

4 Modelo de Interação Proposto

O modelo de tarefas descreve a estrutura das tarefas que serão apoiadas pelo sistema. No entanto, não é suficiente para dar forma à solução computacional tal como esta será percebida pelo usuário. Na modelagem das tarefas, cada meta e suas tarefas são modeladas separadamente, não incluindo a “relação” entre as metas. Por exemplo, na modelagem de tarefas do sistema de Quadro de Avisos, existem as metas *Consultar avisos* e *Remover avisos*. Como foi visto na seção 3.3, a modelagem de estas metas e suas tarefas é feita separadamente, ou seja, em diagramas distintos. Mas é importante, em algum momento do processo de *design*, identificar a necessidade de estas metas estarem disponíveis em um mesmo momento de interação. A “proximidade” entre estas metas facilita o uso do sistema, pois para um usuário que estiver buscando um aviso para ser removido, é mais prático que ele possa remover o aviso procurado assim que ele seja encontrado. Isto evita de ele ter que sair da busca e ir a outro local na interface, para remover o mesmo aviso encontrado. Portanto, antes de se especificar a interface propriamente dita, a partir do modelo de tarefas deve-se projetar a interação de forma cuidadosa.

Como dito anteriormente, a fundamentação teórica deste trabalho é a Engenharia Semiótica. Esta teoria vê a interação como uma conversa entre o usuário e um representante do *designer* que está cristalizado na interface (preposto do *designer*). Como a interação é vista como uma conversa, as ferramentas de *design* de IHC devem apoiar o *designer* no projeto de todas as conversas que podem acontecer quando o usuário utiliza a aplicação. As ferramentas devem motivar o *designer* a expressar o seu processo de tomada de decisão e o resultado deste processo através do seu preposto, maximizando as chances de os usuários entenderem melhor o artefato projetado, evitando rupturas na comunicação. Parte deste processo de tomada de decisão pode estar expresso, por exemplo, nos signos de interface e parte pode estar explicitamente comunicado através do *Help*. Além

disto, as ferramentas também devem facilitar a reflexão do *designer* sobre suas tomadas de decisão na concepção do artefato, tentando minimizar as chances dele inserir inconsistências em seu *design*.

Neste trabalho, explorou-se a metáfora de interação como conversação (Brennan, 1990) para desenvolver um modelo que representa a interação como um conjunto de conversas que os usuários podem travar com o preposto do *designer*, sem especificar ainda os *storyboards* ou a interface propriamente dita. Nesta conversa o preposto vai “falar” alguma coisa, o usuário vai “falar” algo, e o preposto vai “responder” (reagir) ao que o usuário falou. Esta troca de turnos entre preposto e usuário compõe o ciclo mínimo de interação (da Silva, 2001), que é composto por três passos: o software diz alguma coisa para o usuário, o usuário diz alguma coisa para o software e o software responde ao usuário.

A notação do modelo de interação proposto é descrita na seção 4.1. A seção 4.2 apresentada brevemente o uso do modelo na avaliação de aplicações existentes.

4.1 Notação

Os elementos básicos do diagrama de interação são cenas, processos do sistema e transições. Uma cena representa uma conversa sobre um determinado assunto, ou tópico. Esta conversa é composta de um ou mais diálogos, que por sua vez são compostos de falas do usuário e do preposto, organizadas como pares conversacionais¹². A grosso modo, pode ser vista como o palco onde a conversa usuário–preposto ocorre, e onde geralmente cabe ao usuário “dar a última palavra”, que poderá causar uma mudança de rumo na conversa. Uma cena poderá ser mapeada, no modelo de interface, por exemplo, em um componente estruturado tal como uma tela, janela ou página.

No diagrama de interação, uma cena é representada por um retângulo com bordas arredondadas e uma identificação do seu tópico de conversa, expresso preferencialmente como verbo no modo infinitivo e do ponto de vista do usuário.

¹² Para uma definição de par conversacional, veja (Marcuschi, 2001).

O(s) diálogo(s) que a compõe(m) pode(m) ser identificado(s) na parte inferior da cena, entre colchetes, e separados do tópico da cena por uma linha horizontal (Figura 4.1). Os diálogos indicam o que o usuário e o preposto podem dizer sobre o tópico da conversa.

