

3 A linguagem MoLIC

A MoLIC, *Modeling Language for Interaction as Conversation*, é uma linguagem de modelagem utilizada para representar a interação humano-computador seguindo a metáfora de uma conversa entre usuário e designer, na qual a Engenharia Semiótica se baseia. A MoLIC foi criada em 2003 (de Paula & Barbosa, 2003) para apoiar a reflexão do designer ao elaborar as possíveis conversas que os usuários poderão travar através da interface do sistema. Considerando um processo de desenvolvimento, a MoLIC serve como uma ponte entre a etapa de análise de requisitos dos usuários e o projeto e construção da interface de sistemas computacionais interativos. Em 2005, Silva & Barbosa (2005) lançaram sua segunda edição, além de um guia prático de construção (Silva & Barbosa, 2007), que, juntos, formam a base da proposta deste trabalho. Em sua segunda edição, a MoLIC é composta de três artefatos inter-relacionados: o diagrama de metas, o diagrama de interação e o esquema conceitual de signos, apresentados brevemente a seguir.

3.1. Segunda Edição da MoLIC

O diagrama de metas organiza as metas dos diferentes papéis de usuário da aplicação, identificadas na etapa de análise, indicando *o quê* os usuários podem realizar com o sistema.

Em seguida, o designer modela a interação usuário-sistema de fato, através do diagrama de interação, que define *como* as metas dos diferentes papéis de usuários podem ser alcançadas durante a interação. A conversa designer-usuário modelada no diagrama de interação deve conter todos os assuntos e diálogos possíveis entre eles, mas sem apresentar detalhes de interface ou de plataforma tecnológica. Ao construir o diagrama de interação, o designer deve modelar todos

os possíveis caminhos de interação que foi capaz de prever, inclusive as eventuais rupturas de comunicação entre os interlocutores.

Em paralelo com a construção do diagrama de interação, o designer monta o esquema conceitual de signos, que define e organiza os conceitos envolvidos no sistema, detalhando os signos usados no diagrama de interação para servir como base entre a modelagem da interação e projeto da interface concreta. Atualmente, o designer especifica em tabelas a identificação, conteúdo, mecanismos de prevenção e recuperação de rupturas e possíveis expressões de todos os signos conceituais envolvidos em cada diálogo.

3.2. Análise Crítica da MoLIC

Ao longo do desenvolvimento deste trabalho, foram observadas algumas limitações da Segunda Edição da MoLIC, que resultaram de uma análise crítica de 21 diagramas de interação MoLIC modelados por cinco designers. Dos 21 diagramas, 17 eram de plataforma web, dos seguintes domínios: aplicações de e-mail, biblioteca virtual, busca, compra, fórum de discussão, gerência de aulas/turmas e gerência de matérias/avisos; 3 deles foram modelados para um posterior desenvolvimento em Prolog representando o comportamento de linguagens de interação; e 1 era de plataforma Windows, representando parte das funcionalidades de um editor de textos.

Na análise realizada, diversos problemas foram encontrados, alguns dos quais relacionados a limitações da própria MoLIC, e outros relacionados a falhas no entendimento do designer sobre como utilizar a linguagem. Cada problema observado será classificado conforme a legenda a seguir, utilizada daqui para frente:

L: limitação da MoLIC; ou

E: problema de entendimento ou uso errado da linguagem.

3.2.1. Limitações da Segunda Edição da MoLIC

A seguir serão descritas as 10 limitações da linguagem observadas na análise crítica dos diagramas de interação.

L1: não fica claro onde começam e terminam as conversas sobre as metas finais

Em diagramas de interação mais complexos, as possíveis conversas designer-usuário em direção uma determinada meta final (i.e., os possíveis caminhos de interação que o usuário pode percorrer para atingi-la) não são de fácil identificação. Isso ocorre em virtude do volume e nível de detalhamento das informações representadas no modelo que dificultam sua leitura. Na Segunda Edição da MoLIC, não há nenhuma representação que apóie o designer a

identificar com mais facilidade e precisão onde começam e onde terminam as metas finais dos (papéis de) usuários do sistema modelado.

L2: não fica claro quais são as conversas preferenciais do designer

Ao modelar um sistema, o designer pode prever diferentes formas de determinado (papel de) usuário atingir sua meta final. Essas alternativas estão comumente relacionadas a diferentes estratégias do usuário para atingir uma meta, a características específicas dos usuários, ou a objetivos de design, tais como alta comunicabilidade, facilidade de aprendizado, eficiência etc. Em alguns casos, o designer pode pressupor que uma dessas conversas é a sua interpretação preferencial para que o usuário atinja determinada meta final neste sistema e em determinado contexto. Nesses casos, é interessante que essa preferência seja representada para que essa informação não se perca, mas a Segunda Edição da MoLIC não fornece instrumentos para tal.

L3: não há uma denominação para a principal cena do diagrama

Em geral, tanto em aplicações desktop como em sistemas web, existe uma cena que concentra diversos inícios de conversa, tornando-se um ponto de partida para o usuário atingir muitas, e em alguns casos todas, metas finais. Essa cena deveria ter uma denominação especial, tanto para destacar sua relevância quanto para facilitar a troca de informações sobre o modelo entre diferentes indivíduos.

L4: a representação de monólogos é redundante

Na Segunda Edição da MoLIC, o monólogo é o elemento utilizado para finalizar uma conversa quando não há nada mais que o usuário possa fazer para atingir sua meta final. No entanto, a mensagem exibida pelo preposto no monólogo é idêntica ou muito similar à sua fala em resposta à fala do usuário, conforme mostra a Figura 2 a seguir.

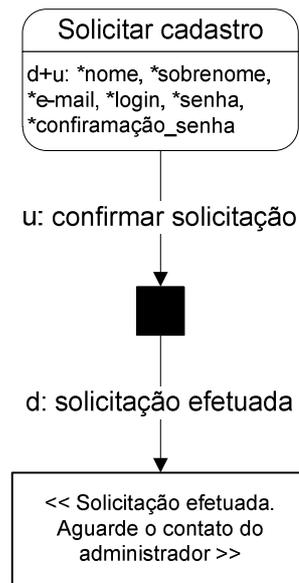


Figura 2: Monólogo exibindo mensagem similar à fala do preposto (figura adaptada de Silva e Barbosa, 2007).

L5: a representação de perlocuções através da pós-condição é pouco abrangente

A fala de transição do preposto tipicamente representa as perlocuções imediatas que têm que ser comunicadas ao usuário, ou seja, uma resposta direta à fala de transição do usuário que deu origem ao processamento do sistema, e que costuma estar fortemente relacionada ao atingimento da meta corrente. No entanto, há algumas perlocuções que são efeitos secundários ou colaterais, que podem afetar a interação futura do usuário com o sistema, mas não constituem uma resposta direta à fala do usuário nem afetam diretamente o curso da conversa corrente. Esses efeitos perlocutórios tipicamente modificam o contexto atual da conversa, e podem ou não ser comunicados ao usuário, conforme a decisão do designer.

Na Primeira edição da MoLIC, as perlocuções eram chamadas de **pós-condição**, mas, a partir da Segunda Edição da MoLIC, passaram a ser denominadas de **implicatura**, para reduzir a probabilidade de que o designer pensasse como um programador ao modelar o sistema, ao invés de adotar o papel de interlocutor em uma conversa com o usuário. Na revisão da linguagem feita recentemente por Silva e Barbosa (2007), o termo passou a ser **perlocução**, mas não foi adotado na prática pelos designers, que continuaram utilizando o termo **pós-condição**, e, assim, representando as perlocuções de forma incompleta.

