

### 3

## Revisão da Primeira Edição da MoLIC e Propostas Preliminares de Extensão

Tendo em mente os fundamentos da MoLIC apresentados no capítulo anterior, este capítulo apresenta uma revisão da primeira edição da MoLIC utilizando como exemplo alguns dos estudos de caso realizados. À medida que apresentamos esta revisão, nos deparamos com limitações, lacunas e questões quanto ao uso da linguagem ainda não endereçadas ou que ainda estavam pendentes na primeira edição da MoLIC. Assim, no decorrer desta revisão foram levantadas 18 questões que serão discutidas na medida em que surgem. Como certas respostas exigiram extensões à MoLIC, discutimos as propostas preliminares de extensão (Coelho e Barbosa, 2003; Silva e Barbosa, 2004) procurando relacionar e diferenciar os elementos da MoLIC estendida, com base nos seus fundamentos (capítulo 2.2).

Em um esforço para facilitar a transição entre a atividade de análise e o projeto de interface, Paula propôs um passo intermediário: a modelagem da interação utilizando a *Modeling Language for Interaction as Conversation* (MoLIC) (Paula, 2003; Barbosa e Paula, 2003), uma linguagem de modelagem da interação humano-computador fundamentada na teoria da engenharia semiótica (de Souza, 2005). MoLIC permite ao designer modelar todas as possíveis conversas que usuário pode (ou deve) ter com o sistema (preposto do designer) para alcançar seus objetivos. Enquanto o designer projeta a conversa usuário-preposto, ele é estimulado a assumir seu papel de participante da conversa para comunicar sua visão sobre a interação (vide seção 2.2.1) através das falas do seu porta-voz, o preposto. Assim, o espaço de design que MoLIC pretende endereçar é aquele definido pela engenharia semiótica (Figura 5), onde a interação humano-computador é considerada a metacomunicação designer-usuário mediada pelo sistema.

Em linha com a perspectiva de reflexão em ação para o design defendida por Schön (1983), MoLIC foi proposta como uma ferramenta epistêmica com a qual o designer pode refletir sobre a solução de interação enquanto a concebe. Durante o projeto de interação, o designer pode conversar com os artefatos de MoLIC (*conversation with materials*) propondo diferentes soluções de interação, analisando-as e decidindo como a interação deverá ocorrer. Além da reflexão em ação individual, MoLIC também representa as decisões de projeto em uma forma que pode ser facilmente compartilhada, analisada e discutida pela equipe multidisciplinar de desenvolvimento (Paula e Barbosa, 2004; Paula et al., 2005). Portanto, seja para uso pessoal ou compartilhado com a equipe de projeto, “MoLIC foi concebida para uso humano, o que não significa ser [prioritariamente] um modelo formal processável por computador” (Paula et al., 2005 p. 1714).

Antes de começar a etapa de projeto de IHC utilizando a MoLIC, o designer deve ter passado por uma etapa de análise para conhecer os elementos envolvidos na (meta) comunicação designer-usuário – os interlocutores, o canal, o código, a mensagem, e o contexto – bem como os objetivos de comunicação dos interlocutores. Na etapa de análise, cenários (Carroll, 1995) e modelos de tarefas (Paternò, 2000; Card et al., 1983) têm sido amplamente utilizados para auxiliar o designer a conhecer o espaço de design de IHC. Paula (2003) discute em detalhes como estas representações podem servir de insumo para o projeto de interação humano-computador baseado na engenharia semiótica. Neste trabalho é suficiente considerarmos que o resultado da análise fornece ao designer o conhecimento necessário do espaço de design de IHC, incluindo (parte de) os signos envolvidos.

Segundo Paula (2003), cenários e modelos de tarefas podem dificultar ao designer compreender as relações existentes entre tarefas e objetivos do usuário, pois não lhe fornecem uma visão global da interação. Por exemplo, imagine que o designer tenha descrito três cenários e/ou três modelos de tarefas para as tarefas *Consultar Dados do Cliente*, *Remover Cliente* e *Alterar Dados do Cliente*. Como cada tarefa (ou cenário) está descrita separadamente, o designer pode dificultar a interação exigindo que o usuário consulte os dados do cliente num momento de interação (num lugar da interface), para somente em outro momento (outro lugar da interface) não acessível do primeiro remover o cadastro do cliente. Tomada esta decisão, que desconsidera as relações entre as tarefas *Consultar Dados do*

*Cliente e Remover Cliente*, o designer também pode incluir inconsistências na solução de interação se ele vier a considerar a relação entre as metas *Consultar Dados do Cliente* e *Alterar Dados do Cliente*, isto é, se ele vier a decidir que assim que o usuário consultar os dados do cliente (num lugar da interface), poderá alterar os seus dados (no mesmo lugar da interface, ou em outro lugar acessível do primeiro). “Esta limitação [dos cenários e modelos de tarefas] atrapalha o projeto de um artefato comunicativo coerente, um requisito essencial para engenharia semiótica” (Barbosa e Paula, 2003 p. 19).

Scogings e Phillips (2001) fazem uma crítica semelhante a modelos de tarefas e, por analogia, também a cenários:

Uma crítica das notações de diálogos existentes é que a relação/ligação entre as tarefas do usuário é freqüentemente obscura (Brooks, 1991). [Modelos de tarefas] definem a estrutura e comportamento do diálogo mas nem sempre revelam as interconexões entre os componentes do diálogo durante a execução de tarefas de um nível mais alto [, mais abstrato]. Isto é, não representam tarefas dentro do *contexto* da estrutura de diálogo (a coleção dos diálogos e dos sub-diálogos que constituem a interação).

Os modelos de diálogos são importantes porque fornecem ao designer meios de descrever a estrutura e o comportamento da interface com usuário sem a aparência e estilo (“look and feel”). Eles permitem a análise do diálogo com relação às tarefas que devem ser suportadas, com respeito à eficiência e à completude. (Scogings e Phillips, 2001 p. 71)

Fundamentado na engenharia semiótica, Leite (1998) propôs a *Linguagem de Especificação da Mensagem do Designer* (LEMD) antes da MoLIC. A LEMD tem por objetivo especificar de forma abstrata (sem detalhes de interface) o comportamento da interface, definindo as possíveis *ações e visualizações* do usuário sobre ela; sem, no entanto, fazer uso da metáfora de interação como uma conversa<sup>1</sup>. Com base neste trabalho, Dahis (2001) propôs um modelo para a especificação de cenários sob uma perspectiva conversacional também fundamentada na engenharia semiótica. No entanto, era uma especificação textual rica em detalhes e carecia de uma visão global da interação. Para tentar resolver esta limitação, MoLIC propõe representar a interação em um modelo

---

<sup>1</sup> Para Leite, “o modelo de interação visa definir quais as *interações básicas* que o usuário precisa desempenhar (...). Uma *interação básica* é a ação que o usuário desempenha com algum *objeto de interação* (um *widget*), possivelmente com o auxílio de uma ferramenta de acionamento. (...) Uma *estrutura de articulação de interações* descreve a maneira como as *interações básicas* podem ser realizadas de modo a formar um comando.” (Leite, 1998 p. 118)

diagramático, complementado por uma especificação textual, mas não tão detalhada quanto a proposta de Dahis. No modelo diagramático, o relacionamento entre os objetivos do usuário podem ser observados sob uma visão global da interação, favorecendo a reflexão do designer sobre a solução sendo concebida.

Assim sendo, a ferramenta epistêmica MoLIC é composta por quatro artefatos inter-relacionados: (1) um diagrama de metas, (2) uma ontologia de signos e (3) um diagrama de interação complementado por uma (4) especificação textual. O diagrama de metas organiza hierarquicamente as metas do usuário, indicando o que ele pode realizar com o apoio do sistema. A ontologia de signos, por sua vez, deve definir e organizar todos os conceitos envolvidos com o sistema, incluindo os artefatos ou informações envolvidos em cada ação realizada pelo usuário ou externos ao sistema<sup>2</sup>. O diagrama de interação define, com o mínimo de complexidade necessária, como as metas do usuário podem ser alcançadas durante sua conversa com o preposto do designer (sistema); para, então, ser detalhado na especificação textual.

É importante lembrar que a MoLIC foi proposta como ponte entre representações da atividade de análise e a definição da interface propriamente dita. Deste modo, espera-se que o designer utilize a MoLIC e as representações empregadas na atividade de análise (por exemplo, cenários e modelo de tarefas) sem ainda pensar em detalhes de interface, como, por exemplo, os *widgets* que compõem a interface ou comportamentos de baixo nível (clicar, arrastar, soltar, digitar e etc). Depois de definida a interação, o designer continua seu trabalho modelando uma interface que permita ao usuário e ao designer, na figura do seu preposto, conversarem conforme o projetado, e neste momento, sim, os detalhes de interface devem ser considerados.

---

<sup>2</sup> Atualmente não existe trabalho que proponha a ontologia de signos da MoLIC. O que existe é apenas uma proposta de *tabela de signos* que lista os signos usados durante a interação e os seus atributos. Quando for proposta, a ontologia de signos irá definir classificações de tipos de signos, relações entre estes tipos e regras que restringem como estes signos podem ou devem ser utilizados. Por exemplo, vamos considerar as manipulações/operações que um usuário poderá ou deverá realizar sobre um signo X. Se a única operação for examinar o valor de X, deverá haver na ontologia regras como as seguintes: (1) com relação ao conteúdo: nenhum diálogo pode definir X como signo cujo valor seja definido pelo usuário; (2) com relação à expressão: o elemento de interface que poderá ser associada a X deve ser *read-only*.

Este trabalho trata principalmente do diagrama de interação e da descrição textual, gerando alguns requisitos para a ontologia de signos. O diagrama de metas não será abordado aqui por considerarmos-lo até o momento suficientemente tratado nos trabalhos de Paula (Paula, 2003; Barbosa e Paula, 2003). Na seção 3.1 apresentaremos os elementos básicos do diagrama de interação e, na seção 3.2, a descrição textual, com o intuito de levantar questões quanto ao uso da linguagem endereçadas neste trabalho. Não é objetivo deste trabalho apresentar na íntegra a primeira edição da MoLIC; somente serão apresentados os itens fundamentais para a compreensão das extensões que irão compor a sua segunda edição.

### 3.1 Diagrama de Interação

Na perspectiva conversacional de interação adotada pela MoLIC, interagir é essencialmente conversar. Uma conversa pode ser vista como uma seqüência de falas cujos interlocutores estão engajados em contribuir com o objetivo compartilhado da conversa (Marcuschi, 1999). É comum, por exemplo, nos engajarmos em uma conversa para orientar um visitante que não conhece determinado lugar, para trocar boas receitas de culinária, ou para compartilhar com um amigo sua situação problemática. Durante uma conversa natural (somente com interlocutores humanos), os interlocutores vão definindo (e redefinindo) o objetivo compartilhado da conversa à medida que elaboram suas falas em função do que já foi dito e do seu objetivo individual naquele momento. Logo, uma conversa é naturalmente um processo dinâmico onde os indivíduos interagem de forma livre e criativa, sem passos e final previsíveis.

Diferentemente de uma conversa natural, a conversa que ocorre em IHC – a metacomunicação designer-usuário mediada pelo sistema, segundo a engenharia semiótica – terá sido projetada pelo designer, isto é, todos os possíveis passos e finais serão pré-definidos (são os grafos do autômato que controla a interação). Em outras palavras, todos os possíveis tipos de falas que o usuário e o designer (na figura do seu preposto) podem emitir durante a conversa fazem parte do projeto do discurso interativo. Isto se faz necessário porque todas as contribuições do designer para esta conversa serão cristalizadas em um sistema computacional que sempre executa um determinado conjunto finito de instruções. Assim, um dos interlocutores – o preposto do designer – literalmente “pensa como uma

máquina”, não dispondo de características humanas importantes para a comunicação em curso: semiose ilimitada, criatividade e etc. Portanto, o sistema interativo que possibilitar a conversa usuário-preposto não somente necessita conter todas as possíveis contribuições do designer para esta conversa, como também deve permitir que o usuário contribua em todas, e somente, as formas projetadas pelo designer.

Para projetar algo naturalmente não projetável e imprevisível como uma conversa, MoLIC foi buscar conhecimento em áreas que estudam este fenômeno de interação humana: a análise de discurso (Brown e Yule, 1983) e a análise de conversação (Marcuschi, 1999). Mesmo que durante a conversa normalmente não se pense em sua estrutura, a análise de discurso procura estruturar o discurso para facilitar seu estudo. “O dado estudado em análise de discurso é sempre um fragmento do discurso e o analista sempre tem que decidir onde o fragmento começa e termina” (Brown e Yule, 1983 p. 69). Se o designer puder projetar fragmentos de conversa, e se existir uma forma de estruturá-los para formar uma conversa inteira, então, este seria um caminho para projetar a conversa usuário-preposto.

Dado que o designer necessita projetar a conversa desde os objetivos dos interlocutores, principalmente os do usuário, até os signos manipulados nas falas, MoLIC utilizou diferentes fragmentos propostos na análise de discurso, cada qual com um nível de abstração e foco. A estrutura da conversa considerada pela MoLIC é a seguinte:

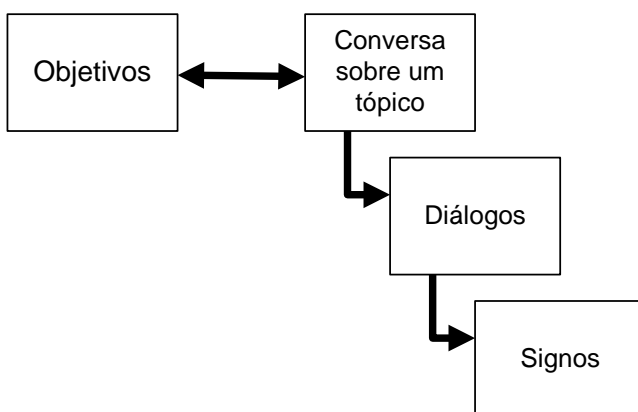


Figura 8: Estrutura da conversa na MoLIC original.

Considera-se que, durante a interação, o usuário e o preposto do designer estejam engajados em colaborar com os objetivos (ou metas) do usuário

presumidos pelo designer. Assim sendo, eles conversam sobre algum assunto ou *tópico* para colaborar com um determinado objetivo do usuário. A noção de *tópico* como “o que está sendo falado/escrito” (Brown e Yule, 1983 p. 73) durante a conversa permite que o designer defina um discurso interativo coerente, ou seja, permite que ele defina quais diálogos estão relacionados e formam algum tipo de estrutura, distinguindo aqueles que estão separados desta estrutura. É importante notar que o designer se considera conversando sobre um determinado tópico, mas não necessariamente significa que o usuário se considere conversando sobre tal tópico. Deste modo, Barbosa e da Silva (2001) adaptaram esta noção de tópico em IHC para: “sobre o quê eu [designer] acho que nós [designer (preposto) e usuário] estamos conversando” (p. 21). À medida que a conversa evolui, os participantes mudam de tópico em função do objetivo em questão.

Aproximando-se da liberdade de uma conversa natural, MoLIC também valoriza o oportunismo na interação, ao invés de valorizar exclusivamente o cumprimento de uma seqüência de passos de uma tarefa. Com esta postura, MoLIC admite um uso mais consciente e criativo daquilo que foi projetado. Considera-se, então, que o usuário possui um objetivo inicial que pode evoluir e se modificar durante a interação. Logo, tanto o objetivo em questão influencia o tópico da conversa, quanto o oposto, por isto esta inter-relação foi representada por uma seta bidirecional na Figura 8.

Uma conversa geralmente percorre diferentes *tópicos* em torno dos quais os interlocutores travam um conjunto de *diálogos*. Um diálogo é composto por falas organizadas em pares conversacionais (Marcuschi, 1999), emitidas ora por um interlocutor ora por outro. Por fim, cada fala transmite uma mensagem composta por um conjunto de signos, no nosso caso signos de interface. Com base nesta estrutura de conversa, MoLIC possui um conjunto de elementos para representar a conversa usuário-preposto no diagrama de interação. Os principais elementos do diagrama são:

- cenas;
- processos do sistema;
- falas de transição; e
- acesso ubíquo

## Cena

Uma *cena* representa uma conversa entre o usuário e o preposto do designer sobre uma certa matéria, assunto, ou *tópico*. O tópico da cena geralmente está associado a (parte de) uma meta ou a (parte de) uma tarefa do usuário, indicando aquilo que usuário pode realizar (conversar) naquele momento de interação (Paula, 2003; Barbosa e Paula, 2003). Pode-se considerar que o tópico de uma cena corresponde à fala inicial do preposto naquele momento, informando ao usuário sobre o que ele poderá falar ali.