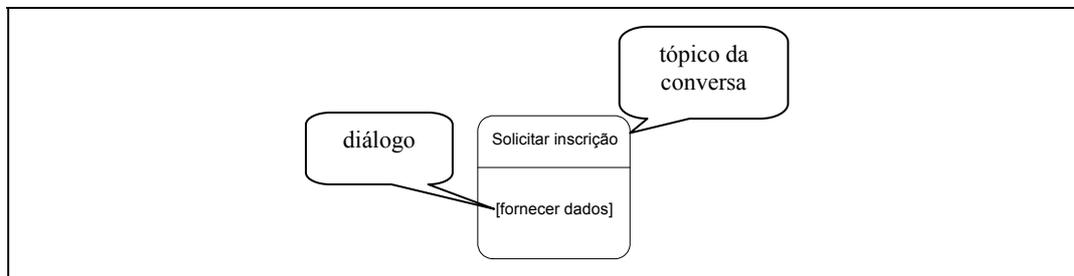


Figura 4.1: Representação de uma cena.

Uma cena especial é a que contém apenas uma fala do preposto. Isto é, uma fala resultante de um processamento, que foi emitida em uma cena em resposta a uma fala do usuário (que causou o processamento). Por exemplo, a cena que contém somente uma fala do preposto cujo o conteúdo deve ser equivalente a: <<Aguarde email do administrador.>>, em resposta a um pedido de inscrição do usuário (Figura 4.2).

Os processos do sistema emitem as falas do preposto do *designer* que afetam o rumo que a conversa pode tomar, como resposta a uma ou mais falas do usuário. Em outras palavras, representa-se um processo quando houver mais de um caminho de interação possível a partir de uma cena, e a decisão sobre que caminho tomar depende de um processamento interno do sistema, revelado apenas pela fala do preposto do *designer* que dele resulta. Portanto, os processos só precisam ser representados quando seu resultado precisa ser comunicado ao usuário. Na prática, isto ocorre sempre que há mais de uma transição possível entre a cena corrente e a(s) próxima(s), e a escolha dentre estas transições depende de um processamento, e não da decisão direta do usuário. Um processo é representado por uma caixa preta (Figura 4.2). Usou-se a noção de “caixa preta” pois o usuário não percebe diretamente o que está ocorrendo em um processamento do sistema, só percebe através de falas do preposto. Esta representação visa motivar o *designer* a projetar cuidadosamente as falas do preposto a serem emitidas após cada processamento, como única forma de comunicar ao usuário o que/como ocorreu, e explicar por quê.

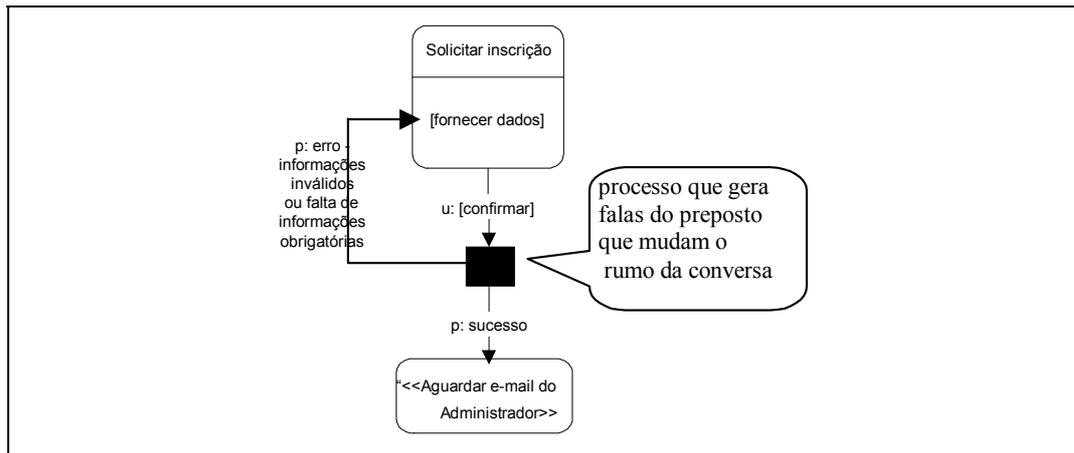


Figura 4.2: Representação de um processo do sistema.

Uma transição representa uma mudança de rumo na conversa, seja causada por uma fala do usuário (a partir de uma cena), ou por uma fala do preposto (geralmente a partir de um processamento). Uma transição é expressa por uma seta preta indicando a direção da transição e um rótulo. Opcionalmente, quando há a possibilidade de se voltar à cena de origem, a transição pode ter uma seta branca indicando esta possibilidade. Isto é, a transição terá em uma extremidade a seta preta indicando a direção da transição e na outra, a seta branca, indicando a possibilidade de se voltar à cena de origem. O rótulo de uma transição é composto de até três partes (Figura 4.3):

1. pré-condições: condições que devem ser satisfeitas para que o usuário ou preposto possa enunciar a fala correspondente. São precedidas pela expressão *pré:*. Podem ser expressas em linguagem natural, através de expressões lógicas ou alguma outra notação de preferência do *designer*. Por exemplo, *pré: o usuário deve ter permissão para alterar o aviso*.
2. uma ou mais falas do usuário ou do preposto do *designer*. No caso do usuário, trata-se do(s) enunciado(s) do usuário que causa(m) a transição. As falas do usuário são expressas no formato *u:[fala do usuário]*. Por exemplo, *u:[confirmar alteração]*. No caso do preposto, trata-se da(s) fala(s) que revela(m) o(s) resultado(s) de algum processamento do sistema que causa(m) a transição. As falas do preposto são expressas no formato *p:fala do preposto*. Por exemplo, *p: erro - login inexistente*.