L6: não há como representar de forma eficiente trechos de interação recorrentes

Na Segunda Edição da MoLIC, não há como representar uma situação genérica que se repete diversas vezes no diagrama de interação, tais como falhas de comunicação com o servidor e problemas de acesso a banco de dados. Dessa forma, o diagrama fica poluído e difícil de manter, já que, a cada alteração, todos os trechos recorrentes devem ser revistos.

L7: dificuldade de mapeamento do diagrama de metas para o diagrama de interação

Na Segunda Edição da MoLIC, as metas finais dos diferentes (papéis de) usuários do sistema são primeiro representadas no diagrama de metas, que representa *o que* os usuários podem realizar com o sistema, para depois serem modeladas no diagrama de interação, que define *como* essas metas podem ser alcançadas durante a interação. Dependendo de como o designer construiu o diagrama de metas, o mapeamento para o diagrama de interação pode se tornar muito custoso. As metas finais podem estar agrupadas no diagrama de metas de tal forma que dificulte, por exemplo, identificar o grau de interseção entre as metas finais dos diferentes papéis de usuário, que é uma forma de avaliar se deve ser modelado um diagrama de interação para cada papel de usuário ou um único para todos.

L8: ausência de discriminação entre prevenções passivas dependentes e independentes do contexto de interação corrente

Nem sempre uma prevenção passiva precisa ser apresentada a todo momento para o usuário. Dependendo do contexto, a prevenção passiva pode ser apresentada apenas quando necessário, para evitar fornecer informações desnecessárias ao usuário. Na Segunda Edição da MoLIC, não há essa distinção, e o designer pode acabar exibindo a prevenção sempre, mesmo quando não necessário, ou escolher um critério próprio para exibi-la, dificultando um futuro entendimento sobre o diagrama dele mesmo ou de terceiros.

L9: ausência de indicação de pressuposições realizadas pelo designer

A Segunda Edição da MoLIC prevê o uso de pré-condições em falas de transição, seja do usuário ou do preposto, utilizadas para restringir os momentos na interação em que se pode ter uma determinada conversa baseada em algumas condições. No entanto, não é prevista a necessidade do designer representar pressuposições sobre por que a fala seria enunciada.

L10: ausência da definição de um tipo de signo que apóie a representação de signos de contexto

Conforme citado pelos autores da Segunda Edição da MoLIC, não era o objetivo dessa revisão fornecer uma representação explícita para o contexto de interação. No entanto, as informações de contexto são de importância fundamental para um entendimento mais aprofundado do problema, e conseqüentemente da solução modelada.

3.2.2. Problemas de Entendimento ou Uso Errado da Linguagem

A seguir serão apresentados os 13 problemas de entendimento e uso errado da linguagem MoLIC, identificados nos diagramas de interação analisados.

E1: dificuldade de mapeamento das metas finais dos usuários no diagrama de interação

Na análise realizada, observou-se que algumas metas não foram adequadamente mapeadas nas cenas do diagrama de interação. Algumas cenas continham muita informação, enquanto outras, diretamente relacionadas, praticamente não continham informação útil. Notou-se que houve certa dificuldade dos designers em agrupar as cenas diretamente relacionadas, possivelmente pelo fato de que a maioria deles não realizou o passo anterior à modelagem da interação que é a identificação das metas finais dos diferentes papéis de usuário.

E2: ausência de pontos de entrada e de saída

Freqüentemente observou-se a ausência de pontos de entrada ou pontos de saída nos diagrama de interação analisados, e, em alguns casos, de ambos os

pontos no mesmo diagrama. A ausência desses pontos dificulta ao designer formular por completo as possíveis conversas sobre as metas finais, já que inicia a interação em uma parte intermediária, e não no início real da interação, e não a finaliza. Da mesma forma, para um leitor desse diagrama, a compreensão do mesmo torna-se incompleta e muito mais custosa, suscitando dúvidas ao longo de toda a leitura.

E3: uso inexistente ou inadequado de acessos ubíquos

Em alguns dos diagramas de interação analisados, observou-se a ausência de acessos ubíquos a determinadas cenas que podiam ser acessadas em qualquer momento de interação, dificultando a um leitor e ao próprio designer perceber qual era o real início da conversa sobre a meta final representada na cena em questão. Por outro lado, também notou-se a utilização de acessos ubíquos em momentos inadequados, nos quais havia restrições sobre o início da conversa que impediam que esta tivesse um acesso ubíquo de fato.

E4: ausência de conversas alternativas e de desistência em uma conversa

Na análise realizada, raramente observou-se conversas alternativas para se atingir uma mesma meta final, o que significa que muito provavelmente o sistema implementado será tão rígido e limitado quanto o diagrama modelado. A ausência de conversas alternativas para se atingir uma meta final implica que o usuário só poderá travar a conversa de uma única forma, não tendo, portanto, flexibilidade para escolher uma outra conversa mais adequada para atingir sua meta com uma determinada estratégia ou em virtude de alguma característica pessoal, tal como preferir usar o teclado ao invés do mouse.

Da mesma forma, a ausência de conversas de desistência foi bastante notada nos diagramas analisados, o que significa que os usuários dos sistemas modelados teriam que travar as conversas até o final, mesmo não querendo concluí-las, ou teriam que abortá-las de forma inadequada e indesejada, como por exemplo através de um acesso ubíquo para a cena principal do sistema, quando não há uma forma de cancelar a edição de seus dados pessoais em um sistema web.

E5: dificuldade de interligar cenas relacionadas e definir título de cenas

Ao analisar os diagramas de interação, observou-se que os designers muitas vezes não enxergaram a oportunidade de interligar, através de falas de transição, cenas diretamente relacionadas, fornecendo meios mais eficazes para os usuários travarem conversas correlatas oportunisticamente. Esta dificuldade pode ser consequência de uma outra, também identificada nessa análise, que é a definição dos títulos das cenas. Nos diagramas analisados, algumas cenas tinham títulos muito abrangentes, generalizando demais a conversa, e outras tinham títulos muito específicos. Em alguns casos, esse problema se agravava quando não havia um padrão no nível de granularidade da definição desses títulos.

E6: sub-utilização de diálogos e signos em cenas

Apesar dos diálogos e signos também poderem ser utilizados para encaminhar o usuário para a próxima conversa, além de permitir a conversa sobre o tópico corrente, observou-se em alguns diagramas que os designers nem sempre exploraram aquela possibilidade, que permitiria enriquecer a cena e apoiar o usuário a decidir qual seria a próxima conversa.

E7: sub-utilização de grupos de diálogos

Analisando os diagramas de interação, observou-se que os designers poderiam ter explorado melhor a organização dos diálogos em grupos, a fim de refletir uma estrutura de subtópicos. Isso se deve, possivelmente, à falta de conhecimento deste elemento da linguagem ou de sua utilidade na modelagem. Quando utilizada, a estrutura de grupos de diálogos favorece tanto o designer que está modelando a interação quanto um futuro leitor do diagrama, já que as informações se encontram visual e conceitualmente melhor organizadas.

E8: ausência de pré-condições de acesso a cenas, grupos de diálogos e falas que partem da cena

Na análise realizada, notou-se que algumas cenas deveriam ter pré-condições de acesso, pois as falas que iniciavam a conversa na cena só podiam ser emitidas em determinadas situações ou por determinados (papéis de) usuários. Isso se deve possivelmente à falta de cuidado na definição dos papéis de usuários da aplicação, etapa de grande importância na qual toda a modelagem da interação se baseia.

Por não terem sido modelados grupos de diálogos nos diagramas analisados, não foi possível observar se as pré-condições estavam sendo utilizadas corretamente neste tipo de elemento, mas em algumas falas que partiam de cenas notou-se a ausência de pré-condições que permitissem que apenas determinados perfis de usuários as emitissem.

