeada da cena Identificar conta.

A Figura 5.23 apresenta o mesmo exemplo da Figura 5.22, mas após a ocorrência de uma ruptura causada pelo fornecimento de dados incorretos. A fala do designer para a recuperação da ruptura é mapeada para dois elementos:

- mensagem de erro no topo da página (Figura 5.23a);
- indicações dos signos que causaram a ruptura (Figura 5.23b).

Este exemplo ilustra como uma preocupação com segurança se sobrepõe a preocupações com a qualidade da comunicação com o usuário. Do ponto de vista estritamente comunicativo, seria bom apresentar ao usuário uma mensagem mais precisa, indicando, se possível, o signo que originou a ruptura, como por exemplo “Agência inexistente”, “Conta não pertence à agência informada” ou “A senha não corresponde à conta informada.” No entanto, tais mensagens constituem em aumento de risco de identificação de contas bancárias de terceiros, o que deve ser evitado ao máximo.

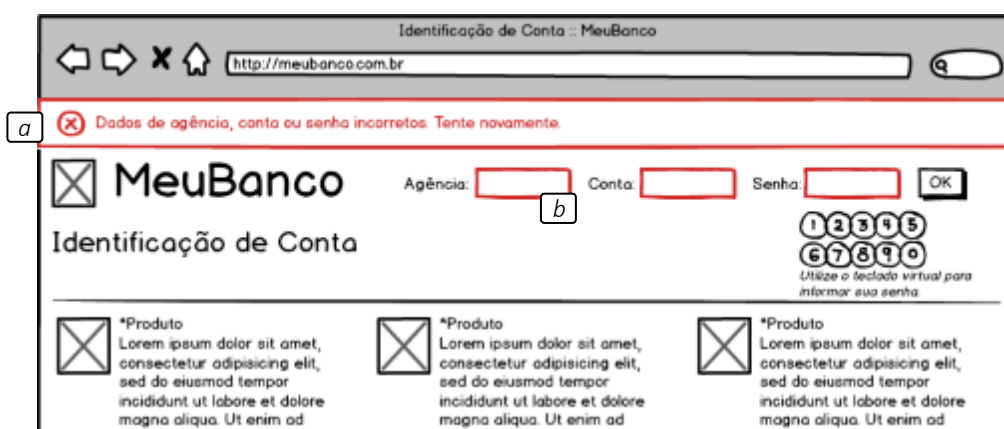


Figura 5.23 Exemplo de página mapeada da cena Identificar conta, com indicações para correção de ruptura.

Como visto na Figura 5.15, às vezes o designer decide oferecer atalhos para o usuário iniciar conversas a partir de uma cena, além do recurso tradicional de acesso ubíquo. A Figura 5.24 ilustra uma página de consulta de saldo com os dois acessos. Observa-se que esse tipo de atalho é mais útil quando há muitos acessos ubíquos, alguns dos quais

ocultos por uma hierarquia de menus, ou quando há uma sequência de operações claras em um fluxo de trabalho.

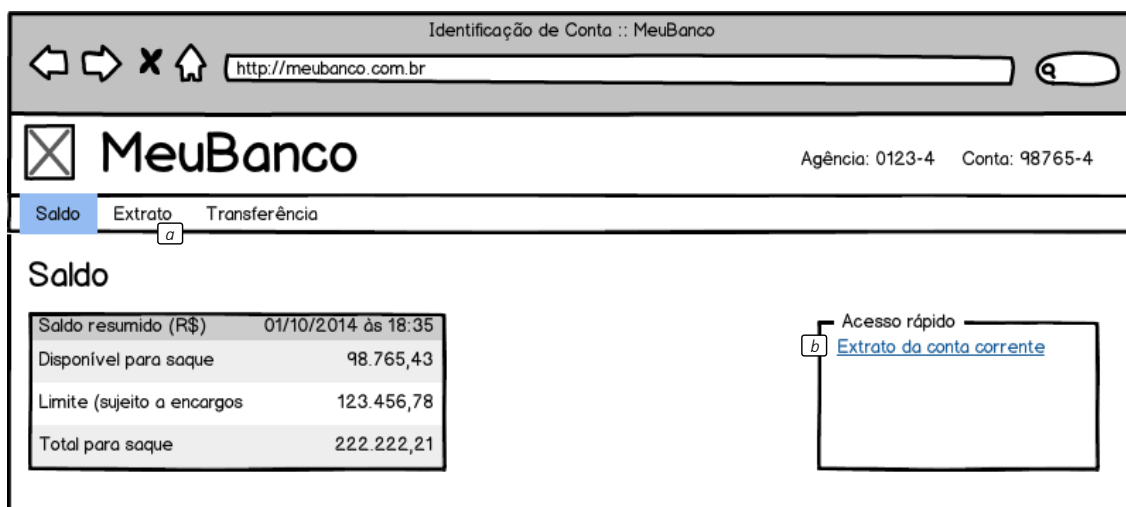


Figura 5.24 Exemplo de página com dois acessos para uma mesma cena (Consultar extrato): (a) acesso ubíquo e (b) fala do usuário para mudança de rumo da conversa.