3. pós-condições: condições que passam a ser verdadeiras durante a transição. São precedidas pela expressão *pós:*. Por exemplo, *pós: login efetuado*.

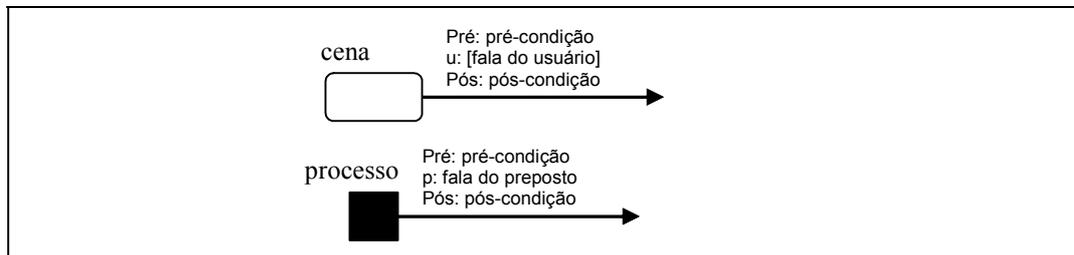


Figura 4.3: Representação de uma transição.

Algumas cenas podem ser acessadas (a conversa descrita na cena poderá ocorrer) em qualquer ponto da aplicação, ou seja, a partir de qualquer outra cena. O acesso a estas cenas, chamado ubíquo, é marcado por uma transição que sai de uma cena preenchida pela cor cinza (por exemplo, o acesso a cena *Efetuar login*, na Figura 4.4). Se houver pré-condições para se acessar a cena, deve-se indicá-las na transição que dá acesso à cena (por exemplo, a *pré: _login=F* na transição que dá acesso à cena *Efetuar login* na Figura 4.4).

Com frequência, as possibilidades de interação, bem como o comportamento de cada cena, variam conforme o estado atual da aplicação. Quando isto ocorre, podemos definir signos de controle da interação cujos valores compõem este estado. No exemplo da Figura 4.4, para poder acessar a cena *Efetuar login*, o usuário não pode estar “logado” no sistema (*_login=F*). Ao efetuar o login com sucesso, este signo de controle recebe o valor *true* (*_login ← T*), indicando que houve uma mudança de estado da aplicação. Estes signos de controle podem ser acrescentados à tabela de signos descrita na seção 3.2, com o tipo *controle*. Os signos de controle não necessariamente aparecerão na interface, mas são necessários para o entendimento e tomada de decisão da equipe de *design*