E9: dificuldade de definição de emissor, conteúdo e obrigatoriedade de signos

Em alguns diagramas de interação analisados, observou-se que a terminologia utilizada para definir signos e diálogos não condizia com a Segunda Edição da MoLIC. Em alguns casos, ainda, os designers tiveram dificuldade para definir quem era o emissor dos signos, e em muitos casos esqueceram da necessidade de indicar que o diálogo era obrigatório para a continuação da conversa.

E10: ausência de mecanismos de prevenção e recuperação de rupturas

Em nenhum diagrama de interação analisado foram observados mecanismos de prevenção e recuperação de rupturas, apesar desta necessidade ter sido identificada em diversos pontos de diversos diagramas. Novamente, isto provavelmente é proveniente da falta de conhecimento sobre tais mecanismos ou de sua utilidade na modelagem de interação. Ao ignorá-los, os designers estão deixando a tarefa de prevenir e remediar erros para a etapa de construção da interface concreta, adiando decisões que poderiam ser tomadas mais cedo para evitar problemas no futuro, tais como retrabalho ou até mesmo indefinição de tais mecanismos.

E11: ausência de falas de transição em cenas

Em alguns diagramas específicos, observou-se algumas cenas que não possuíam falas de transição do usuário partindo delas. Em algumas delas, a meta realmente havia sido atingida, mas em outras o usuário não havia necessariamente completado a meta, mas não tinha o que falar a não ser por acessos ubíquos, o que restringia a gama de rumos que a conversa poderia tomar.

E12: ausência de indicação de efeitos perlocutórios em falas de transição

Pelo fato de terem sido modelados conforme a Segunda Edição da MoLIC, os diagramas de interação analisados não podiam representar por completo os efeitos perlocutórios em falas de transição. Apesar disso, os designers raramente utilizaram a pós-condição como um recurso para explicitar parte dos efeitos perlocutórios nas falas de transição cuja emissão provocava conseqüências em outra(s) conversa(s).

E13: ausência de indicação de conversas entre diferentes interlocutores

Em alguns diagramas de interação analisados, os designers não representaram conversas entre usuários, em especial de papéis diferentes, nem com sistemas externos ao que estava sendo modelado. Isso possivelmente se deve à falta de conhecimento, por parte dos designers, sobre os elementos da linguagem utilizados com esta finalidade.

3.3. Terceira Edição da MoLIC

Além da análise dos diagramas de interação que culminou na lista de limitações da Segunda Edição da MoLIC apresentada na seção 3.2.1, foi realizada uma elicitación informal com especialistas da linguagem e professores que a utilizam em aulas de graduação, visando capturar e explicitar textualmente (parte de) o conhecimento tácito desses indivíduos e os erros de interpretação comumente apresentados quando se está aprendendo a utilizar a linguagem.

Com base na análise dos diagramas e na elicitación informal com especialistas, foi elaborada uma revisão da linguagem que resultou na Terceira Edição da MoLIC cujas modificações serão descritas a seguir.

Para resolver a **L1: não fica claro onde começam e terminam as conversas sobre as metas finais** foi proposta a inclusão de dois elementos na linguagem: **abertura de conversa sobre meta final** e **fechamento de conversa sobre meta final**.

A **abertura de conversa sobre meta final** é o início da conversa sobre uma meta final de um (papel de) usuário, indicado nas falas de transição do usuário, sem ruptura, que chegam à primeira cena da conversa sobre esta meta. Tais falas de transição são representadas com uma bola preta pequena na extremidade inicial, fazendo alusão ao ponto de entrada no sistema, e com a indicação de abertura após o rótulo da fala, através da expressão **gco: meta final**, em que **gco** significa *goal-related conversation opening* (i.e. abertura de conversa relacionada à meta).

Se houver a abertura de conversa sobre duas ou mais metas finais na mesma fala de transição do usuário, estas são representadas entre vírgulas (e.g. gco: meta1, meta2, meta3).

O designer pode optar por representar ou não a abertura e o fechamento de conversa sobre as metas finais, mas preferencialmente de forma consistente em todo o diagrama (isto é, ou os representa sempre ou não os representa nunca).

Representação diagramática na Figura 3: Pode-se iniciar a conversa sobre uma meta final através de falas de transição do usuário partindo de um ponto de

entrada do sistema (Figura 3a), de um acesso ubíquo (Figura 3b) ou de uma cena (Figura 3c).

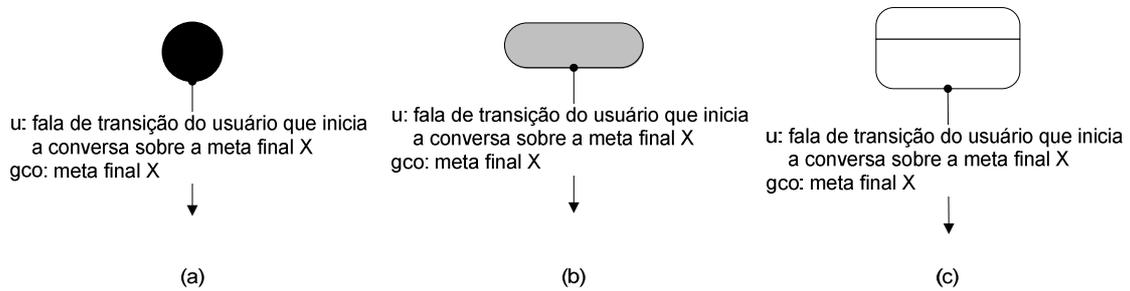


Figura 3: Representação diagramática de abertura de conversa sobre meta final a partir de (a) um ponto de entrada do sistema, (b) um acesso ubíquo ou (c) de uma cena.

Exemplo na Figura 4: A conversa sobre a meta final **Enviar e-mail** inicia-se na fala de transição do usuário **u: novo e-mail**, partindo do acesso ubíquo, onde é representada a abertura de conversa sobre esta meta final: **gco: Enviar e-mail**.

PUC-Rio - Certificação Digital Nº 0611886/CA

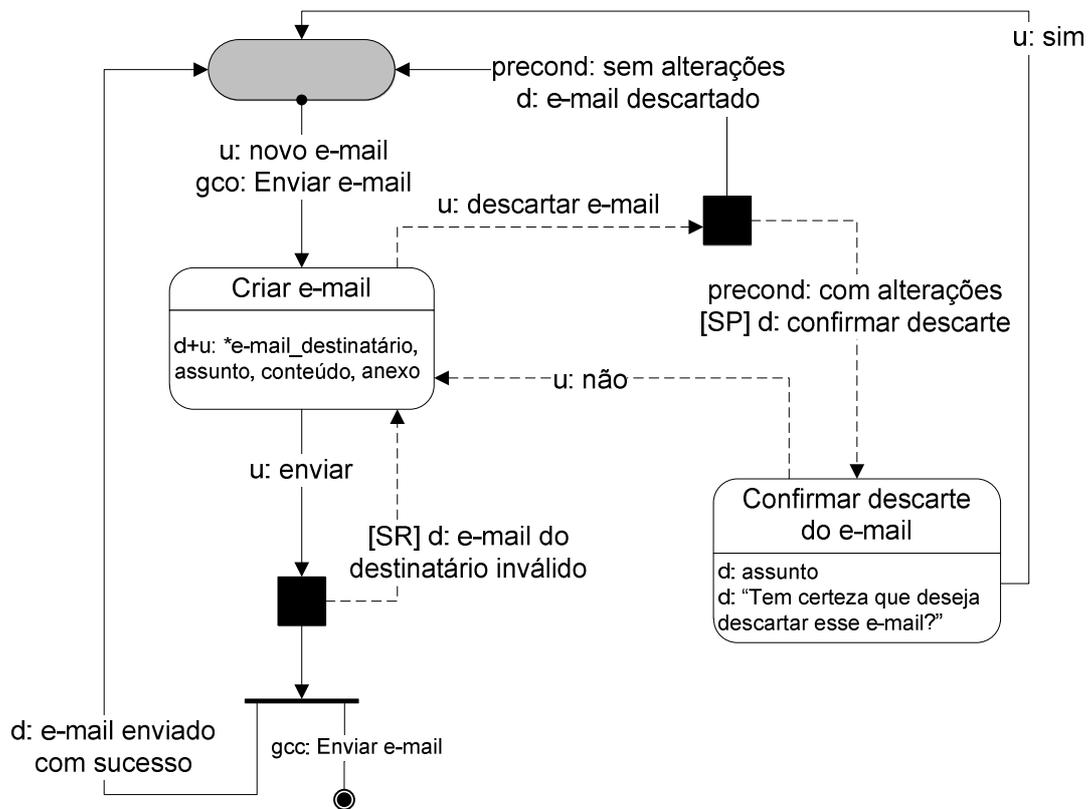


Figura 4: Exemplo de abertura de conversa sobre meta final.