Uma cena pode conter *diálogos*. Um diálogo é composto por falas do usuário e do preposto organizadas em pares conversacionais, de acordo com algum subtópico da cena. A conversa descrita por uma cena poderá ocorrer em um elemento estruturado de interface, tal como uma tela, janela ou página *web*.

Uma cena é representada no diagrama de interação por um retângulo com bordas arredondadas e uma identificação do seu tópico. Por exemplo, a cena de tópico *Buscar documento* está representada em sua forma mínima (sem diálogos) no lado esquerdo da Figura 9, e com seu compartimento de diálogos contendo o diálogo *[informar critério de busca]* no lado direito desta figura.

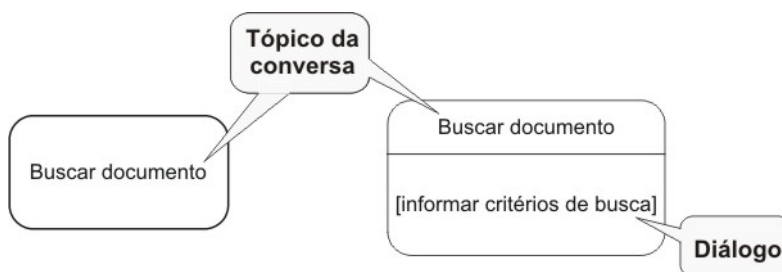


Figura 9: Exemplo de cenas no diagrama de interação.

Dispor os diálogos no compartimento de diálogos da cena não indica a ordem em que eles devem ser travados, não aponta os diálogos mutuamente exclusivos, nem sugere alguma pré-condição para que eles ocorram dentro da cena. Todavia, existem situações em que estruturar os diálogos é importante, como, por exemplo, quando o designer precisa indicar a seqüência dos diálogos de um *wizard*, ou quando ele decide que o usuário deve escolher entre utilizar um tamanho predefinido de página (A4, A3, etc.) ou informar um tamanho personalizado, ou ainda quando ele precisa indicar que um determinado conjunto



de diálogos somente é possível ser travado por uma classe específica de usuários. Então, em situações onde o designer sentir necessidade,

**Questão 1.** Como definir a estrutura de diálogos dentro da cena?

Uma estrutura de diálogos deve definir: (1) a ordem em que os diálogos devem ser travados, (2) quais diálogos são mutuamente exclusivos, (3) se existe alguma pré-condição para que eles ocorram dentro da cena, e que pré-condição é esta.

Para definirmos a ordem dos diálogos, poderíamos seguir a notação proposta por Paula (2003) para representar a seqüência de tarefas no modelo hierárquico de tarefas adaptado por ela. Ela utilizou uma seqüência numérica para indicar a seqüência em que as tarefas devem ser executadas e interrogações para indicar independência de ordem na execução. Se seguirmos esta mesma idéia no caso de uma cena com quatro diálogos, por exemplo, poderíamos representar que os quatro diálogos podem ser travados completamente independentes de ordem (Figura 10a), em seqüência (Figura 10b) ou ainda o primeiro e último diálogo dependente de ordem, mas os intermediários não (Figura 10c) (ou seja, podem ser travados na ordem 1,2,3,4 ou 1,3,2,4). Apesar da possibilidade de uso desta notação, é importante notar que Paula não chegou a propor uma forma de estruturar diálogos.

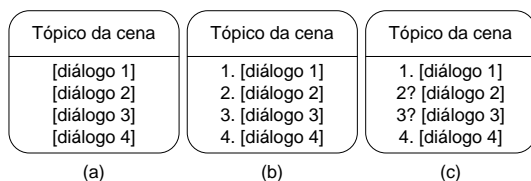


Figura 10: Possível representação da seqüência de diálogos.

Quando formos definir os diálogos mutuamente exclusivos, também poderíamos seguir a notação que Paula propôs para representar tarefas alternativas. Ela utilizou uma seqüência de letras para indicar que usuário deveria escolher uma tarefa dentre um conjunto para ser executada naquele momento da interação. Seguindo esta idéia, por exemplo, representamos na Figura 11 quatro diálogos completamente independentes de ordem (a), no caso de serem mutuamente exclusivos (o usuário ou trava o diálogo 1 ou 2 ou 3 ou 4) (b), ou o primeiro e último diálogo independentes de ordem e os outros dois mutuamente

exclusivos (o usuário trava o diálogo 1 e 4, mais o 2 ou o 3) (c). Silva e Barbosa (2004) propuseram ainda uma outra opção para representar diálogos mutuamente exclusivos: agrupá-los entre colchetes precedidos pelo operador *ou* (d).

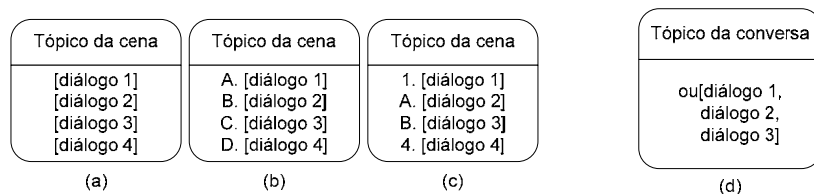


Figura 11: Possíveis representações de diálogos mutuamente exclusivos.

Representar uma pré-condição para cada diálogo da cena seria muito custoso, repetitivo e dificultaria a legibilidade do diagrama de interação, principalmente quando apenas alguns diálogos tiverem pré-condições ou um grupo de diálogos tiver a mesma pré-condição. Deveríamos, então, somente representar as pré-condições quando forem necessárias e representá-las uma única vez quando um conjunto de diálogos tiverem a mesma pré-condição (os diálogos seriam agrupados). Por exemplo, a Figura 12 apresenta uma cena onde o [diálogo 1] só pode ser travado pelo administrador do sistema e os demais diálogos podem ser travados por todos os usuários.

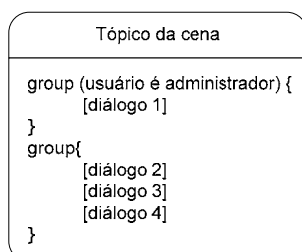


Figura 12: Possível representação de pré-condições para diálogos.

Esta discussão sobre a *estruturação dos diálogos* nos faz lembrar das *estruturas de articulação da interação* proposta pela LEMD (Leite, 1998). Antes de compararmos estas propostas, temos que ter em mente o objetivo de cada uma delas. A estruturação de diálogos tem como foco a organização da conversa descrita pelos diálogos, isto é, seu objetivo é organizar os sub-tópicos, o(s) sub-conteúdo(s) da conversa. Já a estrutura de articulação de interações “descreve a maneira como as interações básicas podem ser realizadas”; onde “uma interação básica é a ação que o usuário desempenha com algum objeto de interação (um *widget*)” (Leite, 1998 p. 118). Em outras palavras, a estruturação de articulação da

interação tem como foco a organização da expressão da conversa na interface, ao descrever ações sobre *widgets* abstratos. Desta forma, poderíamos dizer que uma estrutura de articulação da interação é uma forma de manifestação, de execução na interface da conversa descrita por um diálogo.

Para decidirmos qual será a representação da estrutura dos diálogos na segunda edição da MoLIC, não basta analisarmos os prós e contras destas propostas, devemos ainda considerar a Questão 8. Ela levanta a possibilidade de uma fala de transição depender de um ou mais diálogos da cena de origem. Portanto, a estrutura de diálogo também deve facilitar a representação da obrigatoriedade de um ou mais diálogos para a fala de transição poder ser emitida. Repare que não tínhamos pensado nesta necessidade antes de nos engajarmos na elaboração de uma proposta única: a segunda edição da MoLIC.

\*\*\*

Existem cenas especiais onde não ocorrem diálogos de fato. Na verdade, elas somente apresentam uma mensagem do preposto em resposta a um processo do sistema. Em seu trabalho original, Paula (2003) representou esta cena como um retângulo de bordas arredondadas contendo a mensagem do designer representada entre << >> (Figura 4.2 em Paula, 2003) ou precedida pela palavra *MENSAGEM* (Figura 4.6 em Paula, 2003) Os exemplos utilizados por Paula estão respectivamente reproduzidos a seguir:

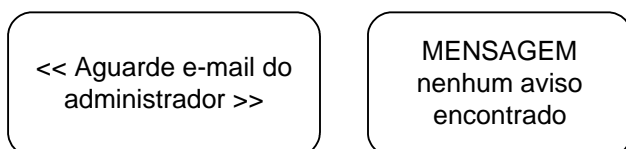


Figura 13: Cenas especiais somente com uma mensagem do designer (*retirado de Paula, 2003*).

Pela forma de representação proposta por Paula (2003), percebe-se que estas mensagens do designer tem conteúdo extensional, ou seja, mensagens com conteúdo definido durante o projeto de interação que serão cristalizadas, expressas na interface no momento da compilação do sistema (ou de construção do hardware, se for o caso). Isto quer dizer que o usuário receberá a mesma mensagem (com o mesmo conteúdo) sempre que a conversa descrita nestas cenas ocorrer. No exemplo da Figura 13, sempre lhe seria apresentado algo como <<Aguarde e-mail do administrador>> ou <<nenhum aviso encontrado>> nas

respectivas circunstâncias de interação. Porém, quando for o caso do designer apresentar para o usuário o resultado de uma busca por documentos para sua simples conferência, a mensagem do designer terá seu conteúdo definido intensionalmente, isto é, com conteúdo e, por conseguinte, expressão determinados de fato durante a interação. Então,

**Questão 2.** Como representar mensagens do designer com conteúdo definido extensional ou intensionalmente?

Uma solução quase imediata é representar os signos que compõem a mensagem do designer com conteúdo intensional, ao invés de representar uma mensagem extensional entre << >>. Deste modo, a mensagem do designer resultante da busca por documentos poderia ser, por exemplo, <<*Nenhum documento encontrado*>>, uma mensagem de conteúdo extensional (Figura 14a), ou *conjunto(documento.\*)!<sup>3</sup>*, uma mensagem de conteúdo intensional (Figura 14b).

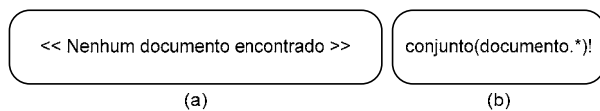


Figura 14: Possível representação de mensagens do designer com conteúdo extensional (a) ou intensional (b).

Paula (2003) definiu uma “mensagem” do designer como sendo uma cena especial onde o designer transmite sua mensagem sem “dialogar” com o usuário (ele apenas “ouve”). Assim, da mesma forma que uma mensagem de conteúdo extensional, uma mensagem de conteúdo intensional também é uma cena especial. Considerando que ambas tiveram origem no mesmo processo do sistema, qual seriam as diferenças e semelhanças entre uma mensagem de conteúdo extensional e um mensagem de conteúdo intensional? Voltemos ao exemplo da Figura 14. Tanto a mensagem “*Nenhum documento encontrado*” quanto a mensagem “*conjunto(documento.\*)!<sup>3</sup>*” tiveram origem num processo do sistema que buscava documentos segundo determinado critério. Como estas duas cenas especiais

tratam do exame de resultado da busca, podemos, então, afirmar que ambas possuem o mesmo tópico da conversa, apesar de não representado. O que muda, na verdade, é o contexto no qual o usuário conversa com o preposto sobre o tópico “examinar resultado de busca”.

Representar diferentes contextos de conversa sobre o mesmo tópico em diferentes cenas (retângulos arredondados) pode levar o designer a interpretações errôneas do diagrama de interação, pois as diferentes cenas não expressam o tópico em comum. Sem a indicação do tópico, o designer poderia pensar que cada cena representa uma conversa completamente diferente, porém, na realidade, as diferentes cenas tratam da mesma coisa, do mesmo tópico, apenas em situações (contextos) diferentes. Se o designer considerar erroneamente cada cena como conversas sobre tópicos diferentes, ele poderia tratar diferentemente partes da conversa que permanecem idênticas ou que se assemelham em diferentes contextos, potencializando, assim, a criação de inconsistências no discurso interativo. Um exemplo simples disto pode ser observado na Figura 15, onde o designer permite ao usuário realizar uma nova busca (fala de transição *u:[nova busca]*) num contexto sem documentos encontrados (Figura 15a), mas não permite (sem fala de transição *u:[nova busca]*) ao usuário realizar nova busca num contexto onde documentos foram encontrados (Figura 15b). Comparar estas duas soluções nesta figura parece simples e de inconsistência facilmente perceptível. No entanto, a percepção de inconsistência não é tão fácil quando estas cenas estão espalhadas por um diagrama de interação mais complexo.

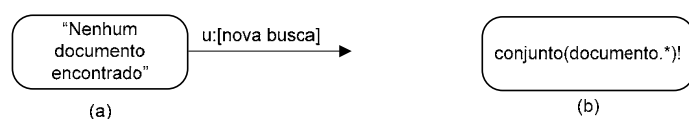


Figura 15: Tratamentos distintos para uma conversa sobre o mesmo tópico em contextos diferentes.

Além disto, é bem comum que estas diferentes mensagens do designer sejam apresentadas no mesmo elemento estruturado de interface (por exemplo,

---

<sup>3</sup> *conjunto(documento.\*)!* – representa um conjunto de documentos (*conjunto(documento)*) onde todos os signos que compõem cada documento (*documento.\**) são apresentado pelo preposto do designer (“!”) durante a conversa.

tela, janela, página web, etc.), mesmo em situações diferentes do contexto. Exemplos corriqueiros desta prática seriam:

- um editor de texto com um documento de 50 páginas ou com um documento de 1 página em branco;
- um navegador web sem página exibida, ou com uma página html simples, ou ainda com uma página web com animação em Flash;
- um programa de *e-mail* (por exemplo, Microsoft Outlook®, Eudora® ou Mozilla Thunderbird®) com caixa de entrada contendo 100 *e-mails* ou vazia; ou
- o Windows Explorer® apresentando um diretório com vários arquivos ou vazio.

Note que em todos estes exemplos os diferentes contextos da mesma cena (com o mesmo tópico) foram implementados na mesma unidade de apresentação. Talvez para comunicar ao usuário que, em condições normais (dentro do contexto mais comum), a conversa sobre aquele tópico ocorre naquele local da interface, mesmo que em determinado momento a conversa possa ser desviada para outro lugar da interface. Seria como se o designer falasse: – É aqui que vamos conversar sobre o tópico X, usuário. Qualquer coisa que você desejar sobre o tópico X comece a conversa por aqui.

Para tentar minimizar este problema, poderíamos representar o tópico da conversa nas mensagens do designer, deixando-as mais parecidas com cenas comuns. A Figura 16 apresenta as mensagens do designer resultante da busca por documentos indicando o tópico da conversa. Esta forma de representação facilita ao designer associar tópicos em diferentes representações da cena, uma para cada contexto de interação. No entanto, a representação dos tópicos nas mensagens do designer ainda não garante que o designer os associe, pois cada tópico pode ser individualmente (1) (re)definido em momentos diferentes do projeto, (2) com nomes diferentes, (3) por pessoas diferentes e (4) ocupar posições distantes no diagrama de interação. Mesmo assim, a possibilidade de equívocos na elaboração e manutenção do diagrama de interação poderia ser minimizada se o designer utilizasse uma ferramenta computacional de edição da MoLIC que gerenciasse a mudança de tópicos.

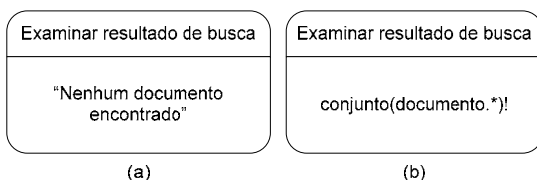


Figura 16: Representação do tópico da conversa nas mensagens do designer.