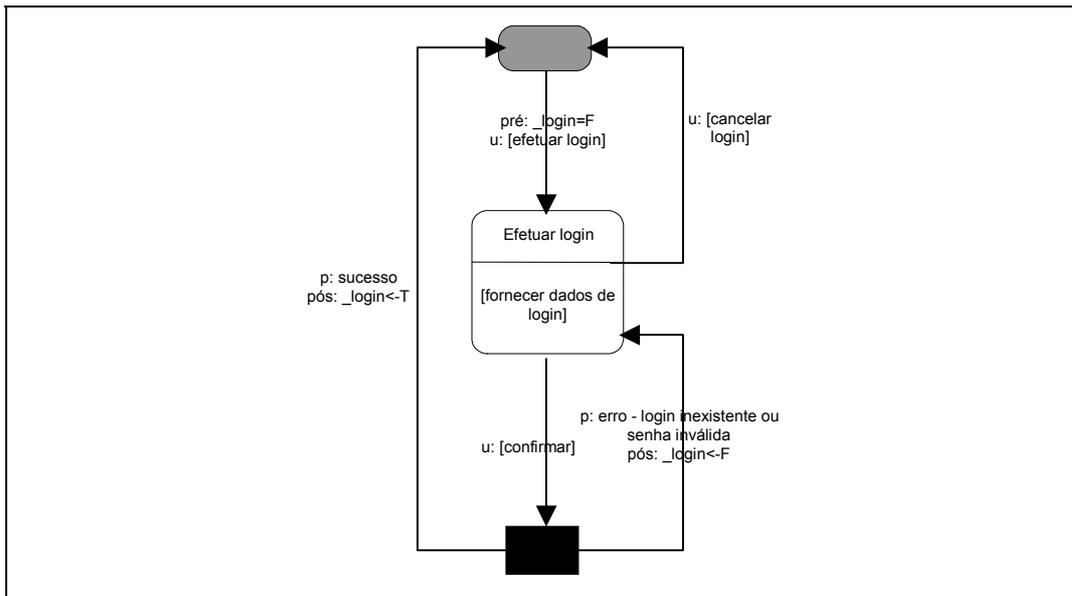


Figura 4.4: Parte do diagrama de interação referente à tarefa *Efetuar login*.

A correspondência entre cenas do modelo de interação e tarefas e metas do modelo de tarefas pode ser feita utilizando-se um retângulo de cor cinza representando a(s) tarefa(s) ou meta(s) que engloba(m) a(s) cena(s). A identificação desta(s) tarefa(s) ou meta(s) pode ser expressa no canto inferior esquerdo do retângulo (Figura 4.5). Este elemento é utilizado para aumentar a rastreabilidade entre os modelos, facilitando a avaliação do impacto de mudanças corretivas ou evolutivas que possam ocorrer, bem como a manutenção da consistência entre modelos, a cada nova versão.

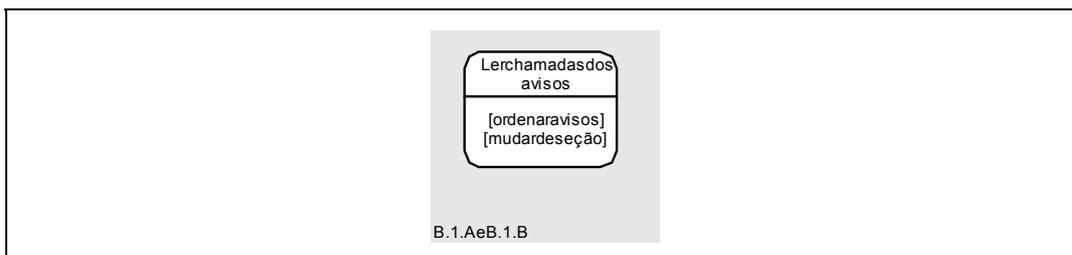


Figura 4.5: Correspondência entre tarefas, metas e cenas.

As possíveis rupturas na comunicação usuário-preposto do *designer* detectadas na modelagem de tarefas (seção 3.3) devem aparecer nas conversas projetadas pelo *designer* no modelo de interação. Por exemplo, o *TRATAMENTO APOIADO: o critério deve ser informado* detectado na especificação textual da tarefa *Fornecer critério* para uma busca (Figura 3.18), deve ser previsto em algum diálogo, como fala do preposto. Esta fala deve ser seguida por uma fala do

usuário, que poderá remediar o seu “erro”, informando um critério. Novas rupturas na comunicação podem ser detectadas na modelagem da interação.

O modelo de interação proposto não apresenta detalhes de interface ou de plataforma tecnológica. Entretanto, os *designers* devem levar em consideração tais fatores, pois as características do ambiente computacional em que o sistema deverá ser executado determinam se uma solução é ou não adequada àquele ambiente. Em outras palavras, as decisões de *design* dependem do estilo de interface e da tecnologia, mas não de sua representação no modelo de interação.

O poder do modelo de interação está na possibilidade de construção de diferentes soluções de interação para um mesmo conjunto de tarefas e metas. A análise destas diferentes soluções – pelos *designers*, clientes e usuários – leva a reflexões sobre soluções alternativas de *design*, possibilitando uma tomada de decisão mais informada e, provavelmente, melhor para a situação em questão.

Pode-se exemplificar uma simples tomada de decisão com base no modelo de interação através das seguintes situações, verificadas em dois projetos distintos que utilizaram a notação proposta:

Exemplo 1: Resultado negativo de busca por um aviso

Quando nenhum aviso é encontrado em uma busca, na solução inicial o usuário receberia uma mensagem indicando o insucesso da busca, e então seria levado novamente à cena *Buscar avisos*, para tentar realizar nova busca (Figura 4.6a). Esta solução, embora bastante comum no ambiente Windows, era inadequada ao ambiente Web, no qual a “razão custo/clique” é mais alta. Sendo assim, optou-se por retornar diretamente à cena *Buscar avisos* (Figura 4.6b). Note que esta cena, ao ser apresentada ao usuário, pode sofrer mudanças que indiquem o ponto de origem desta conversa: 1) uma fala do usuário solicitando uma busca; ou 2) uma fala do preposto do *designer*, em cujo caso deve apresentar uma mensagem indicando o insucesso da busca.