Já o **fechamento de conversa sobre meta final** é o fim da conversa, bem sucedido ou não, sobre uma meta final de um (papel de) usuário, podendo ser representado tanto acoplado em uma cena como após um processamento do sistema. Existem três formas de se fechar a conversa sobre uma meta final: quando o usuário conclui a meta com sucesso, quando há rupturas de comunicação que levam a uma captura de erro ou quando o usuário desiste de atingir a meta em algum ponto da conversa que levaria à sua conclusão. Para se evitar poluir o diagrama desnecessariamente, sugere-se que a marcação de fechamento de conversa sobre meta final apresentada na Figura 56 só seja representada em caso de conclusão com sucesso, pois os casos mal sucedidos já ficam marcados de outra forma no diagrama.

O fechamento da conversa é indicado pela expressão **gcc: meta final**, na qual **gcc** significa *goal-related conversation closing* (i.e. fechamento de conversa relacionada à meta).

O designer pode optar por representar ou não a abertura e o fechamento de conversa sobre as metas finais, mas de forma consistente em todo o diagrama (isto é, ou os representa sempre ou não os representa nunca).

Representação diagramática na Figura 5: O fechamento de conversa sobre uma meta pode ser representado acoplado a uma cena (Figura 5a) ou após um processamento do sistema em conjunto com a fala do preposto que encerra a conversa sobre a meta, através de uma bifurcação (Figura 5b). A bifurcação é necessária neste último caso, pois o fechamento de conversa sobre a meta final tem que estar associado a uma fala específica de transição do preposto, que é a que indica o sucesso na conclusão da meta final.

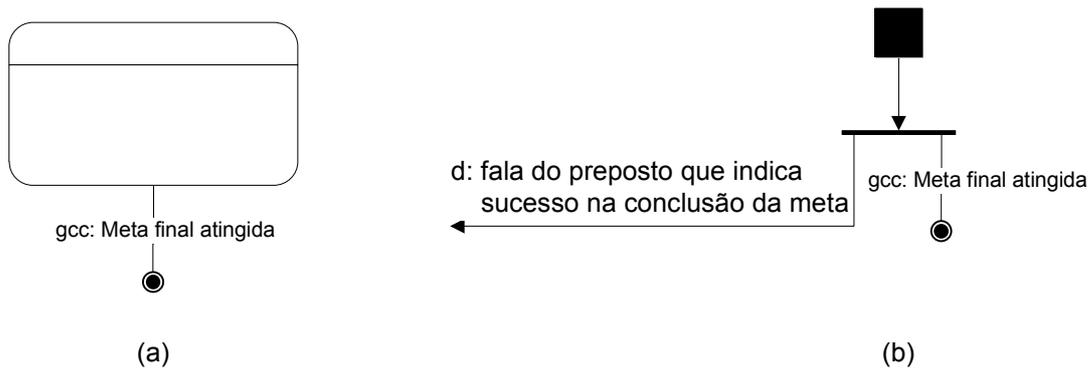


Figura 5: Representação diagramática de fechamento de conversa sobre meta final (a) acoplado a uma cena e (b) após um processamento do sistema.

Após o fechamento com sucesso da conversa sobre uma meta final, podem ocorrer três situações:

- i. O designer não tem nenhuma hipótese sobre o que o usuário fará em seguida. Neste caso, o preposto apenas indica o fechamento de conversa sobre a meta final, caso seja acoplado a uma cena (Figura 6a), ou o fechamento de conversa em conjunto com a fala do preposto que indica sucesso na conclusão da meta final, indo para uma cena vazia sem falas partindo dela (Figura 6b). A partir deste momento, em ambas as situações, o usuário só pode iniciar outra conversa a partir de acessos ubíquos;

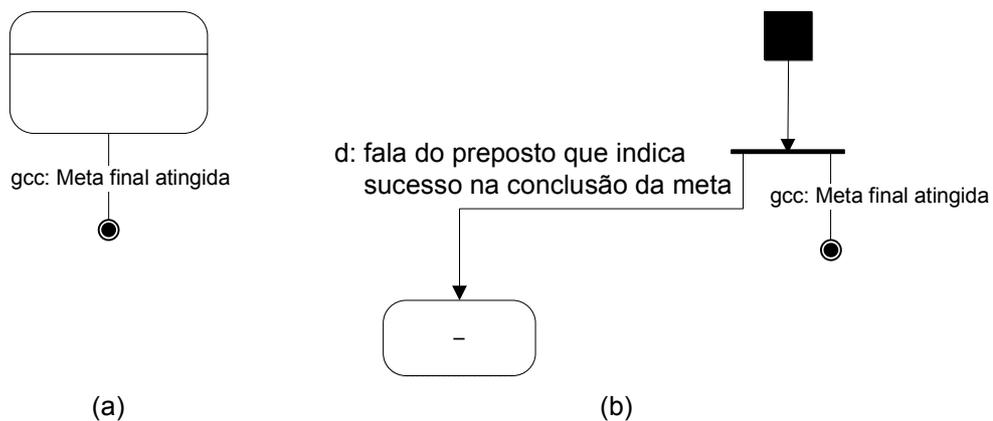


Figura 6: Designer não tem nenhuma hipótese sobre o que o usuário fará quando atingir a meta final em uma (a) cena ou (b) após um processamento do sistema.

- ii. O designer tem uma ou mais hipóteses sobre o que o usuário fará em seguida e dá a ele a oportunidade de decidir o que fazer, através de uma ou mais falas de transição, que partem da cena em que o usuário se encontra ao atingir a meta final (Figura 7a), ou da cena vazia para onde foi levado através de uma fala do preposto que indica sucesso na conclusão da meta final (Figura 7b). Em ambos os casos, as hipóteses do designer são indicadas através de pressuposições, e o usuário pode iniciar outra conversa a partir de uma dessas conversas previstas pelo designer ou por qualquer acesso ubíquo do diagrama; ou

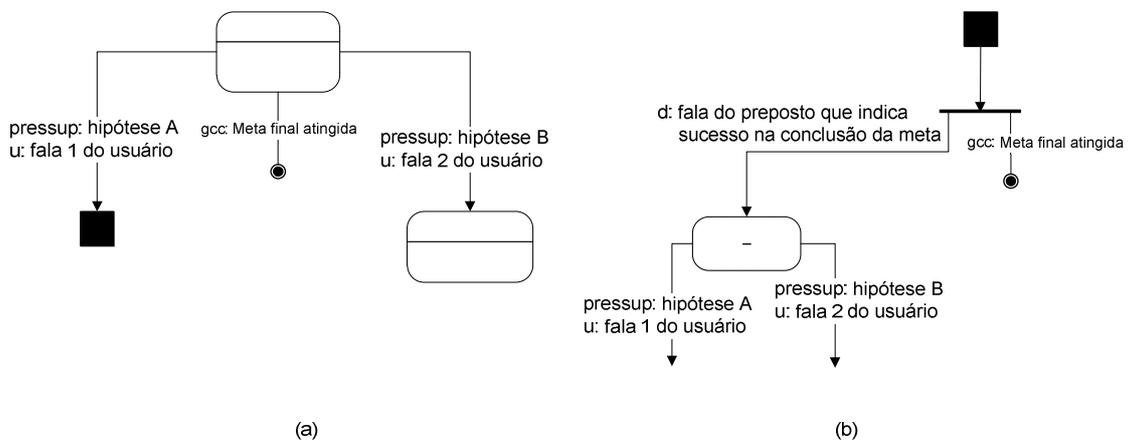


Figura 7: Designer formula hipóteses e permite que o usuário decida o que fazer quando atinge a meta final em uma (a) cena ou (b) após um processamento do sistema.