Uma forma de o designer não correr o risco de considerar uma cena com o mesmo tópico em contextos diferentes como cenas completamente diferentes é representar a conversa nos diferentes contextos sob uma única representação de tópico. Seguindo a proposta de grupos de diálogos com pré-condições, podemos representar uma única cena contendo diferentes grupos de diálogos ou mensagens do designer (Figura 17a) para cada contexto, indicado as respectivas pré-condições do contexto. Além disto, poderíamos investigar uma notação que mantivesse a representação de um único tópico e representasse separadamente a conversa em diferentes contextos (Figura 17b). Entretanto, qualquer uma destas duas representações dificulta a legibilidade do diagrama de interação, a primeira (Figura 17a) mistura e aglomera contextos diferentes e, conseqüentemente, as falas de transição envolvidas, e a segunda (Figura 17b) limita o reposicionamento dos retângulos que antes eram independentes, podendo aglomerar outros elementos do diagrama.

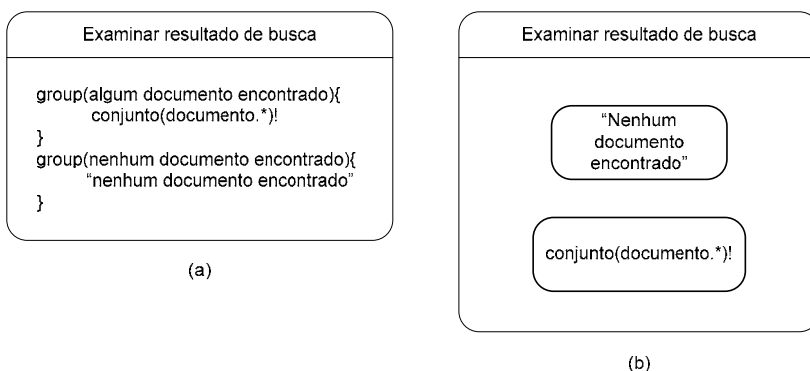


Figura 17: Representações de mensagens do designer sob a indicação única do tópico.

Seria interessante investigar como uma ferramenta computacional de edição de diagramas MoLIC poderia apoiar o designer na verificação de consistência entre contextos diferentes de uma cena. O designer poderia projetar a interação indicando separadamente o tópico em cada mensagem do designer (Figura 16) apoiado por um gerenciador de tópicos da ferramenta computacional. E, somente

quando ele desejasse verificar a consistência do discurso, ele recorreria a uma visão aglomerada da cena, algo semelhante às propostas da Figura 17.

\*\*\*

Na MoLIC original, essa cena especial de mensagem “contém apenas uma fala do preposto” (Paula, 2003 p. 47) e, conseqüentemente, não representa tópico nem possui diálogos. No entanto, existem casos onde o usuário pode pedir que o preposto do designer expresse sua mensagem de outro jeito (um ajuste de código, sintaxe ou foco) antes de prosseguir com a conversa. Por exemplo, imagine que o usuário deseja ordenar os documentos apresentados por uma mensagem do designer em resposta a uma busca realizada, ou escolher quais atributos dos documentos devem estar visíveis. Deste modo,

**Questão 3.** Como representar o ajuste de código, de sintaxe e de foco da mensagem do designer feito pelo usuário?

Em alguns casos, o designer pode permitir que o usuário ajuste o código, a sintaxe e o foco de sua mensagem antes de decidir prosseguir com a conversa em outra cena. Uma forma muito comum de ajuste da mensagem do designer é a reordenação dos elementos apresentados por ela, em função de algum dos seus atributos. Ajustar a mensagem do designer significa ajustar alguma característica dos signos que a compõem. Em termos de conversa, o ajuste de código, de sintaxe e de foco seria equivalente ao usuário pedir-lhe para pronunciá-la novamente com nova forma de expressão (por exemplo, apresentar os elementos da mensagem: em ordem ou não, em forma de lista, em forma resumida (*Thumbnails*) ou com detalhes dos atributos) e/ou restringindo/expandindo o conteúdo apresentado (por exemplo, exibir somente *e-mails* não lidos, o conteúdo do diretório X, ou a página X do documento). Nestes casos, o conteúdo da mensagem não seria modificado porque ele já foi completamente “pensado” no processo do sistema. Aquilo que mudaria, então, é a *forma de expressão* e o *foco* da mensagem do designer.

Assim sendo, o ajuste da mensagem do designer pode ser representado por um conjunto de diálogos que ajustam ou influenciam os signos que a compõem. Estes diálogos podem ser representados num compartimento abaixo do compartimento da mensagem do designer. Por exemplo, a cena *Examinar resultado de busca* (Figura 18) apresenta uma mensagem do designer indicando



os documentos encontrados (*conjunto(documentos.\*!)*). Os diálogos que ajustam a apresentação destes documentos estão definidos no último compartimento da cena: o diálogo [*escolher ordem dos documentos*], [*escolher atributos visíveis*] e [*escolher ordem dos atributos*].

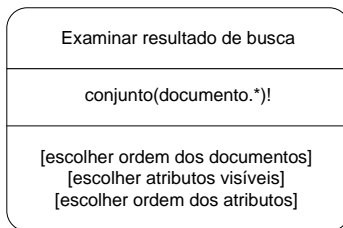


Figura 18: Representação de diálogos que manipulam a mensagem do designer.

\*\*\*

Durante a conversa descrita por uma cena, podem ocorrer interações (diálogos) que não fazem parte do sistema sendo projetado, mas sim do ambiente em que o sistema projetado se insere. Considere uma página *web* projetada pelo designer, por exemplo. O usuário pode manipulá-la usando a barra de rolagem do navegador *web*, aumentar o seu tamanho da fonte no navegador *web*, alterar no sistema operacional a configuração da impressora padrão antes de imprimir nela a página *web*, ou ainda diminuir o volume das caixas de som (hardware!) enquanto ouve uma música tocada pela página *web*. Sendo assim, resta a questão:

**Questão 4.** O designer deve representar na MoLIC as interações do usuário com aplicações externas ao sistema modelado? Por quê? Em que casos ele deveria representar na MoLIC estas interações?

MoLIC é uma ferramenta epistêmica proposta para apoiar o designer no projeto do discurso interativo do sistema sendo projetado. Assim sendo, o diagrama de interação representa apenas a conversa de um único usuário com o preposto do designer cristalizado na interface. Qualquer interação do usuário fora do sistema sendo projetado *não* deve fazer parte do diagrama de interação, simplesmente porque não faz parte da conversa usuário-preposto sendo projetada. No entanto, é claro que a conversa usuário-preposto poderá ser afetada por interações do usuário com sistemas externos, da mesma forma que qualquer conversa é influenciada pelo ambiente em que ocorre.

Apesar disto, existem situações onde a própria conversa usuário-preposto considera um sistema externo como interlocutor da conversa. Por exemplo, quando o usuário interage com uma página *web* para alterar a página inicial (*homepage*) cadastrada no navegador, o navegador *web* é considerado um interlocutor externo à conversa entre o usuário e o preposto cristalizado na página *web*. Outro exemplo seria quando selecionamos um arquivo no Windows Explorer® e o passamos para um programa de *e-mail* criar e enviar um novo *e-mail* com este arquivo em anexo. Nestes casos onde um sistema externo é considerado um interlocutor da conversa usuário-preposto, a interação do usuário com o sistema externo deve fazer parte do projeto de interação sim, e esta conversa com um ator externo deve ser representada conforme definido na Questão 16<sup>4</sup>.

### Processos do sistema

Existem momentos durante a interação onde o próximo tópico da conversa (a próxima cena) é determinado pelo sistema em resultado de um processamento. Estes momentos são indicados no diagrama de interação por um *processo do sistema*, representado graficamente por uma caixa preta (Figura 19). Usou-se a noção de “caixa preta”, pois o usuário não percebe o que está ocorrendo em um processo do sistema, até que o preposto do designer o comunique. Esta forma de representação visa motivar o projeto cuidadoso das falas do preposto do designer emitidas após cada processamento, como *única* forma de comunicar ao usuário o que ocorreu no processamento, como, suas conseqüências, e explicar o por quê. As respostas de um processo do sistema são oportunidades para o designer comunicar a sua visão sobre a interação, ou seja, oportunidades de manifestação da metacomunicação designer-usuário durante a interação do usuário com o sistema (seção 2.2.1).

---

<sup>4</sup> Questão 16 - Como projetar a influência da interação de um usuário sobre a interação de outro usuário (externo ao contexto de interação representado)?

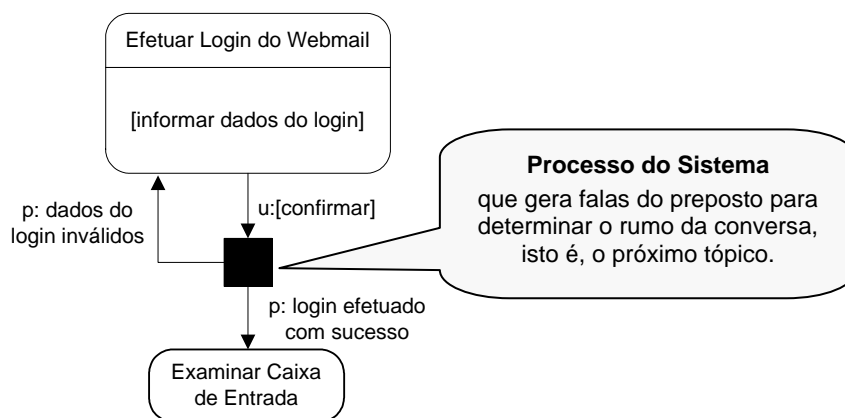


Figura 19: Rumos da conversa em função do processamento do *login*.

A Figura 19 apresenta um exemplo de processo do sistema para efetuar o *login* do usuário em seu *webmail*. Quando ocorre algum erro no processamento do *login*, o preposto do designer emite a fala *p: dados do login inválidos* para informar ao usuário sobre o que aconteceu e voltar para a cena de origem, onde o usuário tem a oportunidade de corrigir os erros e tentar efetuar o *login* novamente. Se, no entanto, o processo do sistema efetuar corretamente o *login*, o preposto do designer emite a fala *p: login efetuado com sucesso* e altera o rumo da conversa para a cena de tópico *Examinar a caixa de entrada*, representada nesta figura em sua forma mínima.

Neste exemplo, o preposto do designer somente comunica ao usuário o resultado do processamento quando o processo do sistema estiver finalizado por completo. Isto quer dizer que, enquanto o sistema estiver processando, o usuário não “ouve” nem “fala” nada antes do preposto responder o resultado final do processamento. Durante o processamento, o usuário apenas está ciente de que o sistema está “pensando”, o que em ambientes gráficos costuma ser representado pelo cursor com uma ampulheta em movimento.

Contudo, esta forma de representação do processo do sistema pode ser imprópria para determinadas situações corriqueiras em sistemas interativos, como, por exemplo, o envio de um e-mail com anexos grandes, uma desfragmentação de disco rígido, ou um processamento mais complexo de imagem. Nestes casos, é conveniente que o usuário tome conhecimento do andamento do processo durante sua execução, seja através de índices (por exemplo, porcentagem de trabalho concluído e quantidade de bytes enviados/copiados) ou de resultados parciais (por exemplo, quais arquivos foram encontrados até o momento).

Além disto, como este tipo de processamento pode ser demorado, em alguns casos o designer pode desejar fornecer ao usuário a oportunidade de desistir de esperar o término do processamento para prosseguir com a interação. Deste modo, será preciso que, durante o processamento, o sistema seja capaz de “ouvir” o usuário manifestar seu novo desejo. Caso um processamento demorado for iniciado por um lapso do usuário, a oportunidade do usuário desistir dele tão logo quanto percebeba seu lapso reduz o custo de recuperação deste mal-entendido na comunicação.

Mais do que interromper o processo do sistema, o usuário pode continuar interagindo com o sistema antes do processo terminar, principalmente se ele utilizar os resultados parciais do processamento. A Figura 20, por exemplo, apresenta os resultados parciais do processo de busca por arquivos do Windows XP® enquanto o restante do disco está sendo investigado. Neste caso, o usuário pode abrir, apagar, mover, copiar, renomear, enfim, interagir com os arquivos encontrados antes do processo de busca terminar.

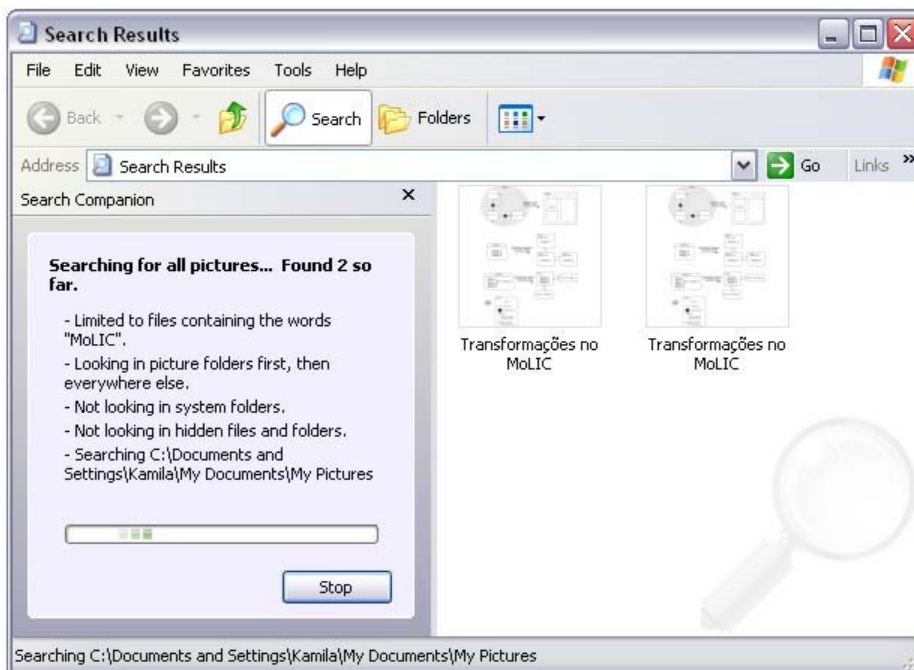


Figura 20: Resultados parciais da pesquisa por arquivos no Windows XP®.

Nem sempre a interação do usuário durante a execução de um processo do sistema atua simplesmente sobre os resultados parciais, ela também pode atuar sobre o próprio processamento para modificar sua forma de execução. No caso de gerenciadores de *downloads*, por exemplo, o usuário pode configurar a forma de

realizar o *download* do arquivo enquanto o arquivo é copiado da Internet. Na Figura 21, por exemplo, vemos uma tela do GetRight® onde o usuário pode escolher quantos fragmentos do arquivo devem ser buscados de uma única vez (item de menu “Download Acelerado (Segmentado)”) e em qual servidor da Internet cada fragmento do arquivo deve ser copiado (itens de menu “Segmento 1”, “Segmento 2” e assim por diante).



Figura 21: Interação do usuário durante o *download* de um arquivo no GetRight®.

Tais lacunas levantam a seguinte questão para a segunda edição da MoLIC:

**Questão 5.** Como representar processos demorados que informem ao usuário sobre seu andamento e que permitam interação do usuário antes de sua conclusão?

A representação gráfica de “caixa preta” (Figura 22a) utilizada para representar um processo do sistema remete à idéia de que o usuário não conhece o que se passa durante o processamento até que o preposto do designer o comunique. Esta comunicação ocorre somente após a conclusão do processo do sistema com a emissão das falas de transição que saem dele.

Em situações onde o usuário precisa estar ciente da evolução do processamento, parte desta “caixa preta” deveria ser apresentada para o usuário.

Desta forma, poderíamos representar uma “caixa branca” ao lado da caixa preta, onde índices da evolução do processo seriam apresentados ao usuário, mas diálogos não poderiam ser travados com o sistema. A única forma de interação do usuário possível numa “caixa branca” seria a emissão de falas de transição que determinem o próximo tópico da conversa. Em geral, as falas de transição que saem de uma “caixa branca” dizem respeito ao cancelamento do processo do sistema, como no exemplo da Figura 22b que cancela a desfragmentação do disco.