Alguns elementos do diagrama MoLIC não possuem representação explícita na interface. Processamentos de sistema, por exemplo, não são representados na interface, a menos que acompanhados de uma cena síncrona informando o andamento do processamento. Além disso, quando uma fala do designer para troca de turno não tem conteúdo (apenas precondition) ou tem como conteúdo algo como d: OK, a própria continuidade da conversa (levando o usuário à cena desejada) pode ser considerada suficiente para indicar o sucesso da solicitação do usuário. Nesse caso, o designer pode julgar desnecessário criar uma representação explícita dessa fala na interface. A definição de tais elementos na MoLIC, no entanto, é imprescindível para se entender e registrar como ocorrerá a interação dos usuários com o sistema, e quando a comunicação explícita é ou não necessária.

5.8 Uso da MoLIC no apoio à avaliação da interação

Araujo (2008) propôs uma série de perguntas para apoiar a leitura e interpretação de diagramas MoLIC, visando a evidenciar o caráter epistêmico da linguagem. Embora o objetivo dessas perguntas não seja fornecer um julgamento de valor à interação representada em MoLIC, ao buscar responder essas perguntas o designer reflete sobre as consequências das decisões de design ali representadas. Dessa forma as perguntas servem como base indireta para uma avaliação formativa da interação.

Segundo Araujo (2008), seu trabalho permitiu suscitar pontos que foram classificados em dez categorias de reflexão:

- Comparação com sistemas similares;
- Elaboração de cenários de uso;
- Levantamento de discussão com terceiros;
- Confirmação ou reformulação de resposta dada em perguntas anteriores;

- Demonstração de dificuldade ou incômodo na utilização de elementos do diagrama de interação;
- Leitura ou releitura do diagrama de interação;
- Possíveis alternativas de modelagem;
- Recordação de decisões de design tomadas na época da modelagem;
- Confirmação de decisões de design tomadas na época da modelagem; e
- Revogação de decisões de design tomadas na época da modelagem.

A seguir apresentamos uma adaptação de algumas das perguntas de Araujo. Para entender quais reflexões se busca suscitar com cada questão, recomenda-se consultar diretamente a dissertação de Araujo (2008).

Sobre os objetivos dos usuários

- Quais são os objetivos de cada papel de usuário? Qual é a interseção ou relação entre eles? De que maneiras esses objetivos podem ser agrupados ou estruturados?

Sobre a abertura e encerramento da conversa com o sistema

- Quais são os possíveis inícios de conversa com o sistema? Qual é o primeiro objetivo que se quer alcançar assim que a conversa se inicia?
- No caso do usuário se identificar no sistema, que informações o sistema tem sobre ele ou sobre o que ele já fez em interações anteriores? De que forma o sistema utiliza essas informações? Como elas influenciam a interação?
- Quando é que o usuário pode sair do sistema, e o que ocorre nesse momento? Antes de encerrar a interação, é necessário realizar alguma operação? Como isso é feito? Que implicações isso traz para a próxima sessão de uso do sistema por esse mesmo usuário?

Início de conversa voltada para um objetivo

- Onde e quando este papel de usuário pode iniciar a conversa sobre este objetivo? Onde e quando o usuário pode declarar sua intenção de realizar este objetivo?
- Quais são as possíveis conversas para este papel de usuário atingir este objetivo? De que maneira essas conversas refletem diferentes estratégias, características do usuário ou objetivos de design? Existe uma conversa preferencial do designer para que o usuário atinja este objetivo? Por quê?
- Como este papel de usuário pode desistir de atingir este objetivo? O que acontece quando ele desiste de atingi-lo?
- Que outras conversas (que não sobre os objetivos) podem ser iniciadas a qualquer momento? Com que frequência são iniciadas, ou o quanto são importantes?

Cenas

- Quais são as possíveis formas deste papel de usuário chegar a esta cena? Por que todas elas são necessárias?
- De que forma o preposto comunica o que este papel de usuário pode/deve fazer nesta cena?
- De que maneira os diálogos e signos estão relacionados ao tópico e às falas do usuário que saem desta cena?
- De que maneira os grupos de diálogos refletem uma estrutura de subtópicos nesta cena?
- Quem pode travar a conversa nesta cena, e quando?

Diálogos e Signos

- Quem está emitindo cada signo? Que valores este signo pode assumir?
- O que o preposto tem a dizer sobre o conteúdo predefinido de cada signo d+u?
- O usuário deve necessariamente falar sobre este signo? Por quê?
- De que forma o preposto orienta o usuário a respeito do que este pode/deve dizer sobre este signo para evitar rupturas na comunicação? De que forma as rupturas estão sendo prevenidas?
- Caso o usuário se engane, como o preposto lhe ajuda a corrigir a conversa sobre este signo?

Falas do usuário e do designer para troca de turno e para mudança nos rumos da conversa

- Após travar este trecho de conversa, o que o usuário pode fazer?
- Como o preposto comunica cada efeito perlocutório da fala do usuário (resultante do processamento do sistema) e devolve o turno da conversa para o usuário?
- Com base na conversa travada nesta cena, que informações seriam interessantes de serem mantidas no contexto, de forma a afetar a continuação da conversa em outras cenas? A que signos articulados essas informações estão relacionadas?