- iii. O designer tem uma hipótese tão forte sobre o que o usuário deveria fazer em seguida, que decide prescrever o próximo rumo da conversa, sem que o usuário participe desta decisão (Figura 8). Neste caso, o usuário obrigatoriamente inicia essa conversa prevista pelo designer.

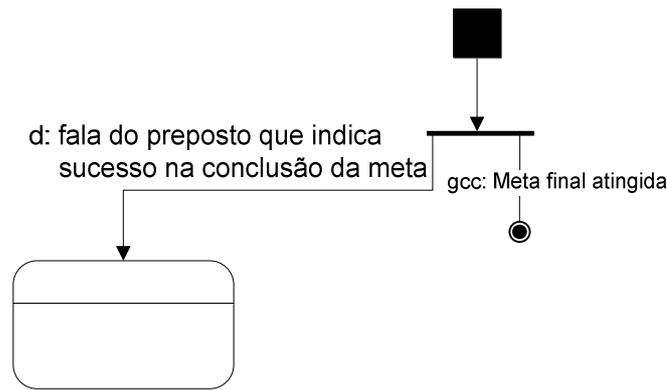


Figura 8: Preposto formula uma única hipótese forte e encaminha o usuário diretamente para a próxima conversa.

Nota: Esta terceira situação só pode ocorrer quando o fechamento de conversa sobre a meta final é representado após um *processamento*, nunca acoplado a uma cena, já que o designer só pode obrigar o usuário a prosseguir a conversa por um determinado rumo a partir de um processamento.

Exemplo na Figura 9: Após cadastrar um novo documento com sucesso, o preposto leva o usuário à cena **Ver documento**, que exibe o documento recém-cadastrado no sistema, finalizando a conversa sobre as metas **Ver documento** e **Cadastrar documento**. Para este momento de interação, o designer formula as hipóteses de que o usuário pode querer gerenciar outros documentos e, portanto, disponibiliza a fala **u: gerenciar documentos**, ou pode querer excluir o documento sendo visualizado, e, para isso, disponibiliza a fala **u: excluir documento X**. Neste momento, o usuário pode iniciar outra conversa por um desses dois rumos ou através de qualquer acesso ubíquo do diagrama.

Após o usuário solicitar a exclusão de determinado documento e este for excluído com sucesso, a meta **Excluir documento** é atingida (**gcc: Excluir documento**), de forma que o fechamento da conversa sobre esta meta é representado em conjunto com a fala do preposto **d: documento X excluído**, levando o usuário obrigatoriamente à cena **Gerenciar documentos**.

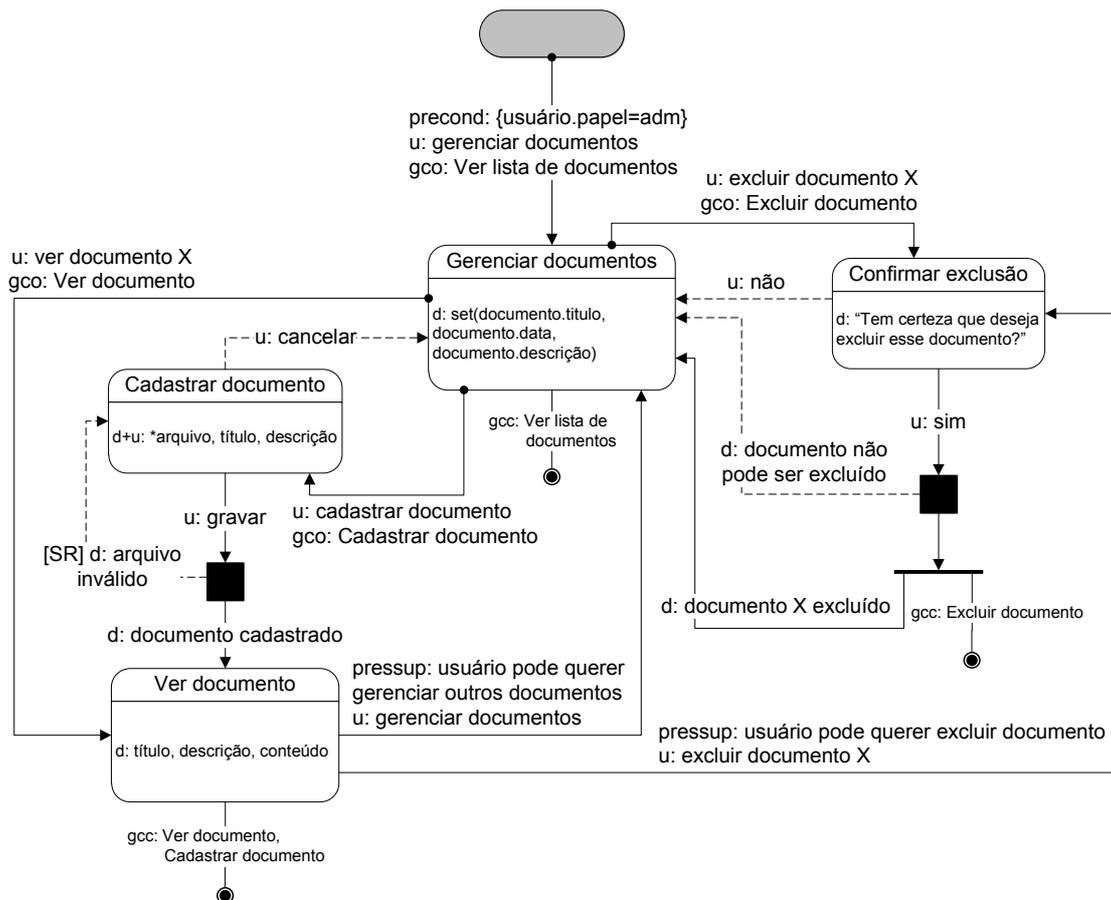


Figura 9: Exemplos de fechamento de conversa sobre metas finais.

Para resolver a **L2: não fica claro quais são as conversas preferenciais do designer** foi proposta a inclusão na linguagem do elemento **conversa preferencial**, que é a conversa preferencial do designer para que determinado papel de usuário, em determinado contexto, atinja sua meta. Em geral, é formado por uma seqüência de falas, processamentos e cenas, e eventualmente por acessos ubíquos, pontos de contato, pontos de entrada e de saída, terminando sempre em uma marcação de fim de conversa sobre a meta em questão.

Exemplo na Figura 10: Apesar de ser possível excluir um documento diretamente da gerência de documentos, a conversa preferencial do designer para o administrador atingir a meta final **Excluir documento** de um sistema de repositório de arquivos é: fala **u: ver documento X**; cena **Ver documento**; fala **u: excluir documento X**; cena **Confirmar exclusão**; fala **u: sim**; processamento de

exclusão de documento; fala **d: documento X excluído**; fechamento de conversa **gcc: Excluir documento**.