Entretanto, em certas situações é desejável que o usuário possa interagir com o sistema antes da conclusão de um processo do sistema para influenciar o próprio processamento ou fazer uso dos resultados parciais. Nestes casos, não basta o preposto do designer informar um índice de evolução do processamento como na Figura 22b, é preciso, na verdade, permitir que o usuário trave diálogos com o sistema durante o processamento.

Assim sendo, podemos dizer que nestes casos a execução de um processo do sistema ocorre associada e simultaneamente à conversa descrita por uma cena, de modo que tanto o processo influencie a conversa quanto o contrário. Esta inter-relação foi representada graficamente por duas setas paralelas que unem o processo do sistema e a cena em questão (Figura 22c). Optamos pelo uso de duas linhas paralelas para fazer alusão a um canal por onde passam falas do preposto comunicando a influência da evolução do processo sobre a cena e falas do usuário comunicando a influência da conversa ocorrida na cena sobre o processo do sistema. Este canal possui linha de largura maior do que as setas para indicar maior intensidade na interinfluência entre um processo do sistema e uma cena. Por fim, a representação de “caixa branca” (Figura 22b) pode ser vista como um caso particular de inter-relação entre um processo do sistema e uma cena (Figura 22c).

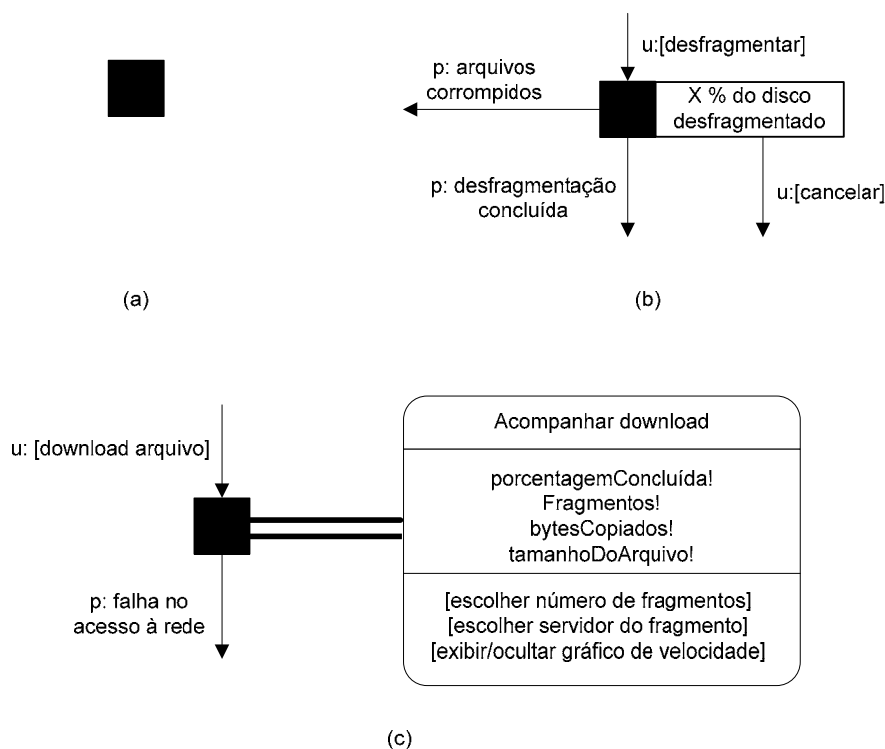


Figura 22: Possíveis representações de processos do sistema.

## Falas de Transição

Como vimos anteriormente, uma cena descreve a conversa do usuário com o preposto sobre um determinado tópico. Em qualquer momento da conversa dentro de uma cena, o usuário pode provocar uma mudança de tópico emitindo uma *fala de transição*. No diagrama de interação, uma fala de transição é representada por uma seta unidirecional preta e um rótulo, ligando cenas, processos do sistema e acessos ubíquos.

Uma fala de transição do usuário geralmente inicia um processo do sistema, mas, para facilitar a leitura do diagrama, é recomendável que o processo do sistema somente seja representado se o rumo da conversa for afetado por ele. Em outras palavras, se o resultado do processamento sempre levar a conversa para apenas para uma cena e o designer decidir não tratar nenhum erro, então não é preciso representar o processo do sistema no diagrama de interação; basta representar a fala de transição do usuário até a cena de destino. Por exemplo, se durante a leitura de *e-mails* o usuário decidir responder a algum, ele, então, pode emitir a fala de transição *u:[responder]* que certamente o levará para outra cena de tópico *Responder e-mail* (Figura 23). Apesar de existir algum processamento

do sistema na transição da cena *Examinar Caixa de Entrada* para a cena *Responder e-mail*, repare que o designer não representou um processo do sistema neste caso, porque existe apenas uma cena de destino e ele optou por não tratar os possíveis erros.

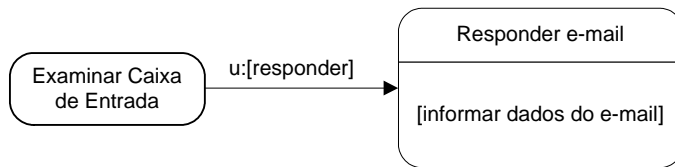


Figura 23: Fala de transição do usuário acessando diretamente outra cena.

Quando, entretanto, uma fala de transição do usuário iniciar um processo do sistema com mais de um possível rumo de conversa, sejam estes rumos de tratamentos de *breakdown* ou não, o processo do sistema deve ser representado no diagrama de interação. Cada possível rumo será representado por uma fala do preposto do designer que sai do processo do sistema em direção à cena de destino correspondente, comunicando ao usuário o resultado do processo (vide exemplo da Figura 19).

Segundo Paula (2003), o rótulo de uma fala de transição é composto por três partes (p. 48 e 49):

1. **pré-condições:** condições que devem ser satisfeitas para que o usuário ou o preposto possa enunciar a fala correspondente. São precedidas pela expressão *pré:*. Normalmente são expressas em linguagem natural. 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.*



A Figura 24 apresenta a forma de representação do rótulo das falas de transição no diagrama de interação. Vale lembrar que as pré e pós-condições podem ser omitidas quando não forem necessárias.

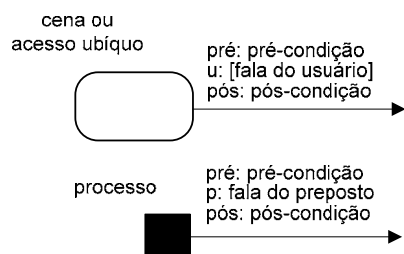


Figura 24: Representação de fala de transição no diagrama de interação (*adaptado de Paula, 2003*).

Na proposta de Paula (2003), estas condições diziam respeito a “o estado atual da aplicação”, e eram escritas em linguagem natural ou indicadas por “signos de controle da interação cujos valores compõem este estado” (p. 49). Um signo de controle exemplificado foi “\_login=F” (lê-se “o usuário não pode estar “logado” no sistema”, p.50) na pré-condição para o usuário efetuar o *login*, e “\_login <- T” (lê-se “signo de controle recebe o valor *true* (...), indicando que houve uma mudança de estado da aplicação”, p. 50) na pós-condição do *login* efetuado. Este uso de pré e pós-condições conduz o designer a pensar em variáveis do sistema como um programador, distanciando-o da metáfora de conversa. Então,

**Questão 6.** De que maneira o uso de pré e pós-condições pode manter o designer no papel de interlocutor da conversa usuário-preposto e não levá-lo a pensar como um programador?

Conforme a estrutura do processo de comunicação discutida na seção *Processos de Significação e Comunicação* e a definição de fala de transição, uma fala de transição transmite uma *mensagem* do *emissor* ao *receptor*, que determina o próximo tópico da conversa. Esta mensagem é representada pela descrição da fala de transição (semelhante a *u:[fala de transição]*, quando o usuário é o emissor, ou a *p:fala de transição*, quando o emissor é o preposto) e se refere ao contexto da conversa, isto é, aos aspectos do ambiente onde a conversa ocorre (e.g., características e cultura dos participantes, e o local e o tempo em que ocorre conversa) e a tudo o que foi dito desde o início da conversa.

Quando conversamos, isto é, emitimos falas e ouvimos alguém falar, é comum utilizarmos dois recursos lingüísticos: a *pressuposição* e a *implicatura* (Brown e Yule, 1983; Barbosa e da Silva, 2001). A pressuposição nos faz considerar como verdade informações do contexto que tornam nossa fala verdadeira antes de pronunciarmos-la. Isto corresponde à nossa predisposição de falar somente a verdade naquele contexto de conversa. Por exemplo, uma secretária de um médico, mesmo com toda a experiência que possa ter, não pode (ou poderia) receitar um medicamento a um paciente antigo do seu patrão por *pressupor* não deter a formação necessária para receitar medicamentos; nem pode (ou poderia) afirmar com absoluta certeza que uma operação de risco terminará em um determinado horário, por *pressupor* que ninguém é capaz de prever com exatidão esse tipo de evento. Já uma implicatura corresponde a considerarmos comunicado algo não dito explicitamente, isto é, algo que pode ser inferido a partir daquilo que falamos. Por exemplo, quando dizemos que está chovendo em uma grande cidade, como Rio de Janeiro e São Paulo, *implica* que o número de veículos em trânsito nesta cidade está maior do que normalmente é, a ponto de piorar o trânsito. Note que esta implicação só foi possível num contexto de uma grande cidade, pois se fosse numa grande fazenda dificilmente o volume de veículos em trânsito aumentaria a ponto de piorar significativamente o trânsito. Vejamos outras definições de pressuposição e implicatura:

Uma pressuposição é uma crença anterior, relacionada a uma fala, que deve ser mutuamente conhecida ou assumida pelo emissor e receptor para que a fala seja considerada apropriada no contexto. (Glossary of linguistic terms da LinguaLinks Library<sup>5</sup>)

Uma implicatura é algo intencionado, implicado, ou sugerido distinto daquilo que é dito. Implicatura pode ser parte do significado de uma sentença ou dependente do

---

<sup>5</sup> “A presupposition is background belief, relating to an [utterance](#), that must be mutually known or assumed by the [speaker](#) and [addressee](#) for the utterance to be considered appropriate in context.” (Glossary of linguistic terms of LinguaLinks Library)

contexto conversacional, e pode ser [culturalmente] convencional ou não convencional. (Stanford Encyclopedia of Philosophy<sup>6</sup>)

Uma implicatura é algo que é inferido a partir de uma fala, mas que não é uma condição para a verdade da fala. (Glossary of linguistic terms da LinguaLinks Library<sup>7</sup>)

Com base nestes recursos lingüísticos, a pré-condição passa a ser considerada pressuposição, a pós-condição passa a ser considerada implicatura, e as “condições” deixam de fazer referência a “o estado atual da aplicação” (Paula, 2003 p. 49) e passam a se referir ao contexto da conversa compartilhado pelos participantes.

Como falamos anteriormente, o contexto da conversa está relacionado com aspectos do ambiente onde a conversa ocorre (e.g., características e cultura dos participantes, e o local e o tempo da conversa) e tudo o que foi dito desde o início da conversa. No âmbito de projeto de IHC, o contexto da conversa diz respeito aos aspectos do ambiente previstos pelo designer e a tudo aquilo que o designer considera que pode vigorar durante a conversa. A MoLIC ainda não possui uma definitiva representação explícita do contexto da conversa. Apesar disso, o designer pode recuperar do diagrama de interação alguns aspectos que compõem o contexto, como, por exemplo, o caminho de interação percorrido pelo usuário desde o início da conversa e a modificação do valor dos signos nas falas de transição (implicaturas). Vale notar que não faz parte da segunda edição do MoLIC fornecer uma representação explícita para o contexto.

\*\*\*

---

<sup>6</sup> “An implicature is something meant, implied, or suggested distinct from what is said. Implicatures can be part of sentence meaning or dependent on conversational context, and can be conventional or unconventional. (Stanford Encyclopedia of Philosophy)”

Stanford Encyclopedia of Philosophy - <http://plato.stanford.edu/entries/implicature>. Última visita em junho de 2005.

<sup>7</sup> “An implicature is anything that is inferred from an [utterance](#), but that is not a condition for the truth of the utterance.” (Glossary of linguistic terms of LinguaLinks Library, op.cit.)

A fala de transição emitida pelo usuário representa seu desejo de concluir a conversa atual ou de mudar o tópico da conversa. Geralmente sua descrição em linguagem natural se assemelha à fala: – Quero “fazer algo” (por exemplo, *u:[confirmar]*, *u:[efetuar login]*, *u:[enviar e-mail]* e etc.). Já a fala de transição emitida pelo preposto manifesta o resultado de um processamento através de uma descrição geralmente semelhante às falas: – Consegui realizar o que me pediu, e o que aconteceu foi “isto”, ou – Não é possível realizar o que me pediu considerando as informações que você me forneceu (por exemplo, *p:usuário cadastrado com sucesso*, *p:dados inválidos*).

Não existe recomendação explícita sobre como descrever as falas de transição. No entanto, percebemos que normalmente a descrição das falas de transição emitidas pelo usuário inicia por um verbo no infinitivo seguido do seu complemento, quando for o caso, e a descrição das falas de transição do preposto geralmente está na voz passiva. Mesmo com esta prática espontânea, podemos encontrar falas de transição que referenciam signos apresentados/modificados na cena de origem, porém suas descrições não indicam quais signos são referenciados, principalmente quando referenciam parte de um signo composto<sup>8</sup>. Por exemplo, a Figura 25a apresenta a fala de transição *u:[visualizar]* indicando que o usuário pode visualizar documentos. Contudo, não está claro se é possível pedir para visualizar apenas um documento ou um conjunto de documentos por vez, nem se a determinação de quais documentos serão visualizados é feita na cena de origem ou de destino. Na Figura 25b, vemos o preposto informar que os dados do cliente estão inválidos através da fala de transição *p:dados inválidos*, mas ele não informa qual ou quais dados estão inválidos.

---

<sup>8</sup> Um signo composto é qualquer agrupamento de signos com algum significado, como, por exemplo, o signo *aluno* é composto pelo signo *nome*, *endereço* e *matrícula*, e o signo *documentos\_encontrados* é composto pelo conjunto de *documentos* resultantes de uma busca, que por sua vez também são signos compostos.

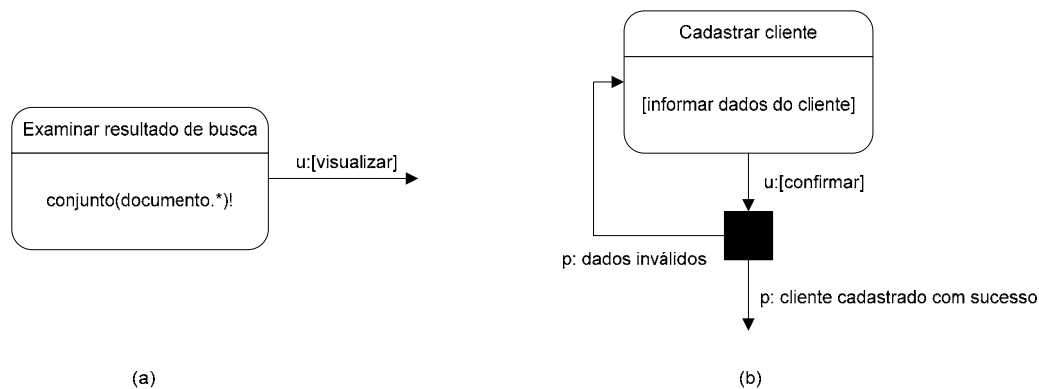


Figura 25: Falas de transição que não indicam os elementos referenciados.

Este problema se torna mais visível quando utilizamos verbos cujos complementos referem-se a signos manipulados na cena de origem. Por exemplo, quando se utiliza *u:[imprimir]* (o quê?) , *u:[remover]* (o quê?) , *u:[visualizar detalhes]* (de quê?) ou *u:[responder]* (a quem?). Logo,

**Questão 7.** Como explicitar os signos envolvidos numa fala de transição que tenham sido apresentados ou modificados na cena de origem da fala?