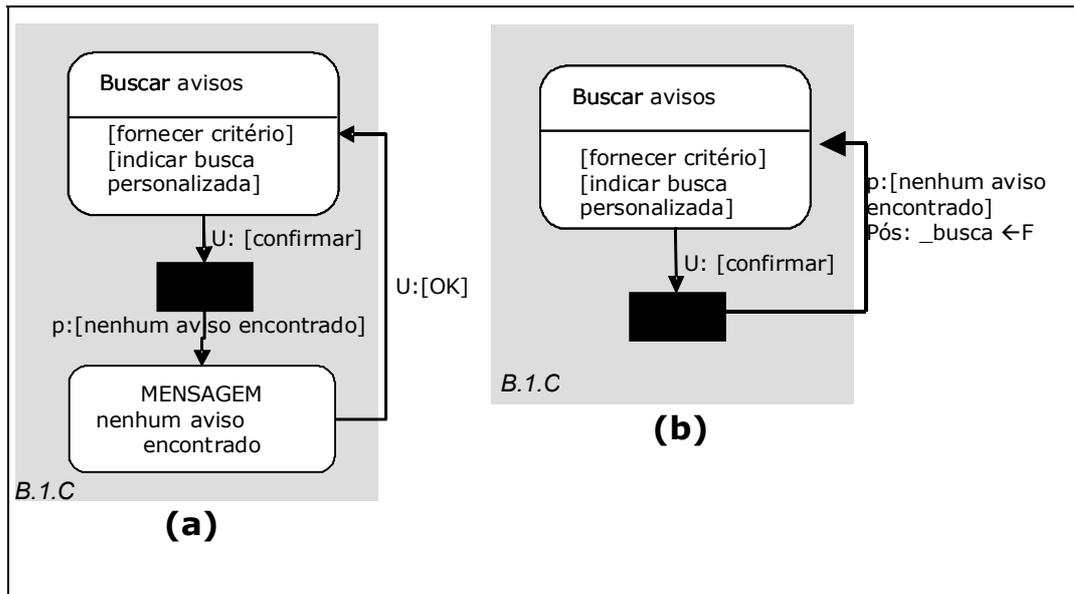


Figura 4.6: Soluções alternativas de interação para indicar resultado negativo de busca (nenhum aviso encontrado).

Exemplo 2: Acesso ao resultado de busca

Na busca por documentos, em um sistema para anotação de documentos na Web (Miceli, 2001), quando há documentos encontrados, a primeira solução de *design* investigada consistia em permitir o acesso ao resultado da busca logo após a própria busca ou a partir da visualização de um dos documentos nela encontrados (destaque na Figura 4.7a). Entretanto, os *designers* julgaram útil manter a lista de documentos encontrados disponível durante toda a sessão, possibilitando futuras consultas do usuário a esta lista (acesso ubíquo para *Examinar resultado da busca*, destaque na Figura 4.7b). Para esta última solução, foi necessário introduzir um signo de controle *_busca*, que indica se há ou não um resultado de busca válido.