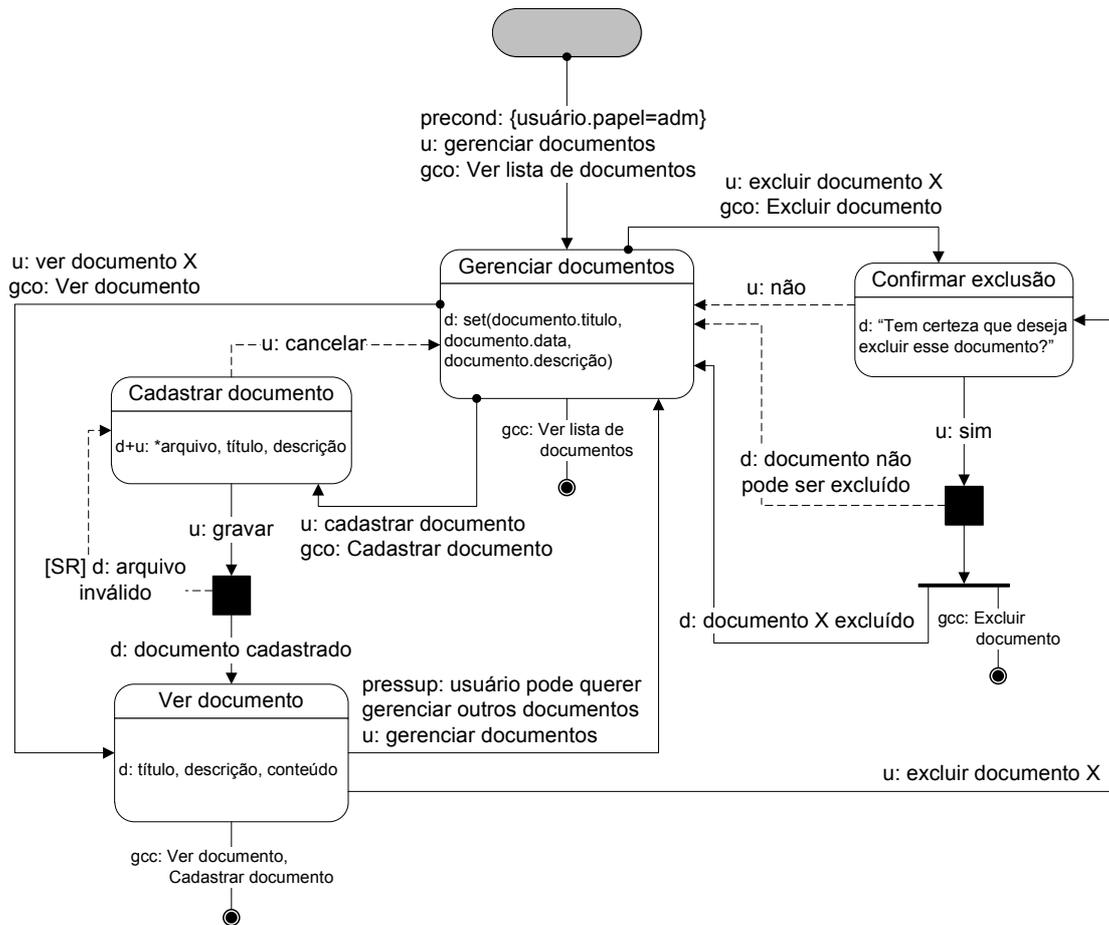


Figura 10: Exemplo de conversa preferencial para exclusão de documento.

Para resolver a **L3: não há uma denominação para a principal cena do diagrama** foi proposta a inclusão na linguagem do termo **cena principal**, que, em geral, é a primeira cena após o ponto de entrada do sistema, caso não seja necessário que o usuário se identifique (*login*). A cena principal do sistema costuma se destacar não apenas pela proximidade ao ponto de entrada, mas também por ter acesso ubíquo e por agrupar os pontos de partida para uma ou mais metas finais do usuário. Na interface, é tipicamente a *home page* em um site web ou a janela principal em uma aplicação desktop.

Exemplo na Figura 11: A cena **Ver caixa de entrada** é a cena principal de uma aplicação de e-mail, agrupando os pontos de partida para várias metas finais do usuário.

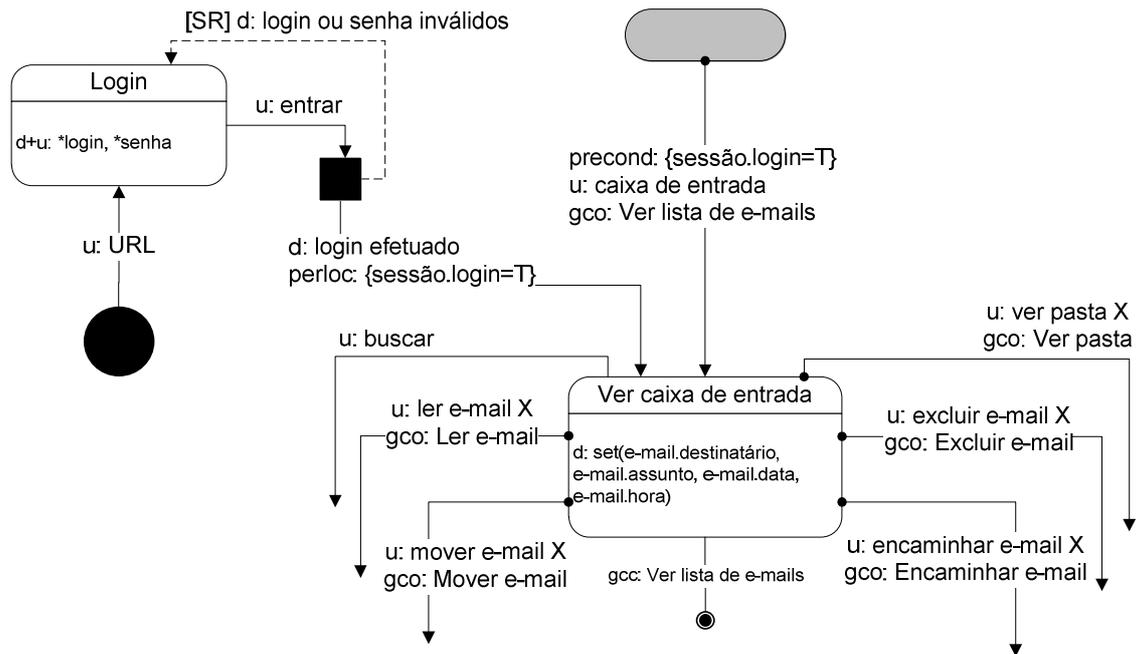


Figura 11: Exemplo de cena principal.

Para resolver a **L4: a representação de monólogos é redundante** foi proposta a substituição do monólogo pela **cena vazia**, utilizada após um processamento do sistema quando o designer não tem nenhuma hipótese ou tem uma ou mais hipóteses fracas sobre o que o usuário fará em seguida. Nesses casos, a fala de transição do preposto que indica o resultado do processamento leva a uma cena vazia, que não possui tópico, diálogos nem signos. O tópico implícito da conversa em uma cena vazia é: “Examinar resposta à <<fala anterior do usuário que ocasionou o processamento>>”.

Representação diagramática na Figura 12:



Figura 12: Representação diagramática de cena vazia.

Exemplo na Figura 13: Após o usuário solicitar seu cadastro com sucesso, a solicitação é enviada por e-mail para o administrador, enquanto o usuário é informado em uma cena vazia de que a solicitação foi efetuada e de que deve aguardar uma resposta por e-mail. O designer optou por representar desta forma, pois as respostas a solicitações de cadastro podem demorar dias para serem

aprovadas ou rejeitadas pelo administrador, portanto não faria sentido, por exemplo, levar o usuário para a tela de *login*.