Nos casos em que uma fala de transição envolver signos da cena de origem, ou seja, quando a fala de transição “levar” signos para um processo do sistema ou a uma próxima cena, a sua descrição deve explicitar os signos envolvidos. Isto já é normalmente feito quando se tratam de signos simples ou signos compostos considerados como uma unidade simples, como, por exemplo, *u:[enviar e-mail]*, *p:e-mail enviado com sucesso*, *u:[cadastrar cliente]* e *p:cliente cadastrado com sucesso*.

Todavia, não se tem tomado o devido cuidado na descrição das falas de transição que referenciam parte de um signo composto ou de um signo que representa um conjunto de signos. Por exemplo, se houver algum problema no cadastramento do cliente, não é suficiente informar ao usuário que houve um problema (*p:dados do cliente inválidos*); é necessário ser mais específico e informar exatamente quais signos estão envolvidos no problema, e por quê. Uma maneira de deixar claro de que signos, partes de signos, ou (sub)conjunto de signos se está falando durante uma transição seria utilizar *templates* de descrição de falas de transição predefinidos.

\*\*\*

Imagine, agora, que o usuário esteja interagindo com o sistema para responder um *e-mail* conforme descrito pela cena de tópico *Responder e-mail* na Figura 26. Numa conversa como esta, existem pelo menos duas falas de transição básicas: uma para confirmar o envio da resposta do *e-mail* (*u:[enviar]*) e outra para desistir do envio<sup>9</sup> (*u:[cancelar]*). Antes de o usuário enviar a resposta do *e-mail*, emitindo a fala de transição *u:[enviar]*, ele é obrigado a travar o diálogo *[informar dados do e-mail]* para fornecer os dados necessários. No entanto, ele não necessita travar diálogo algum para desistir de enviar a resposta do *e-mail*. Ou seja, a fala de transição *u:[cancelar]* não depende de qualquer diálogo da cena de origem.

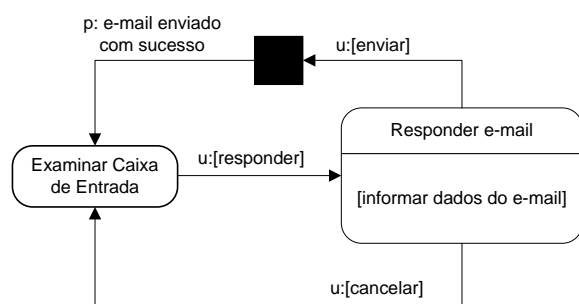


Figura 26: Falas de transição da cena *Responder e-mail*: uma dependente de diálogo (*u:[enviar]*) outra independente (*u:[cancelar]*).

Uma situação um pouco mais complexa do que esta aparece na tela de *login* do Globo Online<sup>10</sup> (Figura 27). No Globo Online existem três formas diferentes de se efetuar o *login*: (1) se usuário for cadastrado no Globo Online, (2) se o usuário for assinante do Globo.com ou (3) cadastrando-se no Globo Online.

<sup>9</sup> Por ser apenas um exemplo, a nossa decisão de design foi *não* representar que o usuário deveria confirmar o cancelamento do envio antes que o sistema efetivamente cancele-o. Contudo, seria perfeitamente possível representar uma confirmação de cancelamento no diagrama de interação MoLIC.

<sup>10</sup> Globo Online - [www.oglobo.com](http://www.oglobo.com). Última visita em junho de 2005.

**GLOBO ONLINE**

**CONTEÚDO GRATUITO PARA CADASTRADOS**

Faça aqui mesmo o seu cadastro, uma única vez, e tenha acesso a todo o conteúdo do Globo Online. É muito simples - e inteiramente grátis.

**JÁ SOU CADASTRADO NO GLOBO ONLINE**

» Por favor, informe e-mail e senha para continuar

Email

Senha

» [Esqueci a senha](#)  
 » [Tire suas dúvidas](#)  
 » [Central do leitor](#)

Quero ter acesso automático ao site nas próximas visitas.

**SOU ASSINANTE GLOBO.COM**

» Por favor, informe usuário e senha para continuar

Usuário  @globo.com

Senha

Quero ter acesso automático ao site nas próximas visitas.

**AINDA NÃO SOU CADASTRADO**

» Caso não seja cadastrado, preencha os campos abaixo para ter acesso a todo o conteúdo do Globo On Line

E-mail:  Confirme o e-mail:

Senha:  Confirme a senha:

Lembrar a senha para ter acesso automático ao site nas próximas visitas

Nome Completo:

Data de Nascimento:  /  /  Sexo:  Masculino  Feminino

Telefone: (  )  De onde acessa:

CEP residencial:  -  [consulte](#) Estado:

Renda mensal:  Escolaridade:

É leitor do Globo impresso?

Figura 27: Tela de login no Globo On-line.

O designer tem pelo menos duas possibilidades para modelar esta conversa sobre *login* utilizando a MoLIC: (a) representar estas três formas de efetuar o *login* em diálogos na mesma cena, por considerar o mesmo tópico (efetuar *login*); ou (b) em cenas distintas, por considerar tópicos diferentes (efetuar *login* no Globo Online, efetuar *login* no Globo.com e cadastrar usuário no Globo Online). Suponha que o designer tenha escolhido trabalhar com um único tópico: *Efetuar Login*. Assim, ele modelaria algo semelhante ao diagrama da Figura 28, com transições que iniciarão diferentes processos do sistema. Note que a fala de transição *u:[confirmar login no Globo Online]* depende do diálogo *[informar dados do login no Globo Online]*, a transição de fala *u:[confirmar login no Globo.com]* depende do diálogo *[informar dados do login no Globo.com]*, a fala de transição *u:[cadastrar]* depende do diálogo *[informar dados do cadastro]*, e, por fim, a fala de transição *u:[cancelar]* independe de qualquer diálogo da cena.

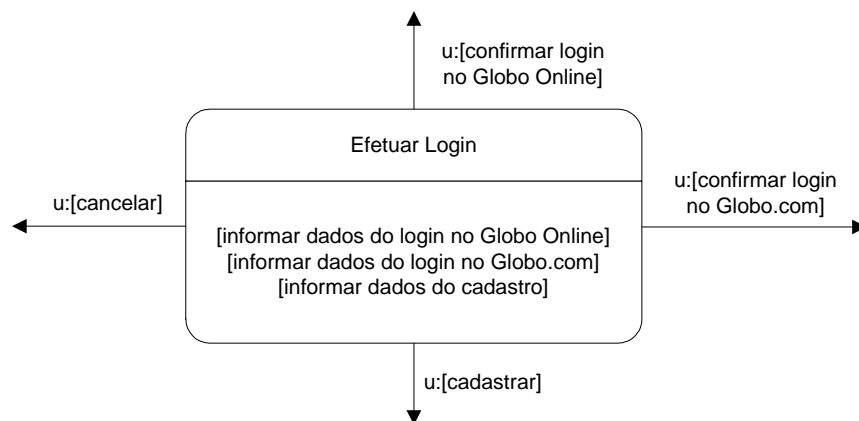


Figura 28: Diagrama de interação para a meta efetuar *login* no Globo Online.

Se voltarmos aos diagramas de interação na Figura 26 e Figura 28, vamos observar que não existe indicação sobre a necessidade de travar diálogos para permitir uma determinada transição. Pensava-se que para resolver este problema bastava indicar que certos signos são obrigatórios dentro da cena. No entanto, como vimos nestes exemplos, não basta informar se um signo é obrigatório (ou opcional) naquela cena, pois ele pode ser obrigatório para uma transição, mas não ser para outra. Desta forma,

**Questão 8.** Como representar a obrigatoriedade de ter sido travado um conjunto de diálogos na cena de origem antes de uma fala de transição ser emitida?

Algumas falas de transição exigem que, antes de serem emitidas, um conjunto de diálogos tenha sido travado na cena de origem. Silva e Barbosa (2004) propuseram representar esta obrigatoriedade indexando numericamente os diálogos requeridos e indicando-os como pressuposição (pré-condição) da fala de transição conforme o exigido (Figura 29a). Para se emitir, por exemplo, a fala de transição *u:[confirmar envio]* (Figura 29b), antes o usuário deve ter travado os diálogos *[texto da mensagem?]* e *[destinatário?]*<sup>11</sup> na cena *Enviar mensagem*.

<sup>11</sup> “?” representa que o usuário é quem determina o valor do signo naquele diálogo.



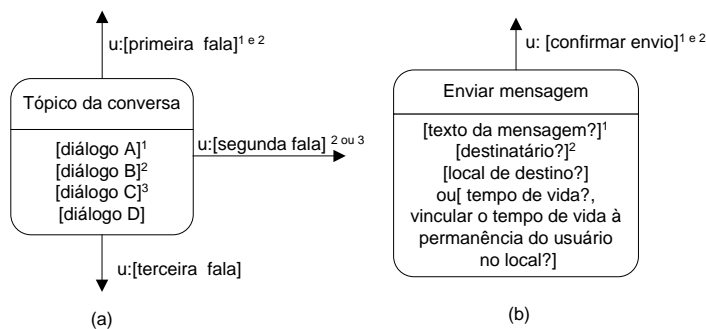


Figura 29: Representação de diálogos obrigatórios para certas transições (*retirado de Silva e Barbosa, 2004 p. 183*).

Esta solução, entretanto, pode dificultar muito a leitura do diagrama de interação, pois o designer precisaria procurar quais diálogos da cena são requeridos pela fala de transição. Uma outra forma de representar este tipo de pressuposição seria agrupar os diálogos conforme a necessidade das falas de transição em um grupo nomeado. Assim, para indicar as pré-condições da fala de transição bastaria referenciar o nome de um grupo e não uma expressão com conjunções e disjunções. A Figura 30 apresenta como ficam agrupados os diálogos do exemplo na Figura 29. Repare que o agrupamento de diálogos pode forçar o designer representar o mesmo diálogo em grupos diferentes (na Figura 30a, por exemplo, o [diálogo B] pertence tanto ao grupo 1 quanto ao grupo 2). Num primeiro momento, esta duplicação parece exigir um esforço maior do designer desnecessariamente. Porém, quando o designer for projetar a interface a partir da interação descrita no diagrama, o reconhecimento rápido de um grupo coeso pode ajudar o designer na decisão de como a interação descrita por uma cena pode ocorrer em diferentes telas de um celular ou PDA (por exemplo, cada grupo em uma tela diferente) ou ainda ajudar na definição do layout de tela (por exemplo, espaços de tela bem definidos para cada grupo associados às respectivas falas de transição).

Por que ao invés de agrupar os diálogos não os consideramos como entidades recursivas, isto é, um diálogo sendo composto por outros? Não optamos por esta forma de representação porque existe um objetivo comunicativo por trás de cada diálogo, normalmente indicado pela sua descrição. No entanto, nem sempre um grupo de diálogos possui um objetivo comunicativo. Além disto, um diálogo é um “cidadão de primeira classe” na MoLIC, mas um grupo de diálogos não.

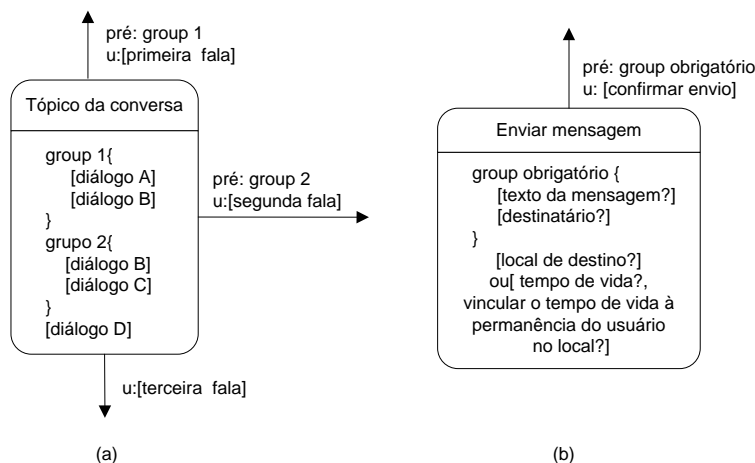


Figura 30: Representação de grupos de diálogos obrigatórios para certas transições (adaptado de Silva e Barbosa, 2004 p.183).

\*\*\*

Todavia, ainda não está claro o significado de travar um diálogo da cena.

**Questão 9.** O que significa travar um diálogo da cena?

Travar um diálogo numa conversa (cena) corresponde à emissão de falas do usuário e do preposto organizadas em pares conversacionais. Durante a conversa, os participantes atribuem e/ou modificam o valor da expressão ou do conteúdo de signos, criam e/ou destroem relações entre signos, e assim por diante. Esta manipulação dos signos pelo usuário envolve um processo de semiose ilimitada, que extrapola o sistema e vai muito além daquilo que o designer pode prever. Mas, para o preposto do designer que está sob o ponto de vista do sistema, travar um diálogo significa “propor”, “ouvir” e “responder a mudanças de” valores dos signos. Deste modo, podemos considerar que um diálogo tenha sido travado quando todos os signos imprescindíveis para a conclusão da conversa (chamados de signos obrigatórios) tenham recebido algum valor para o seu conteúdo, seja definido pelo usuário ou simplesmente aceito o valor padrão proposto pelo designer.

\*\*\*

Uma fala de transição também pode depender de uma outra transição, normalmente uma outra fala de reparo de breakdown. Por exemplo, o designer do

site Submarino<sup>12</sup> decidiu que somente quando a conta de um cliente fosse bloqueada por uma fala de transição de *breakdown*, o usuário poderia fazer a transição para uma cena de desbloqueio da conta (Figura 31).



Figura 31: Tela de *login* do Submarino.com quando a conta está bloqueada.

**Questão 10.** Como representar que uma fala de transição que sai da cena depende de outra fala de transição que chegou anteriormente na mesma cena?

Apesar desta necessidade ainda não ter aparecido em estudos de casos anteriores, é perfeitamente possível representar esta forma de interação com a proposta original da MoLIC. Basta representarmos como pressuposição (pré-condição) de uma fala de transição as condições de contexto determinadas pela (implicatura de) outra fala de transição que chegou anteriormente na cena. Por exemplo, voltemos ao caso do Submarino.com (Figura 31). Ali o usuário só podia emitir uma fala de transição para desbloquear sua conta quando ela já havia sido bloqueada por outra fala de transição que chegou anteriormente à cena. A Figura 32 apresenta uma forma de representar este exemplo do Submarino.com no diagrama de interação. Repare que todos os elementos utilizados no diagrama de interação foram definidos na proposta original da MoLIC.

<sup>12</sup> Submarino – [www.submarino.com.br](http://www.submarino.com.br). Última visita em junho de 2005.

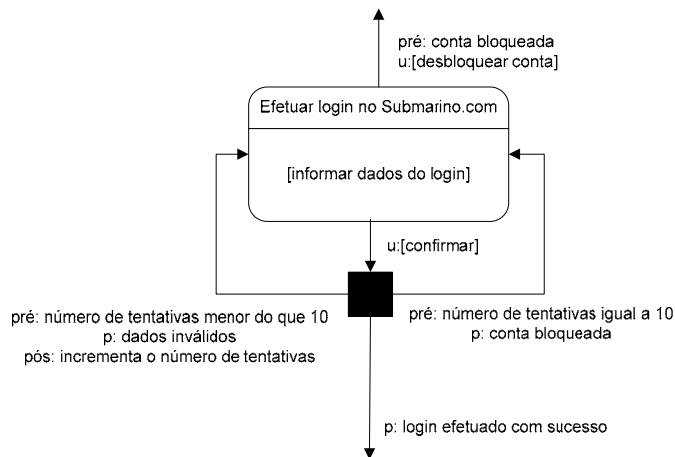


Figura 32: Fala de transição dependendo de outra fala de transição anterior.