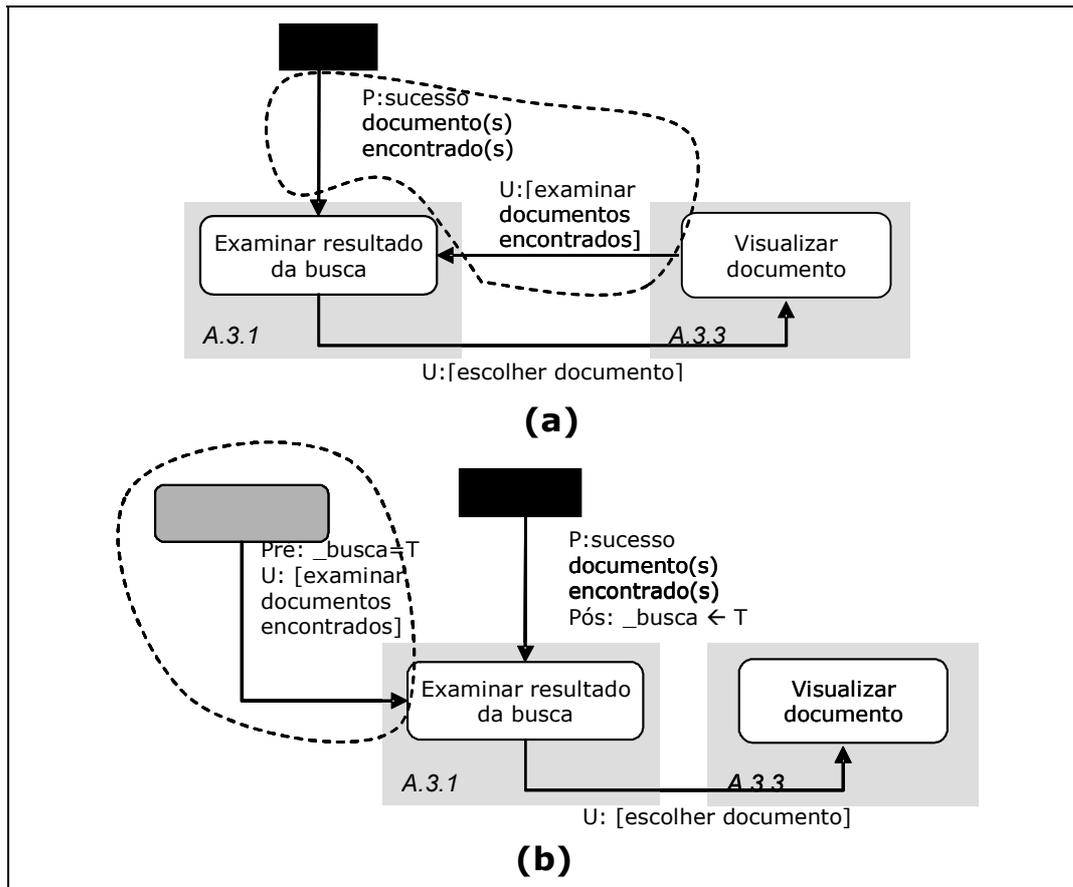


Figura 4.7: Soluções alternativas de interação para indicar acessos distintos ao resultado de busca.

As decisões de *design* que são tomadas durante a construção do modelo de interação precisam, com frequência, ser apresentadas aos usuários para sua apreciação. Para isto, pode ser necessário gerar cenários que reflitam as alternativas de solução propostas, para facilitar a discussão com os usuários.

Como complementação do diagrama de interação, é necessário definir em detalhes cada diálogo, incluindo os signos envolvidos, os tipos de elementos de interface correspondentes (texto livre, escolha simples, escolha múltipla, rótulo, etc.) e informações adicionais como obrigatoriedade de signos e presença de valores *default*. A partir deste detalhamento, torna-se mais direta a construção dos *storyboards* correspondentes a cada cena.

Para não poluir o diagrama de interação com esses elementos, propõe-se que o diagrama seja complementado com uma especificação textual. Os signos que aparecem na especificação textual do modelo de interação são os mesmos que estão representados no modelo de tarefas. Enquanto no modelo de tarefas eles

aparecem ligados a uma tarefa, no modelo de interação eles aparecem como falas de usuário ou preposto, contextualizadas pelos diálogos e cenas que compõem.

A Figura 4.8 apresenta a especificação textual da cena *Buscar Avisos* (Figura 4.6). No diálogo [*fornecer critério*], o signo envolvido é *critério*. Como este critério será fornecido pelo usuário, seu tipo é definido como <texto livre>. Além disto, é obrigatória a informação deste signo pelo usuário. Na especificação textual isto é representado da seguinte forma: <texto livre: obrigatório>.

Cena: Buscar Avisos

A [fornecer critério]

critério? <texto livre: obrigatório>

B [indicar busca personalizada]

busca_personalizada? <escolha simples: obrigatório>

Figura 4.8: Especificação textual da cena *Buscar Avisos*.

O objetivo do modelo diagramático de interação é dar uma visão global aos *designers* do discurso interativo como um todo. A especificação textual fornece os detalhes de cada conversa. O modelo de interação facilita a reflexão do *designer* sobre suas tomadas de decisão, pois ele consegue se imaginar tanto no papel do usuário quanto no do seu preposto, ao projetar ou verificar as falas de cada um, nos atos comunicativos. Na Figura 4.9 pode-se verificar uma parte do modelo diagramático de interação do sistema de Quadro de Avisos.