5.9 Conclusões

Este capítulo apresentou a MoLIC, linguagem de modelagem da interação como uma conversa, fundamentada na Engenharia Semiótica. Descrevemos como a MoLIC enfoca a comunicação, para o usuário, das situações de uso previstas pelo designer, enfatizando o papel do designer como interlocutor durante a interação, através de seu preposto (a interface de usuário). Utilizando a MoLIC, o designer representa e reflete sobre as possibilidades de interação que estarão disponíveis para os usuários alcançarem seus diversos objetivos através da interação com a solução computacional sendo projetada.

A MoLIC foi proposta inicialmente para representar a interação de um único usuário com o sistema. Embora tenham sido propostos pontos de contato entre diagramas MoLIC elaborados para diferentes papéis de usuários, tal solução está muito

aquém do que é necessário para apoiar o projeto da interação em sistemas colaborativos. Está em andamento uma pesquisa para estender a linguagem para esse fim.

Outra questão de pesquisa em aberto diz respeito ao projeto de sistemas interativos adaptáveis e adaptativos, que atualmente não possuem apoio da linguagem senão implicitamente, através de signos articulados indicando o escopo de uma configuração (se vale por toda a sessão, para aquele usuário ou para o objeto sendo manipulado, como um documento, por exemplo), assim como no exemplo de período de consulta do extrato (sessão.período).

Referências

- Annett, J. “Hierarchical Task Analysis”. In: Diaper, D. e Stanton, N. (eds.), *The handbook of task analysis for human-computer interaction*. Mahwah, NJ: Lawrence Erlbaum Associates, pp. 67 –82, 2003.
- Annett, J. e Duncan, K.D. Task analysis and training design. *Occupational Psychology*, 41, pp. 211–221, 1967.
- Araujo, A.C.I.C. (2008) *Apoio ao Design e à Interpretação de Modelos de Interação Humano-Computador Representados em MoLIC*. Dissertação de Mestrado. Departamento de Informática, PUC-Rio.
- Barbosa, S.D.J.; Silva, B.S. (2010) *Interação Humano-Computador*. Campus/Elsevier. ISBN: 978-85-352-3418-3.
- Card, S.; Moran, T.P.; Newell, A. (1983) *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates. ISBN: 978-0898598599.
- Carroll, J.M. (ed., 2003) *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. Morgan Kaufmann. ISBN: 978-1558608085.
- Cooper, A.; Reimann, R.; Cronin, D. (2007) *About Face 3: The Essentials of Interaction Design*. New York, NY: John Wiley & Sons. ISBN: 978-0470084113.
- Courage, C. & Baxter, K. (2005) *Understanding your users: a practical guide to user requirements, methods, tools, and techniques*. San Francisco, CA: Morgan Kaufmann Publishers. ISBN: 978-1558609358.
- de Souza, C.S. (2005) *The Semiotic Engineering of Human-Computer Interaction*. The MIT Press. ISBN: 978-0262042208.
- Diaper, D.; Stanton, N. (eds., 2003) *The Handbook of Task Analysis for Human-Computer Interaction*. CRC Press. ISBN: 978-0805844320.
- Kieras, D. GOMS Models for Task Analysis. In: D. Diaper & N. Stanton (eds.), *The handbook of task analysis for human-computer interaction*. Mahwah, NJ: Lawrence Erlbaum Associates, pp. 83 –116, 2003.
- Lawson, B. (2006) *How Designers Think: The Design Process Demystified, 4th edition*. Oxford, UK: Architectural Press. ISBN: 978-0750660778.
- Paternò, F. *Model-Based Design and Evaluation of Interactive Applications*. London, UK: Springer-Verlag, 1999.

- Paula, M. G. (2003) *Projeto da Interação Humano-Computador Baseado em Modelos Fundamentados na Engenharia Semiótica: Construção de um Modelo de Interação*. Dissertação de Mestrado. Departamento de Informática, PUC-Rio.
- Rogers, Y.; Sharp, H.; Preece, J. (2011) *Interaction Design: Beyond Human-Computer Interaction, 3rd edition*. John Wiley & Sons. ISBN: 978-0470665763.
- Rosson, M.B. & Carroll, J.M. (2001) *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. San Francisco, CA: Morgan Kaufmann Publishers. ISBN: 978-1558607125.
- Sangiorgi, U.B. (2010) *Apoiando o Projeto e a Avaliação da Interação e da Interface: um Estudo e uma Ferramenta*. Dissertação de Mestrado. Departamento de Informática, PUC-Rio.
- Silva, B.S. (2005) *MoLIC 2a Edição: Revisão de uma Linguagem para Modelagem da Interação Humano-Computador*. Dissertação de Mestrado. Departamento de Informática, PUC-Rio.
- Silva, B.S.; Martins Netto, O. A; Barbosa, S.D.J. (2005) Promoting a Separation of Concerns via Closely-Related Interaction and Presentation Models. *Proceedings of Latin-American Conference on Human-Computer Interaction, CLIHC*, 2005.