\*\*\*

Nos trabalhos em que Paula apresentou a MoLIC (Paula, 2003 e Barbosa e Paula, 2003), o único momento em que ela aborda a diferença entre uma mensagem do designer (cena especial) e uma fala de transição emitida por ele foi num exemplo que compara duas soluções de interação (Paula, 2003 p. 52 e 53). Ela comparou uma solução utilizando uma mensagem do designer com outra utilizando uma fala de transição em resposta a um processo do sistema que não encontrou avisos procurados. Veja a comparação feita por Paula neste caso:

Quando nenhum aviso é encontrado em uma busca, na solução inicial o usuário receberia uma mensagem [do designer] indicando o insucesso da busca, e então seria levado novamente à cena Buscar avisos, para tentar realizar nova busca [Figura 33a]. 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 [utilizando uma fala de transição emitida pelo preposto] [Figura 33b]. 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. (Paula, 2003 p. 52)

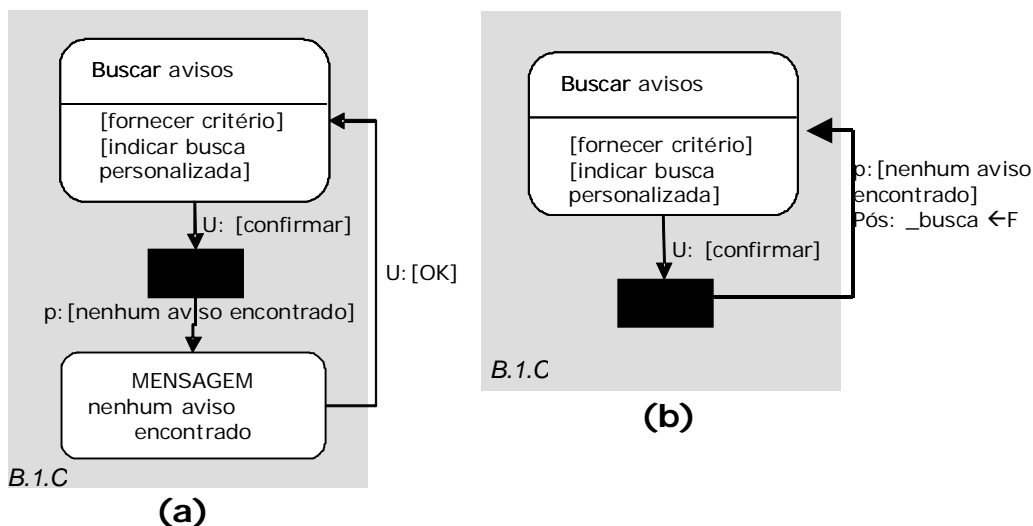


Figura 33: Soluções alternativas de interação para indicar resultado negativo de busca (nenhum aviso encontrado) (retirado de Paula, 2003 p. 53).

Com esta comparação feita por Paula, podemos concluir que uma mensagem do preposto apresentada em uma cena tende a ter um custo de interação maior do que uma fala de transição: o usuário precisa emitir mais uma fala para atingir o mesmo objetivo. Ela sugeriu que em ambientes *Web* fossem utilizadas falas de transição por possuírem menor custo de interação (“razão custo/clique”). Isto quer dizer que sempre que o designer puder, ele optará pelo menor custo de interação? Se não,

**Questão 11.** Quando e com que intenção de comunicação o designer deve utilizar uma cena especial (mensagem do designer em resposta a um processamento) ou uma fala de transição emitida pelo preposto do designer?

Existem pelo menos duas questões a serem consideradas na escolha entre representar uma fala de transição e representar uma cena que apresenta uma mensagem do designer: a questão do custo de interação e a questão da ênfase dada àquilo que se quer comunicar. O custo de interação está ligado ao número de falas que o usuário precisa emitir e ouvir para atingir a meta desejada. Quando a interface for construída, este custo de interação será medido em função de “cliques”, “comandos” ou qualquer forma disponível para o usuário expressar suas falas. Obviamente é desejável que o sistema tenha um baixo custo de interação. Todavia, a busca por um baixo custo de interação não pode ignorar a necessidade de se enfatizar algumas mensagens do designer. Por exemplo, quando

um editor de texto tenta abrir um arquivo corrompido, não é conveniente comunicar o ocorrido com uma mensagem que fica exibida por poucos segundos e depois desaparece sem a intervenção do usuário. Neste caso é mais conveniente apresentar uma mensagem para o usuário e fazer com que ele indique quando conseguiu perceber e interpretar a mensagem para, então, continuar a interação. Isto não garante que o usuário vai perceber (“ler”) a mensagem, nem que ele vai compreendê-la, é apenas um salvaguarda do designer contra possíveis mal-entendidos do usuário. Seria algo semelhante ao designer dizer: – “Eu acho importante que você entenda X. Entendeu?”.

Desta forma, podemos considerar uma fala de transição emitida pelo proposto como uma mensagem com menor ênfase, que não exige do usuário uma outra fala em resposta para continuar a interação. Manifestações comuns em interfaces de falas de transição emitidas pelo proposto seriam, por exemplo, um *beep*, uma mudança de ícone, uma janela que se fecha, uma janela que se abre, uma mudança no conteúdo do navegador *web* ou um arquivo que aparece num diretório depois de ter sido copiado para lá.

Portanto, se o designer desejar comunicar uma mensagem ao usuário sem ser enfático e sem exigir uma fala do usuário confirmando que o ouviu dizer alguma coisa, ele deve expressar esta mensagem através de uma fala de transição. De outra forma, caso o designer deseje ser enfático e requerer de uma confirmação do usuário de que pelo menos foi avisado de algo, ele deve utilizar uma cena para apresentar sua mensagem. Assim, cabe ao designer equilibrar o custo de interação e a ênfase na comunicação de acordo com o sistema sendo projetado e com sua intenção de comunicação.

\*\*\*

Durante a realização de estudos de caso, nos deparamos com várias situações em que tivemos dúvidas se representávamos uma fala de transição ou um diálogo dentro da cena. Por exemplo, considere que desejamos representar a possibilidade do usuário mudar de página no Adobe Reader®. No lado esquerdo da Figura 34, representamos esta possibilidade utilizando um diálogo onde o usuário pode informar a página que deseja visualizar. Já no lado direito, representamos a mudança de página através das falas de transição *u:[próxima*

*página*] e *u:[página anterior]*. Aparentemente estas duas soluções estão corretas e representam tudo o que o designer queria dizer sobre a mudança de página.

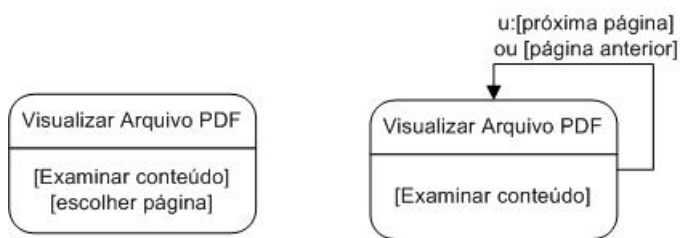


Figura 34: Diferentes soluções de interação para escolha de página no Adobe Reader ®.

Porém, ainda existem algumas questões a respeito:

**Questão 12.** Existe diferença entre diálogos e falas de transição que voltam diretamente (sem intermediários) para a cena de origem? Quando deveríamos utilizar uma ou outra solução? Será que poderíamos utilizar falas de transição que voltam diretamente à cena de origem para representar ações do usuário como *undo*, *redo*, *reset dialog (form)* e etc.?

Uma fala de transição tem o objetivo de mudar o tópico da conversa, ou seja, sair de uma cena e chegar em outra cena, diferente da origem. Todavia, uma fala de transição do usuário que sai de uma cena pode ser interceptada por um processo do sistema. Neste caso, o rumo da conversa é determinado pela fala de transição emitida pelo preposto em resposta ao processamento. Se durante o processo do sistema ocorrer uma ruptura de comunicação, o preposto pode decidir voltar à cena de origem para permitir que o usuário se recupere da ruptura. Assim sendo, não se deve utilizar uma fala de transição com origem e destino na mesma cena, pois esta representação não condiz com a semântica de mudança de tópico, nem de recuperação de ruptura na comunicação.

Um diálogo tem por objetivo apresentar/modificar signos, que podem ser objetos, relações entre objetos ou ações sobre objetos. Considere, por exemplo, um editor de texto. O objeto texto é tradicionalmente modificado pelos diálogos *[adicionar texto]*, *[remover texto]* e *[mover texto]*, e a última ação feita/desfeita sobre o objeto texto também é tradicionalmente modificada pelos diálogos *[desfazer última ação]* e *[refazer última ação]*. Deste modo, um diálogo é capaz de mudar o contexto da conversa e influenciar outros signos da cena, mas não mudam o tópico da conversa.

Portanto, uma fala de transição deve ser usada para representar a mudança de tópico da conversa, e diálogos devem ser utilizados para a apresentação/modificação de signos da cena, mesmo com mudança no contexto.

\*\*\*

Como é natural e freqüente que mal-entendidos ocorram em uma conversa, a engenharia semiótica ressalta a importância do preposto não só comunicar ao usuário que houve um mal-entendido, mas também projetar uma maneira de apoiá-lo na recuperação deste mal-entendido, isto é, projetar uma forma do usuário retomar a conversa em direção ao seu objetivo.

Em certas situações, a mudança de rumo na conversa pode corresponder ao esforço de reparo de uma ruptura na comunicação, também chamada de *breakdown*. Sendo assim, para cada signo ou diálogo em que o designer preveja uma possibilidade de ruptura na comunicação, ele deve especificar as falas de transição de *breakdown* associadas, seguidas pelo apoio ao usuário na recuperação deste mal-entendido.

Voltemos ao exemplo do *login* no *webmail* (Figura 35). Neste caso, além do rumo da conversa onde o *login* é efetuado com sucesso, o designer também previu um possível *breakdown* quando os dados do *login* forem inválidos. Estes possíveis rumos da conversa estão representados no diagrama de interação como falas de transição emitidas pelo preposto em resultado do processamento. No caso do *breakdown*, o apoio à recuperação foi voltar à cena de origem onde o usuário pode tentar efetuar o *login* novamente, depois de consertar o provável erro.

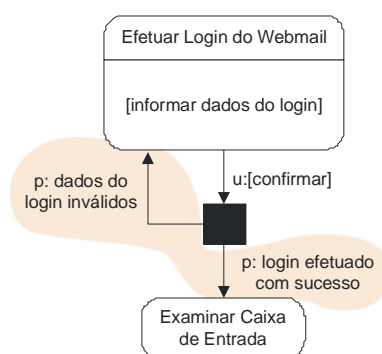


Figura 35: Falas de transição em resposta ao processamento do *login*.

Atualmente, o designer somente consegue diferenciar falas de transição normais, em que tudo ocorre como esperado, daquelas que representam



*breakdown*, lendo suas descrições em linguagem natural (Por exemplo, *p: dados do login inválido*). Considerando que a MoLIC privilegia a reflexão do designer sob uma visão global da interação,

**Questão 13.** De que maneira o designer poderia diferenciar visualmente as falas de transição normais do preposto daquelas que são uma forma reparo de *breakdown*, de modo que a representação possa chamar sua atenção tanto para o tratamento que ela inicia quanto para sua (in)existência?

Considerando a importância para a engenharia semiótica do tratamento de rupturas de comunicação, acreditamos ser vantajoso trazer para a representação gráfica a diferenciação das falas normais (sem reparo de *breakdown*) daquelas de reparo de *breakdown*. Deste modo, dado que o designer conheça as notações do diagrama de interação, uma apreciação visual rápida do digrama lhe permitirá saber se existem e quais são os caminhos de interação fornecidos pelo designer para a recuperação de mal-entendidos na conversa.

Uma fala de transição sem reparo de *breakdown* continua, então, sendo representada por uma seta unidirecional preta de linha cheia. Já a fala de transição de reparo de *breakdown* passa a ser representada por uma seta unidirecional preta de linha tracejada.

A Figura 36 apresenta o mesmo exemplo da Figura 32 diferenciando a representação gráfica de falas de transição normal daquelas de reparo de *breakdown*. Repare que numa apreciação visual rápida já é possível diferenciar as falas de transição de reparo de *breakdown* daquelas que não são, mesmo sem ler suas descrições. Por um lado, ver uma fala de transição de reparo de *breakdown* chama a atenção do designer para a recuperação do mal-entendido que ela inicia (voltar à cena de origem – Efetuar login do Submarino.com), e por outro, não ver uma fala de transição de reparo de *breakdown* o estimula a verificar se realmente algum mal-entendido não pode ocorrer naquele determinado ponto da interação.

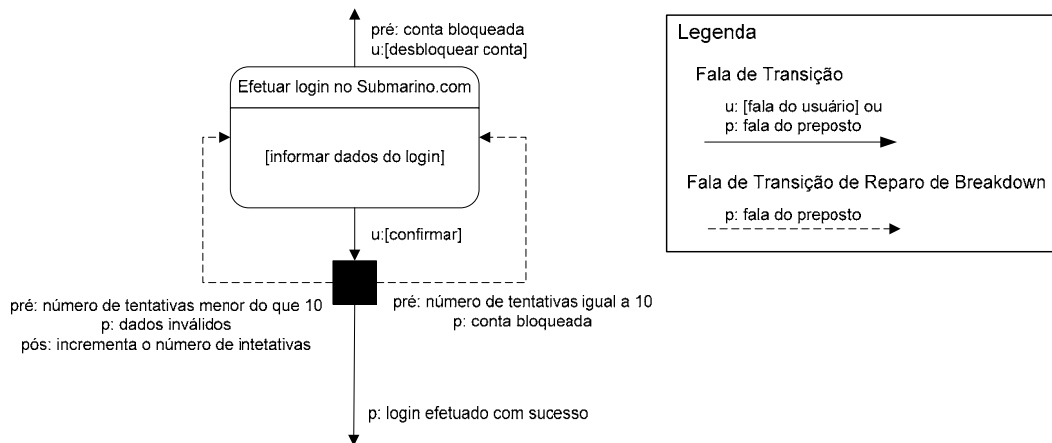


Figura 36: Diferenciação das falas de transição normal daquelas de reparo de *breakdown* emitidas pelo preposto.

\*\*\*

### Acesso Ubíquo

“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” (Paula, 2003 p. 49). A este ponto de onde o usuário pode sair e voltar de qualquer outro lugar e em qualquer momento da interação, chamamos de acesso ubíquo. Um acesso ubíquo é representado no diagrama de interação por um retângulo cinza com bordas arredondadas identificado pela letra U de “ubíquo” mais um número. Num *webmail*, por exemplo, o acesso à cena *Escrever e-mail* normalmente é feito por um acesso ubíquo, pois de qualquer ponto da aplicação é possível escrever e enviar um *e-mail* (Figura 37).

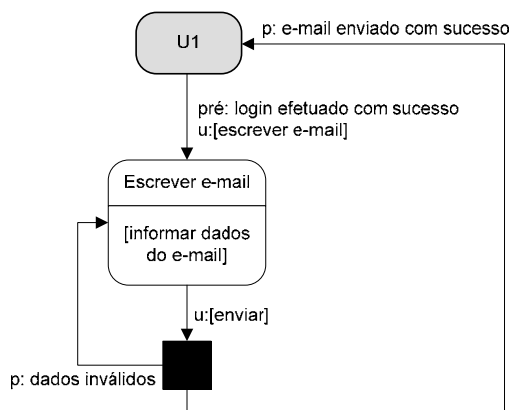


Figura 37: Acesso ubíquo à cena *Escrever e-mail* em um *webmail*.

Assim, uma cena ubíqua é considerada um ponto de acesso ubíquo de onde podem sair (por exemplo, *u:[escrever e-mail]*) e para onde podem voltar (por exemplo, *p:e-mail enviado com sucesso*) falas de transição de qualquer cena do modelo de interação. No entanto, o designer ainda pode ter algumas dúvidas:

**Questão 14.** Exatamente de onde os acessos ubíquos podem ser alcançados? Realmente de todas as cenas do diagrama de interação, ou de algumas cenas específicas? Como representar quais cenas efetivamente podem ser acessadas a partir de um acesso ubíquo?

Na etapa de projeto de IHC em que estamos apenas preocupados com a interação, é suficiente considerar que um acesso ubíquo pode ser alcançado a partir de qualquer cena, contanto que as condições do contexto da conversa para as respectivas falas de transição (i.e. os pressupostos) sejam obedecidas. No momento em que estivermos projetando a interface que possibilite a ocorrência da interação projetada, aí sim o local da interface onde os acessos ubíquos podem se dar passa a ser importante. Portanto, somente no projeto de interface devemos especificar os locais onde os acessos ubíquos realmente estarão disponíveis. Logo, os questionamentos da Questão 14 estão fora do escopo deste trabalho, mas devem ser retomados quando se pensar no projeto de interface a partir da interação projetada na MoLIC.