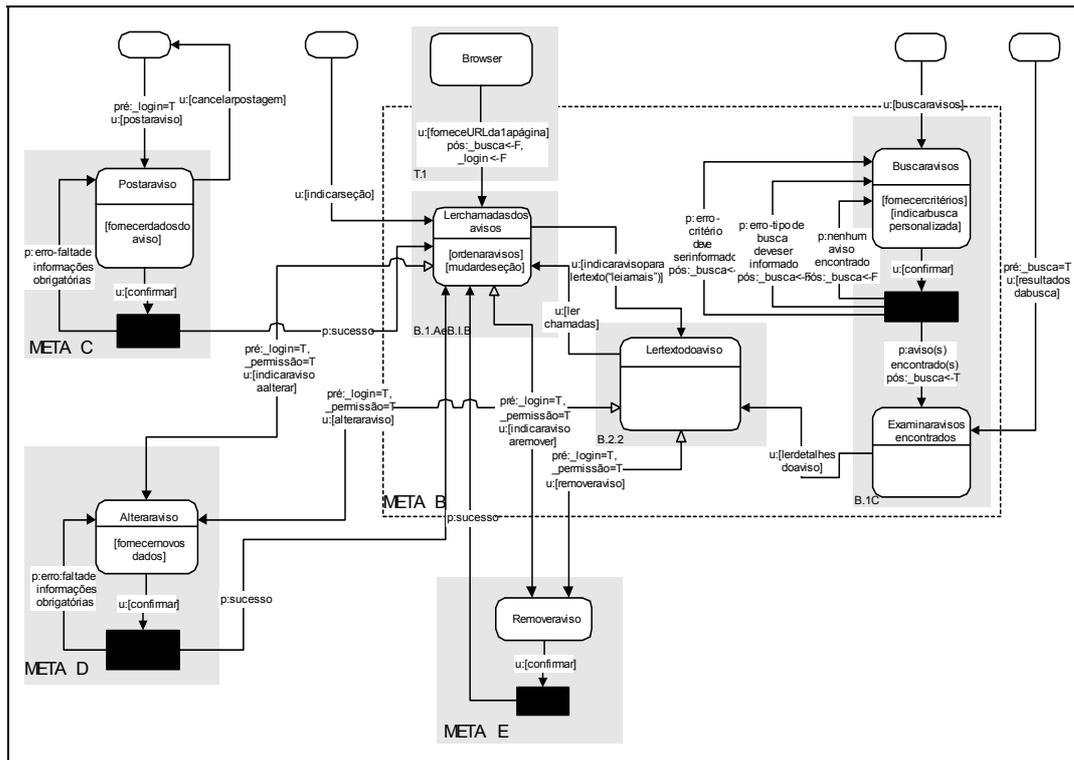


Figura 4.9: Parte do modelo de interação do Quadro de Avisos.

4.2 Uso do Modelo para Avaliação de Aplicações

Este capítulo descreveu até o momento o uso do modelo de interação para **projetar** aplicações interativas. Esta seção descreve brevemente a possibilidade de se **avaliar** aplicações existentes através da análise do modelo de interação. Acredita-se que seja possível realizar uma avaliação por inspeção baseada neste modelo, como exemplificado a seguir.

A Figura 4.10 apresenta um modelo de interação resultante da engenharia reversa de uma aplicação bancária Web existente. O modelo foi construído a partir da visita e navegação pelo site do banco, e não é o único modelo possível (foi construído de acordo com o ponto de vista do avaliador). Este modelo representa a parte de consulta de saldo. Como pode ser visto, o usuário precisa efetuar login, fornecendo informações em dois momentos distintos (cena *Fornecer dados e Informar senha*). Se o login for efetuado com sucesso (*P: sucesso*), o usuário deve escolher um tipo de operação na cena *Visualizar operações (U: [escolher operação consultar])*. Depois da escolha da operação desejada, no caso consultar, o usuário deve escolher qual o tipo de consulta desejada na cena *Visualizar*

operações e tipos de consultas (U: [escolher tipo consulta - saldo]). O usuário então poderá examinar o saldo na cena *Visualizar saldo e operações*.