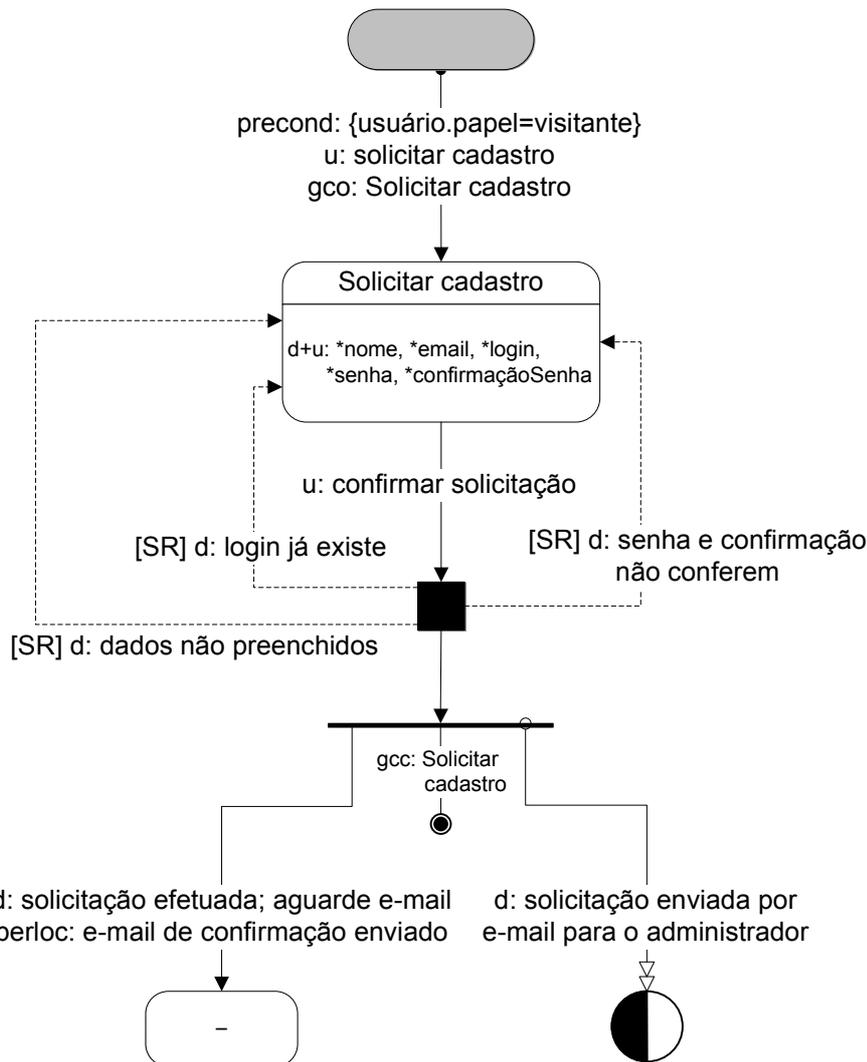


Figura 13: Exemplo de cena vazia.

Para resolver a **L5: a representação de perlocuções através da pós-condição é pouco abrangente** foi proposta a substituição do termo pós-condição por **efeito perlocutório**, melhor explicada a seguir.

A fala de transição do preposto tipicamente representa as perlocuções imediatas que têm que ser comunicadas ao usuário, ou seja, uma resposta direta à fala de transição do usuário que deu origem ao processamento do sistema, e que costuma estar fortemente relacionada ao atingimento da meta corrente (Figura 14a). No entanto, há algumas perlocuções que são efeitos secundários ou colaterais, que podem afetar a interação futura do usuário com o sistema, mas não constituem uma resposta direta à fala do usuário nem afetam diretamente o curso

da conversa corrente. Esses efeitos perlocutórios tipicamente modificam o contexto atual da conversa, e podem (Figura 14b) ou não (Figura 14c) ser comunicados ao usuário, conforme a decisão do designer.

O efeito perlocutório é precedido do prefixo **perloc:**.

Representação diagramática na Figura 14:

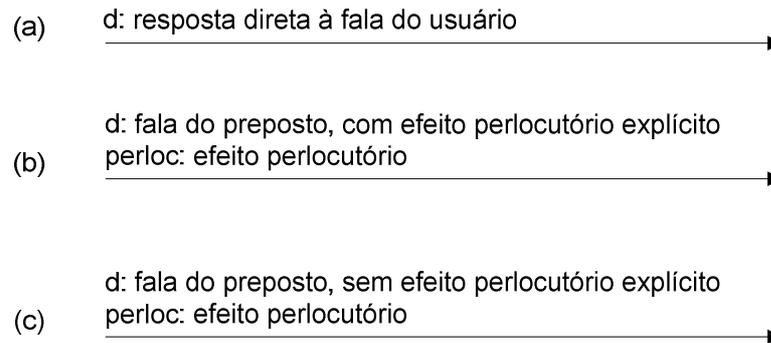


Figura 14: (a) Resposta direta à fala do usuário, (b) fala do preposto com efeito perlocutório explícito e (c) sem efeito perlocutório explícito.

Exemplo na Figura 15: No cadastro de um novo usuário, tanto a fala de sucesso (**d: usuário cadastrado**) quanto a de falha (**d: dados não preenchidos ou inválidos**) são efeitos perlocutórios representados simplesmente como respostas diretas à fala do usuário **u: cadastrar**.

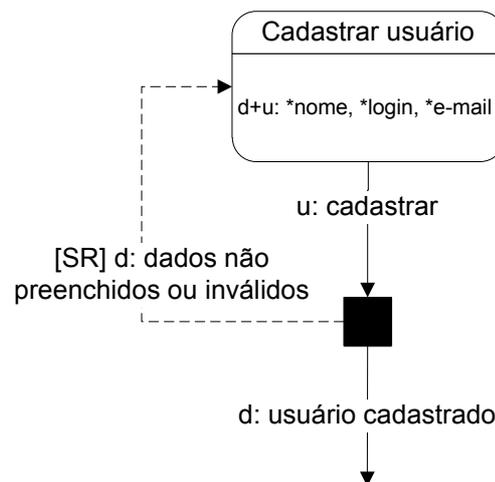


Figura 15: Exemplo de resposta direta à fala do usuário.

Exemplo na Figura 16: O efeito perlocutório da fala **d: pelo menos 1 item encontrado** (**perloc: {resultado=T}**), é comunicado explicitamente para o usuário: “**durante toda esta sessão, este resultado estará disponível**”.

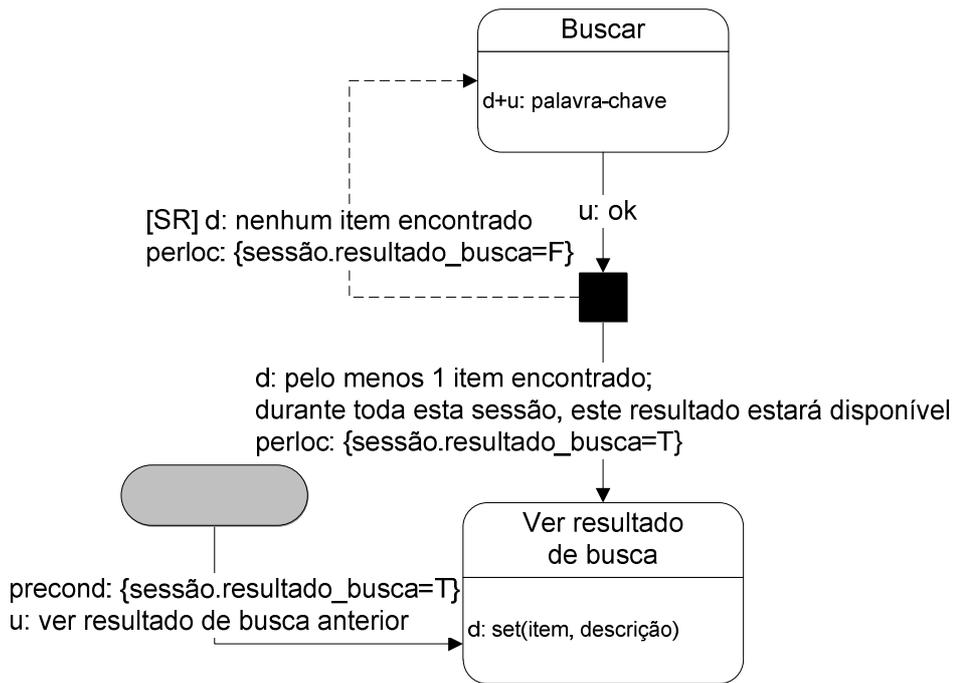


Figura 16: Exemplo de fala do preposto com efeito perlocutório explícito.