\*\*\*

Outro problema referente a pontos de acesso no diagrama de interação se manifesta quando o lemos: não existe nenhuma indicação por onde começar nem onde terminar. Disto segue outra questão:

**Questão 15.** Como representar no diagrama de interação onde começa e termina a interação do usuário com o preposto?

Uma solução simples é utilizar as propostas de Statecharts (Harel, 1987) para representar o estado inicial e um estado final. A única diferença é que em MoLIC não existe o conceito de estado da aplicação (e sim estado e contexto da conversa). Poderíamos, então, considerá-los pontos (ou momentos) em que a conversa (interação) se inicia (ponto de entrada) e termina (ponto de saída) (Figura 38).



Figura 38: Representação do começo e fim da interação.

\*\*\*

MoLIC foi proposta para modelar a interação de um único usuário com o seu sistema (o preposto do designer). No entanto, em aplicações multi-usuário, alguns diálogos travados durante a interação usuário-sistema têm como interlocutor, direto ou indireto, um ator externo ao contexto imediato de interação (Silva e Barbosa, 2004). Quando um usuário interage (“ouve” e “fala”), consciente ou inconscientemente, com outro usuário através do sistema sendo projetado,

**Questão 16.** Como projetar a influência da interação de um usuário sobre a interação de outro usuário (externo ao contexto de interação representado)?

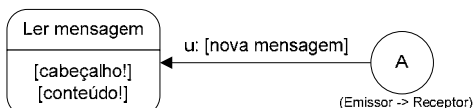
Para projetar a influência da interação de um usuário sobre a interação de outro usuário (externo ao contexto de interação representado), Silva e Barbosa (2004) propuseram a elaboração de um diagrama de interação MoLIC para cada ator ou papel de usuário envolvido na interação (considerando, por exemplo, um modelo de usuário proposto por (Prates, 1998)). Em cada diagrama será projetada a interação usuário-sistema sob o respectivo ponto de vista, levando o designer a refletir sobre a interação de cada (papel de) usuário com o sistema separadamente. Por exemplo, no projeto de interação de um sistema de troca mensagens instantâneas como o NiTA<sup>13</sup>, foram construídos dois diagramas de interação MoLIC: um para o usuário no papel de emissor de mensagens e outro para o papel de receptor.

Depois de construir os diferentes diagramas de interação MoLIC para cada usuário, o designer deve utilizar *pontos de contato* entre os diagramas para relacioná-los. Um ponto de contato representa um canal que transmite, direta ou indiretamente, a influência da interação de um usuário (ou do seu preposto) para a

---

<sup>13</sup> NiTA é um acrônimo para Notes in The Air, um sistema de comunicação baseado em localização. Para maiores informações consulte (Gonçalves et al., 2004).

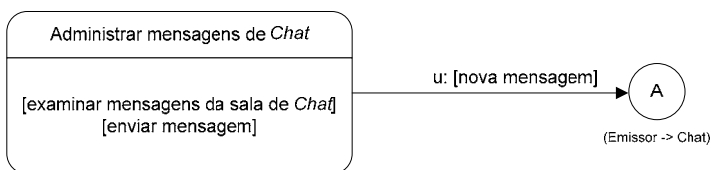
interação de outro(s) usuário(s) externo(s) ao contexto imediato de interação, e vice-versa. Escolheu-se representar graficamente um ponto de contato por um círculo com um rótulo de onde chegam e saem falas trocadas entre o usuário e o(s) ator(es) externo(s). A Figura 39, por exemplo, apresenta o diagrama de interação MoLIC na perspectiva do receptor de mensagens do NiTA. A fala *u:[nova mensagem]* representa a influência da interação emissor-NiTA (usuário-sistema) sobre a interação receptor-NiTA nas respectivas instâncias do sistema.



MoLIC do receptor da mensagem no NiTA

Figura 39: Exemplo de ponto de acesso (*adaptado de Silva e Barbosa, 2004 p. 183*)

Repare que, tanto na Figura 39 quanto na proposta de Silva e Barbosa (2004), a influência da interação de um ator externo sobre a interação do usuário foi representada da mesma forma que uma mudança de tópico na MoLIC. Não convém dois conceitos diferentes serem representados graficamente da mesma forma. Para percebermos o problema mais facilmente, vejamos como fica a compreensão do diagrama de interação de uma sala de *Chat* apresentado na Figura 40. Sempre que o usuário envia uma mensagem para a sala de *Chat*, a fala *u:[nova mensagem]* é emitida. Onde a interação continua depois disto? Ela continua na mesma cena ou existe um próximo tópico da conversa não representado?



MoLIC do emissor da mensagem no Chat

Figura 40: Diagrama de interação para a administração de mensagens num *Chat*.

Para evitar possíveis erros de interpretação do diagrama de interação, propomos outra representação gráfica para a influência da interação do usuário sobre um ator externo (ou vice-versa). A influência da interação também será representada por uma seta preta com um rótulo, mas terá na origem um círculo vazado e no destino uma seta dupla vazada, conforme apresentado na Figura 41.

Caso a influência da interação seja causada por um *breakdown*, a linha da seta será tracejada como na fala de transição de reparo de *breakdown*.

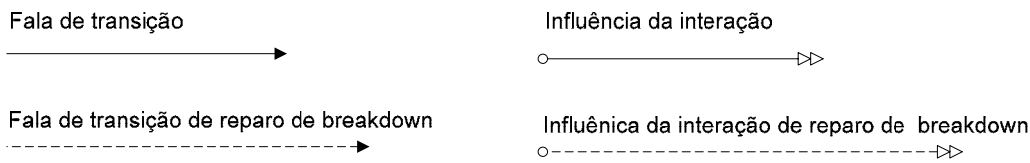


Figura 41: Diferenciação de falas de transição e da influência da interação.

A diferenciação da fala de transição e da influência da interação do usuário sobre um ator externo resolve o problema da administração de mensagens do Chat (Figura 40), conforme apresentado na Figura 42. Na representação gráfica da Figura 42 não existe mais a dúvida em relação à mudança de tópico da conversa. A interação descrita pela cena *Administrar mensagens de Chat* pode influenciar a sala de Chat sem, no entanto, mudar o tópico da conversa.

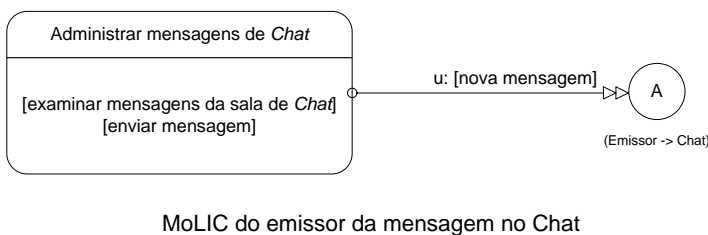


Figura 42: Influência da interação com um Chat sobre um ator externo.

Quando for necessário combinar a transição entre cenas e a influência da interação sobre um ator externo, a fala de transição chega em uma barra preta que se bifurca em uma fala de transição e uma influência da interação (semelhante à transição *fork* do diagrama de atividades na UML). Por exemplo, quando um processo do sistema de um programa de e-mail valida os dados do e-mail antes enviá-lo (Figura 43), a resposta do preposto se bifurca para influenciar a interação de um ator externo (o receptor) e também efetuar a transição para a cena *Examinar caixa de entrada*, através de fala das respectivas falas do preposto *p:novo e-mail* e *p:e-mail enviado com sucesso*. Se durante o processo do sistema que valida o e-mail for encontrado algum signo não obrigatório sem valor definido, mas que na grande maioria das vezes possui algum valor, o usuário necessita confirmar o envio da mensagem. A confirmação do usuário neste caso se bifurca na transição para a cena *Examinar caixa de entrada* e na influência da interação sobre um ator externo, semelhante ao que ocorre em condições normais.

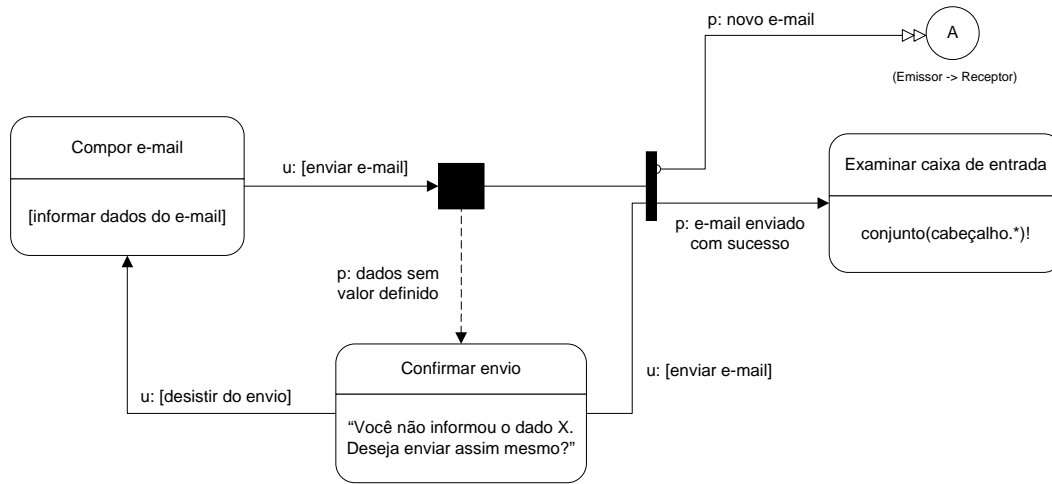


Figura 43: Influência da interação conjugada com transição entre cenas.

Em certos casos, durante a interação o usuário pode “falar” com algum ator externo através do sistema, sem, no entanto, “ouvi-lo”. No Microsoft Word®, por exemplo, o usuário pode enviar o conteúdo do documento por e-mail sem esperar resposta no próprio Word. Quando o usuário apenas “falar” com algum ator externo, não fará sentido o designer utilizar um ponto de contato comum (com outro diagrama de interação MoLIC) (Figura 44a) para representar esta comunicação, e, além disto, as conseqüências desta comunicação estão fora do escopo do sistema sendo projetado. Até porque não existirá outro diagrama de interação MoLIC para ser relacionado por um ponto de contato com outro diagrama de interação. Nestas situações, é mais apropriado o designer utilizar um ponto de contato com um sistema externo, representado graficamente por um círculo metade branco e metade preto (Figura 44b). O lado branco do círculo representa que o usuário, o preposto e o designer conhecem (algo sobre) a entrada do canal de comunicação e aquilo que é transmitido por ele. Já o lado preto do círculo representa o desconhecimento de (exatamente) como/onde é saída deste canal de comunicação, e principalmente das conseqüências das falas recebidas através dele.



(P<sub>1</sub> → P<sub>2</sub>)

(a) ponto de contato com outro modelo de interação



(P<sub>1</sub> → P<sub>2</sub>)

(b) ponto de contato com outro sistema

Figura 44: Representação de ponto de contato (a) com outro modelo de interação e (b) com outro sistema.

### 3.2 Descrição Textual da Interação

Para complementar o diagrama de interação, Paula (Paula, 2003) propôs uma representação textual da interação que define os detalhes da conversa, isto é, que define os signos envolvidos nos diálogos de cada cena e informações adicionais sobre eles. Estas informações adicionais podem incluir:

se o usuário deve definir o valor do signo, se o sistema deve prover um valor padrão para o signo (e como este valor é calculado), o elemento de interface abstrato associado ao signo ([rótulo,] escolha simples, [escolha múltipla,] texto livre, etc.), [o formato esperado,] o grau de conhecimento que o usuário tem sobre o signo, informação adicional para ser comunicada aos usuários com o objetivo de auxiliá-los a compreender e/ou manipular o signo, e algumas anotações adicionais que o designer pode desejar representar. (Barbosa et al. 2004 p.105)

O objetivo principal deste detalhamento é servir de insumo para a construção dos *storyboards* (ou do modelo de interface) correspondentes a cada cena. Por exemplo, imagine uma cena onde o usuário informa os dados para cadastrar um novo cliente (Figura 45a). A definição do diálogo [*informar dados do cliente*] não é insumo suficiente para a construção do(s) storyboard(s) correspondente(s), pois ainda falta definir quais são os signos apresentados/modificados por este diálogo. A Figura 45b apresenta a descrição textual da cena *Cadastrar cliente* em complemento ao diagrama de interação (Figura 45a). Nesta descrição textual são definidos os signos *nome*, *data de nascimento* e *CPF* do cliente que compõem as falas do diálogo [*informar dados do cliente*], e as respectivas informações adicionais. Note que a descrição textual fornece insumos para o projeto e construção da interface quando informa, por exemplo, que o signo *nome* é um texto livre e obrigatoriamente definido pelo usuário, que o signo *data de nascimento* deve estar em um determinado formato e ser selecionado pelo usuário, e que o signo *CPF* é fornecido pelo usuário como um texto livre num formato predefinido.

Como representado na Figura 45 (a) e (b), normalmente a descrição textual é realizada separadamente do diagrama de interação para manter a legibilidade do diagrama, e, por consequência, continuar favorecendo a reflexão do designer sobre uma visão global da interação (vide seções 2.2.2 e 2.2.3). Apesar de ser possível representar parte do detalhamento do diálogo dentro do diagrama de



interação como apresentado na Figura 45c, é notório o quanto esta representação híbrida polui a visão geral da interação.

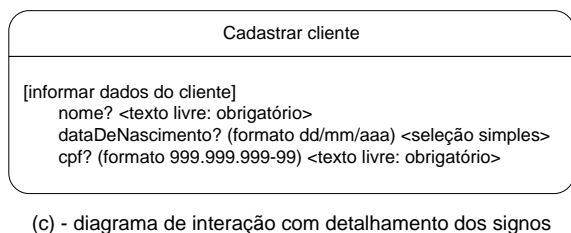
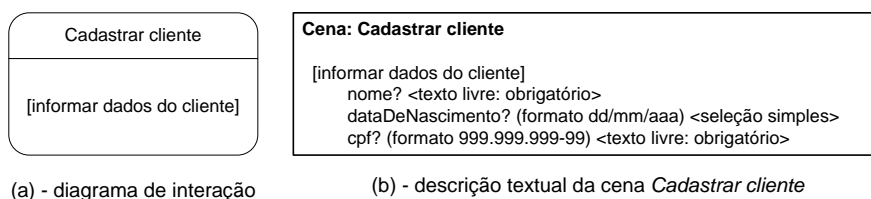


Figura 45: Exemplo de descrição textual da cena *Cadastrar cliente*.

Ainda falta investigar ...

**Questão 17.** Como estas informações adicionais referentes ao signo se relacionam com suas dimensões de conteúdo, expressão e intenção de comunicação?

Compreender as dimensões de conteúdo, de expressão e de intenção de comunicação do signo permite que o designer possa projetar uma forma eficiente de comunicação e, além disto, fornece questões importantes para serem avaliadas nessa comunicação. Por exemplo, quando uma ruptura de comunicação for encontrada na avaliação de comunicabilidade (Prates et al., 2000), o avaliador poderia se questionar sobre quais dimensões dos signos envolvidos afetaram a comunicabilidade e, uma vez identificadas as dimensões problemáticas do signo, ele sugeriria possíveis soluções apropriadas para as respectivas dimensões.

No caso do projeto de interface, que pode ser realizado utilizando-se *storyboards*, é importante que o designer tenha bem clara a intenção de comunicação que envolve o signo, isto é, o seu significado e os efeitos que se pretende causar no ouvinte com a comunicação sobre ele. É importante notar que os efeitos pretendidos dependem de quem determina o valor do signo, ou seja, dependem de quem expressa o conteúdo do signo em algum código disponível na interface. Se for o designer quem determina o valor do signo, sua intenção de comunicação é fazer com que o usuário compreenda não somente o significado do

signo, mas também como o usuário pode ou não usá-lo naquele sistema (de significação) – por exemplo, as possíveis operações sobre o signo e o relacionamento entre signos. Se for o usuário quem determina o valor do signo, a intenção de comunicação passa a ser comunicar ao preposto aquilo que o usuário deseja que o sistema faça ou considere como verdade.