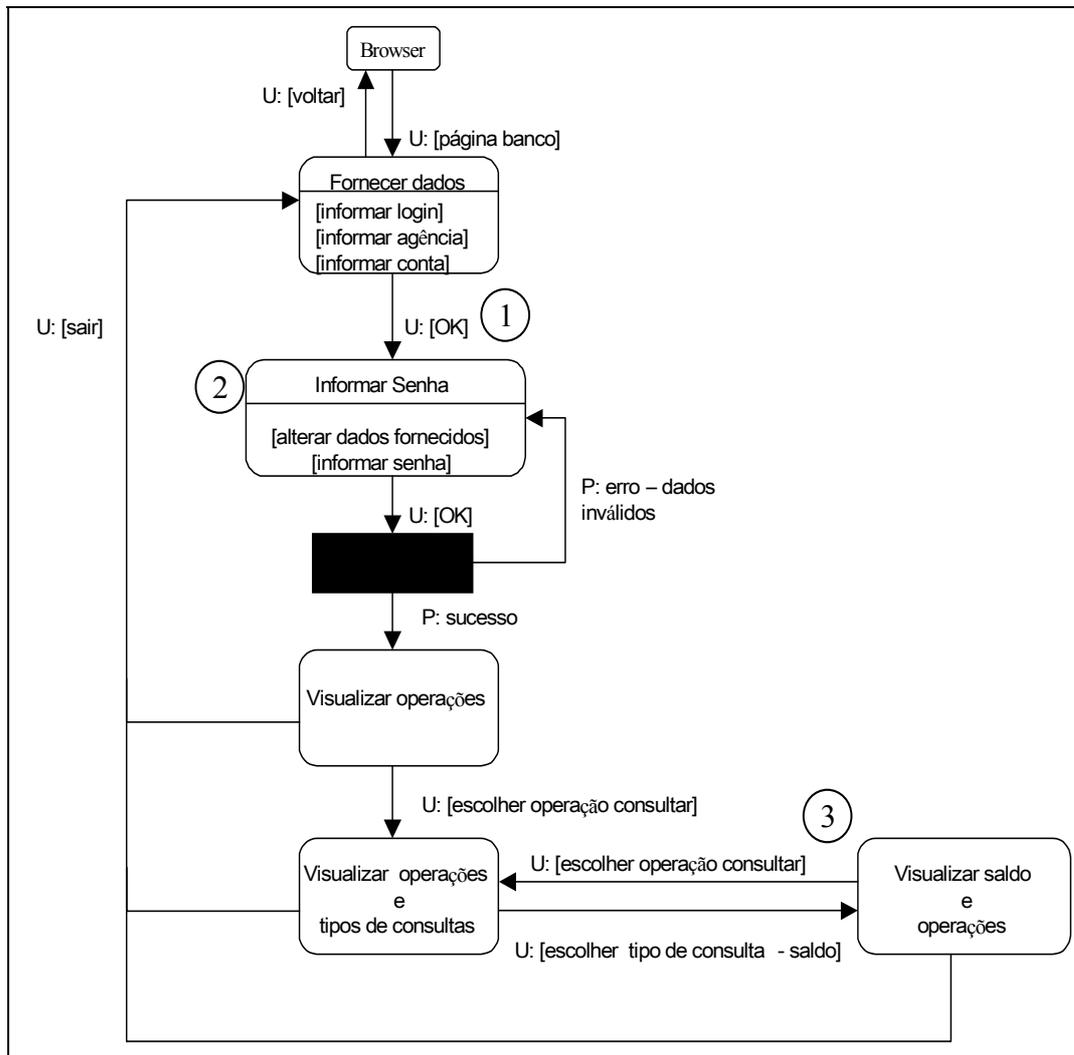


Figura 4.10: Modelo de interação da consulta de saldo em um banco através da Web.

Após a análise do modelo de interação, verificou-se alguns problemas nesta parte do sistema bancário. O primeiro é na transição da cena *Fornecer dados* para a *Informar senha*. Pode-se perceber que não existe um processamento do sistema entre estas cenas (indicação 1 na figura). Isto significa que os dados fornecidos pelo usuário na primeira cena (no caso da aplicação, na primeira tela) não são validados antes do acesso à segunda cena (segunda tela), ou seja, o preposto não “ouve” o usuário (apesar de ter tido oportunidade para isto). Então, se o usuário informou a sua conta errada (na primeira tela), por exemplo, ele poderá informar a sua senha (na segunda tela) e só aí um processo de validação ocorrerá. Vale

observar que nesta última cena, *Informar senha*, existe a possibilidade de alterar os dados fornecidos na primeira cena.

Um outro problema percebido é que na cena *Informar senha* não existe a opção *sair* ou *voltar* (indicação 2). Então, se o usuário quiser cancelar a tarefa, ele tem que utilizar os botões do navegador Web. Não existe um cancelamento através do sistema.

O último problema detectado ocorreu na escolha dos tipos de operações e consultas desejadas. Como se pode ver no modelo, quando o usuário está na cena *Visualizar saldos e operações*, e deseja escolher outro tipo de consulta, como por exemplo, consulta de extrato, ele, apesar de já ter escolhido a opção consultar, precisa novamente indicar que quer consultar algo, para depois escolher outro tipo de consulta (indicação 3).

Como foi visto na descrição acima, o modelo de interação auxiliou o avaliador na detecção de alguns problemas de interação e inconsistências na aplicação. Embora o uso do modelo para avaliação de interação tenha sido pouco explorado, um desdobramento interessante deste trabalho consiste em investigar a utilidade deste modelo na atividade de avaliação. Em outras palavras, se a avaliação baseada neste modelo provê indícios sobre possíveis problemas na conversa usuário-preposto do *designer*, que tipos de indícios podem ser coletados, e que recomendações para reprojeto podem ser extraídas desta análise.