Ao contrário do que normalmente se pensa e/ou pratica, as possíveis formas de expressão do signo dependem não somente do seu conteúdo e do momento da interação em que ocorre, mas também da intenção de comunicação do seu enunciador (quem determina o valor do signo). Assim, tendo em mente a intenção de comunicação sobre o signo, o designer está melhor preparado para definir *restrições* sobre a(s) forma(s) de expressão mais apropriada(s) ao seu conteúdo nos diferentes momento da interação.

Neste trabalho, o tratamento da intenção de comunicação sobre o signo está limitado à identidade do signo, isto é, ao seu significado convencionado pela cultura ou pela tecnologia, independente do momento de interação utilizado. A Tabela 1 organiza os atributos do signo em função de sua identidade, conteúdo e expressão.

| <b>DIMENSÃO</b>   | <b>ATRIBUTO</b>   |
|-------------------|---|
| <b>Identidade</b> | <p><b>Identificador (id)</b><br/>O nome que identifica o signo.</p>   |
|                   | <p><b>Descrição</b><br/>Descreve o significado do signo. (Pode conter “informação adicional para ser comunicada aos usuários para auxiliá-los compreender e/ou manipular o signo” (Barbosa et al., 2004 p.4))</p>   |
|                   | <p><b>Origem</b><br/>Indica se o signo pertence ao domínio do problema (signo de domínio) ou se foi criado (signo de aplicação) ou modificado por causa da tecnologia (signo transformado). O emprego de signos de aplicação exige um cuidado especial do designer para comunicar ao usuário o seu significado, pois seu significado não está (ainda) estabelecido na cultura do usuário.</p> |

|                  |   |
|------------------|---|
| <b>Conteúdo</b>  | <b>Tipo</b><br>O tipo do conteúdo caracteriza os valores que o signo pode assumir.  |
|                  | <b>Domínio</b><br>Conjunto dos valores possíveis que o signo pode assumir, ou restrições sobre esses valores.   |
|                  | <b>Valor default</b><br>O valor default é o primeiro valor que o signo assume quando é definido (ou apresentado na interface). Ele é a sugestão do preposto que o usuário pode aceitar ou não.  |
| <b>Expressão</b> | <b>Elemento de interface (<i>widget</i>) recomendado</b><br>Indica qual é o <i>widget</i> recomendado pelo designer para a apresentação/modificação do signo. O <i>widget</i> recomendado normalmente é abstrato (por exemplo, texto livre, escolha simples, escolha múltipla, rótulo), isto é, independente de implementação e de plataforma, dado que ainda não é recomendado pensar em detalhes de interface. No entanto, vale ressaltar que a implementação da forma de expressão do signo pode fazer uso de outro <i>widget</i> equivalente (por exemplo, na hora de projetar ou mesmo de implementar a interface, o designer pode decidir trocar um <i>widget</i> de escolha múltipla por outro de escolha simples em função do espaço disponível). |
|                  | <b>Formato esperado</b><br>O formato esperado é a sugestão do designer para a restrição da expressão do signo na interface.   |

Tabela 1: Atributos do signo em função das suas dimensões de identidade, conteúdo e expressão.

Além destes atributos, ainda é necessário definir a obrigatoriedade do signo, ou seja, definir se é necessário atribuir algum valor ao conteúdo do signo, mesmo que seja aceitar o valor default. A obrigatoriedade, que também pode ser entendida como uma restrição ao conteúdo do signo (o conteúdo não pode ser vazio), não foi representada na tabela acima porque nem sempre é uma característica intrínseca do signo, mas sim do uso que fazemos dele. Por exemplo, apesar de sabermos que uma pessoa pode ter filhos, ela simplesmente pode optar por não ter. Portanto, ser fértil não implica (não é obrigatório) ter filhos, principalmente considerando a evolução dos métodos contraceptivos. De outra forma, mesmo não conhecendo a mãe, todas as pessoas (obrigatoriamente) possuem uma (o relacionamento ter/ser mãe obrigatoriamente relaciona duas pessoas).

Quando dizemos que um signo é composto por outros signos, a relação entre o signo composto e as suas partes pode *exigir* que alguma(s) parte(s) não possua(m) valor nulo, isto é, que um signo-parte tenha conteúdo obrigatório quando relacionado com um signo composto. Além disto, ser um signo-parte obrigatoriamente definido pela relação com um signo composto não determina que em todos os momentos da interação o signo-parte também será obrigatório (ou seja, pode ter um conteúdo vazio). Por exemplo, um signo-parte pode ser obrigatório para o cadastramento de um signo composto, mas ser indiferente quando se pesquisa signos compostos. Deste modo, a obrigatoriedade intrínseca do signo estará definida nas relações entre os signos representados na ontologia de signos e poderá ser modificada na descrição textual da cena em função do uso do signo.

Vejam, por exemplo, quais seriam os atributos do signo “nome do autor” em um sistema de biblioteca. Na cena onde o usuário informa os dados para cadastrar um autor, o signo “nome do autor” tipicamente teria as seguintes características:

| DIMENSÃO   | ATRIBUTO  | VALOR  |
|------------|---|--|
| Identidade | Identificador                                       | nome do autor  |
|            | Descrição   | O nome do autor descreve o nome de um autor de obras publicadas na biblioteca. |
|            | Origem  | signo de domínio   |
| Conteúdo   | Tipo  | texto  |
|            | Domínio   | -  |
| Expressão  | Elemento de interface ( <i>widget</i> ) recomendado | texto livre  |
|            | Formato esperado                                    | primeiro nome, (iniciais dos) sobrenomes intermediários e último sobrenome.    |

Tabela 2: Valores dos atributos do signo “nome do autor” para o cadastramento do autor.

Quais destes atributos mudariam quando o signo “nome do autor” fizesse parte dos diálogos da cena onde uma obra é cadastrada? Em geral, somente mudariam os atributos relacionados com a expressão do signo. Assim, o *widget* recomendado seria do tipo seleção simples (somente é possível selecionar um elemento do conjunto de autores cadastrados), cujo formato poderia ser <último sobrenome, “,”, primeiro nome, iniciais>.

Em outro momento da interação, o usuário poderia examinar o resultado de uma pesquisa pelo nome do autor das obras da biblioteca. Neste caso, não seriam somente o *widget* recomendado e o formato esperado que mudariam, mas também o enunciador do conteúdo do signo. Então, quando o enunciador do signo “nome do autor” passa a ser o preposto do designer, geralmente o *widget* recomendado é um rótulo e o formato esperado é <último sobrenome, primeiro nome e iniciais dos sobrenomes intermediários>.

Depois de exemplificarmos a variação dos valores destes atributos em função do momento de interação em que eles se encontram, resta-nos a questão: Quais destes atributos deveriam ser definidos na ontologia de signos e quais deveriam ser definidos na descrição textual de cada cena? Os atributos que dizem respeito à identidade e ao conteúdo do signo não deveriam mudar durante a interação. Sendo assim, eles deveriam ser definidos uma única vez na ontologia de signos. Já os atributos que dizem respeito à expressão do signo dependem de quem determina o valor do signo, isto é, o enunciador do signo, e da cena onde o signo é apresentado ou modificado. Deste modo, a ontologia de signos definirá valores default dos atributos relacionados com a expressão para duas situações: uma situação onde o usuário é o enunciador do signo (onde o signo se manifesta como um “dado de entrada”) e outra onde o preposto do designer é o enunciador (onde o signo se manifesta como um “dado de saída”). Quando o designer for detalhar os signos envolvidos nos diálogos de cada cena, ele só vai redefinir na descrição textual da cena aqueles atributos do signo que forem diferentes do valor default definido na ontologia de signos. Passa a valer, portanto, a última definição do valor dos atributos do signo. Esta forma de tratar a definição dos atributos do signo é baseada na representação de sistemas de *frames* (Minsky, 1975).

Voltemos ao exemplo do signo “nome do autor” no sistema de bibliotecas. A Tabela 3 descreve os atributos do signo “nome do autor” que estão definidos na ontologia de signos, com uma expressão default para cada enunciador do signo. No momento apropriado do projeto de interação, o detalhamento das cenas redefinirá os atributos necessários do signo “nome do autor”, conforme, por exemplo, o apresentado na Figura 46.

Quando estruturamos os atributos do signo em função da identidade (relacionada com a intenção do projetista), do conteúdo e da expressão do signo,

estimulamos o designer a pensar sobre estas dimensões importantes no processo de comunicação. Além disto, tornamos mais visíveis as diferenças e semelhanças destes atributos em cada cena, além da diferença (ou semelhança) geralmente vista somente como “dado de entrada ou de saída”.

| Signo: nome do autor |   |  |
|----------------------|---|--|
| Dimensão             | Atributo  | Valor  |
| Identidade           | Identificador                                       | nome do autor  |
|                      | Descrição   | O nome do autor descreve o nome de um autor de obras publicadas na biblioteca. |
|                      | Origem  | signo de domínio   |
| Conteúdo             | Tipo  | texto  |
|                      | Domínio   | -  |
|                      | Valor Padrão  | -  |
| Enunciador: Preposto |   |  |
| Expressão            | Elemento de interface ( <i>widget</i> ) recomendado | rótulo   |
|                      | Formato esperado                                    | último sobrenome, primeiro nome e (iniciais dos) sobrenomes intermediários.    |
| Enunciador: Usuário  |   |  |
| Expressão            | Elemento de interface ( <i>widget</i> ) recomendado | texto livre  |
|                      | Formato esperado                                    | nenhum   |

Tabela 3: Definição dos atributos do signo “nome do autor” na ontologia de signos.

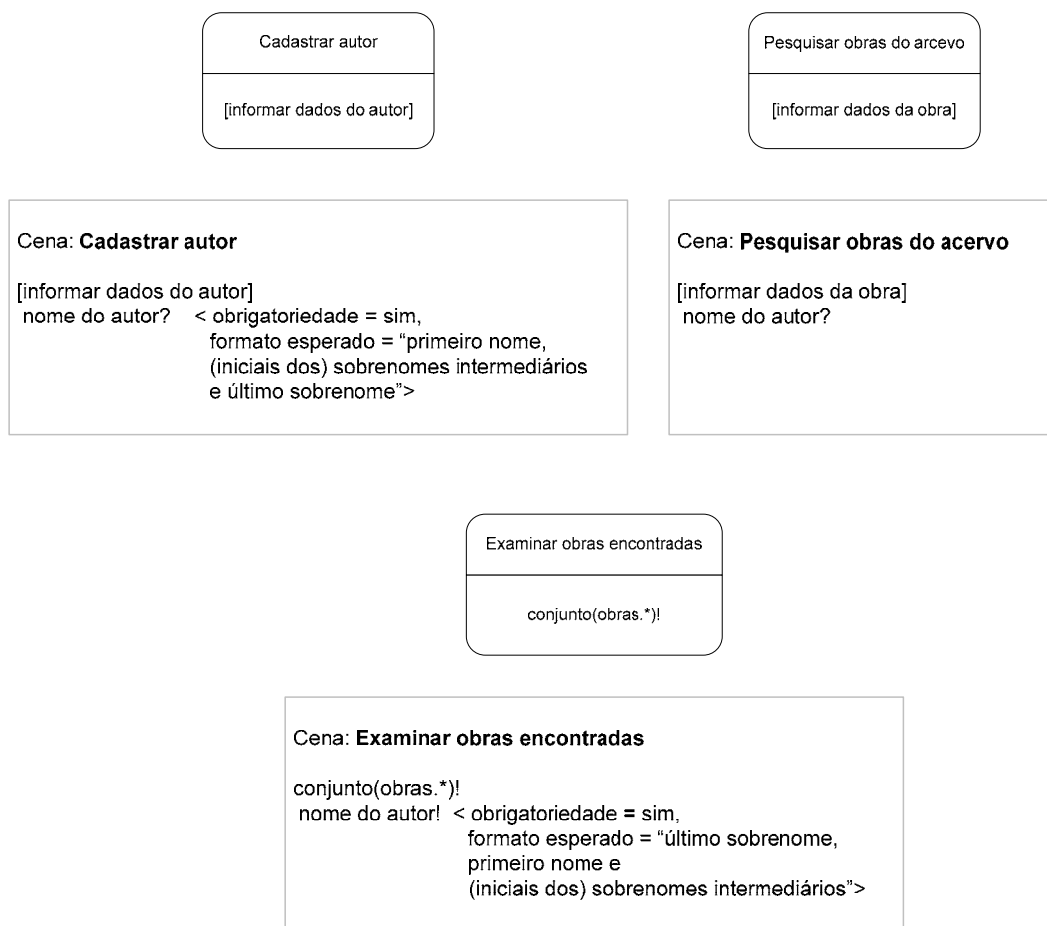


Figura 46: Redefinição dos atributos do signo na descrição textual de cada cena.

\*\*\*

Por fim, devemos lembrar do propósito de projeto baseado em modelos: trabalhar com um determinado foco e nível de abstração para tornar o espaço de problema mais facilmente tratável, por considerar um conjunto reduzido de interesses (Hoover et al., 1991). Este propósito se torna mais relevante ainda quando o modelo é utilizado por uma equipe multidisciplinar engajada no projeto, pois seu uso terá diferentes enfoques e propósitos em diferentes períodos no decorrer do projeto (Paula et al., 2005; Paula e Barbosa, 2004). Considerando que as questões sobre o uso da MoLIC endereçadas neste trabalho vão exigir propostas de extensão que aprofundam o tratamento de diferentes interesses (visão geral vs. detalhamento, interação vs. interface (recomendação de *widget* de interface, e etc),

**Questão 18.** Como organizar a atividade de design de IHC utilizando a MoLIC de modo que um conjunto reduzido de interesses, com foco e escopo bem definidos, possam ser endereçados a cada instante, favorecendo a participação de diferentes profissionais da equipe multidisciplinar de projeto?

Para facilitar o tratamento de diferentes questões de projeto de IHC, (Silva et al., 2005) propõem que o projeto de interação utilizando a MoLIC seja realizado em diferentes etapas, cada qual privilegiando um conjunto reduzido de questões de projeto. Eles identificaram duas etapas no projeto de interação, chamadas simplesmente de primeira e segunda etapa.

Na primeira etapa do projeto de interação, o objetivo do designer é especificar todas as possíveis conversas entre o usuário e o sistema (preposto do designer), mas ainda sem detalhá-las, não definindo, por exemplo, a estrutura dos diálogos, quais signos serão manipulados por eles e os atributos dos signos. O resultado desta primeira etapa deve ser um diagrama de interação ainda sem detalhes da conversa, que irá evoluir conforme o designer aprende e reflete sobre a interação. Projetar a interação sob uma metáfora de conversa sem entrar em detalhes necessários para definição da interface facilita a cooperação de uma equipe multidisciplinar (Paula et al. 2004, Paula e Barbosa, 2005), estimulando a reflexão, a análise e a discussão sobre a conversa (interação) sendo projetada.

Na etapa segunda etapa do projeto de interação, por sua vez, o designer concentra esforços (1) na estruturação dos diálogos definidos na etapa anterior, (2) nas possíveis restrições sobre o rumo da conversa (falas de transição que pressupõem um conjunto de diálogos travados), (3) na definição e estruturação dos signos manipulados pelos diálogos, e, por fim, (4) no detalhamento da ontologia de signos. Estas informações adicionais do projeto de interação servirão de insumo para o projeto e a implementação da interface em etapas futuras do processo de desenvolvimento.

\*\*\*

Neste capítulo, levantamos questões referentes ao uso da MoLIC. Conforme elaboramos as respostas destas questões, percebemos a necessidade de extensões à linguagem MoLIC e passamos a considerar as propostas preliminares de extensão existentes. Logo notamos que certas propostas preliminares de extensão eram desconexas e por vezes conflitantes. Isto nos exigiu um esforço



para aprimorá-las para manter a linguagem consistente e coerente com os seus fundamentos (capítulo 2.2). No próximo capítulo, enfim, apresentaremos o resultado deste estudo: a segunda edição da MoLIC.