Exemplo na Figura 17: O efeito perlocutório da fala **d: pelo menos 1 item encontrado (perloc: {resultado=T})**, não é comunicado explicitamente para o usuário, apesar do resultado de busca ficar disponível durante toda a sessão.

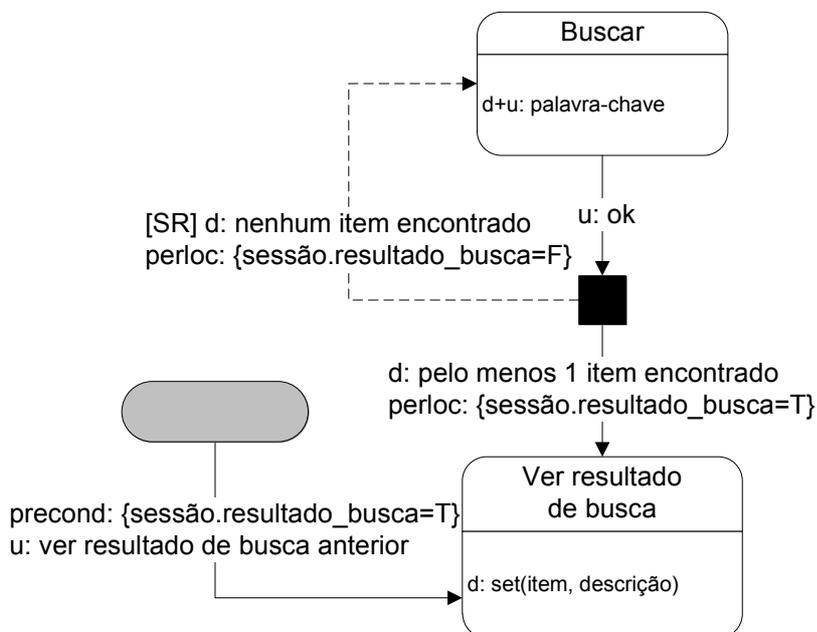


Figura 17: Exemplo de fala do preposto sem efeito perlocutório explícito.

Para resolver a **L6: não há como representar de forma eficiente trechos de interação recorrentes** foi proposta a inclusão do elemento **estereótipo**, utilizado para representar, num diagrama à parte, uma situação genérica que se repete no diagrama de interação. É muito comum representar situações em que pode haver rupturas sistêmicas, tais como falhas de comunicação com o servidor e problemas de acesso a banco de dados, as quais se evita representar no diagrama de interação em si para não poluí-lo desnecessariamente.

Exemplo na Figura 18: Este estereótipo será usado sempre que, dada uma cena X qualquer, o usuário emitir uma fala que dispara um processamento cujo resultado pode ser uma falha na conexão com o banco de dados.

Uma dessas situações está ilustrada na Figura 19: na cena **Cadastrar usuário**, o usuário emite a fala **u: cadastrar**, que dispara um processamento cujo resultado pode ser, dentre outros, uma falha de conexão com o banco de dados. Se essa ruptura ocorrer, será tratada conforme o estereótipo da Figura 18: após a falha na conexão com o banco de dados, o preposto leva o usuário a uma cena para examinar as causas do problema e permitir que ele tente realizar o processamento novamente, volte para a cena **Cadastrar usuário**, ou desista de atingir esta meta, indo para a cena principal do sistema. O resultado concreto da aplicação desse estereótipo está ilustrado na Figura 20 para um melhor entendimento do leitor, mas não deve ser representado explicitamente.

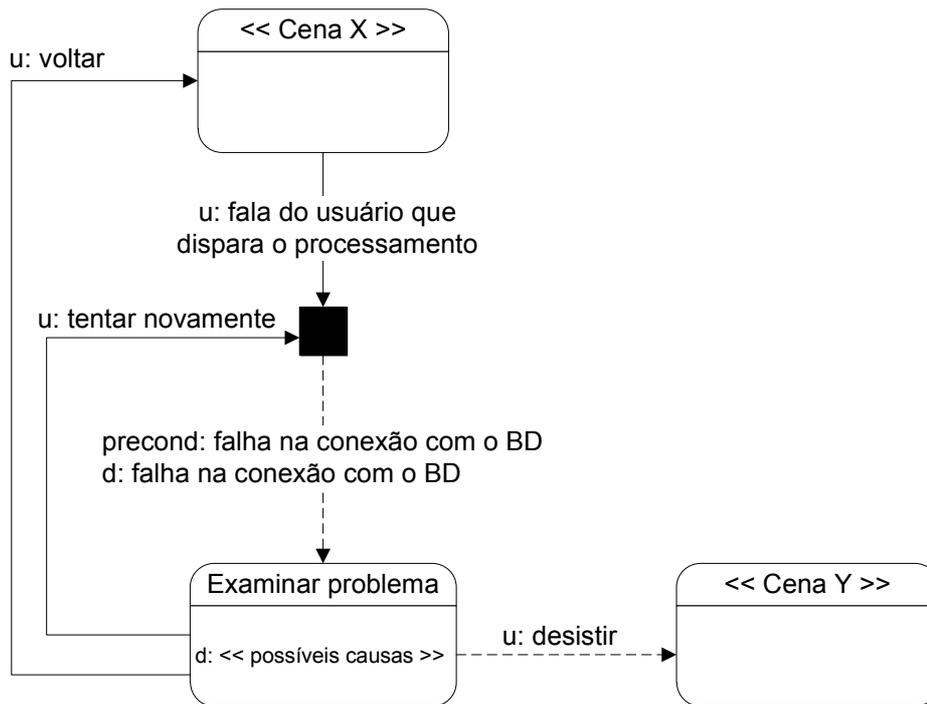


Figura 18: Exemplo de estereótipo.

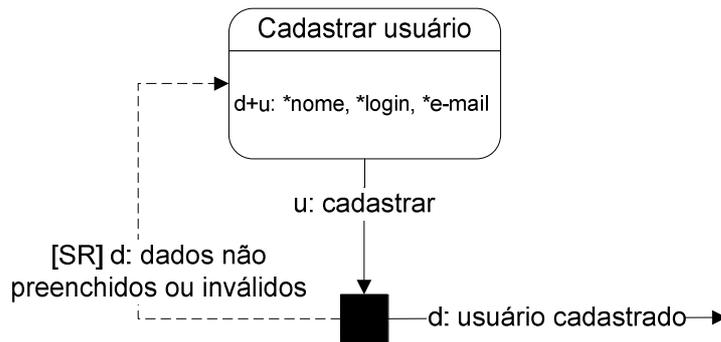


Figura 19: Trecho do diagrama em que o estereótipo se aplica.

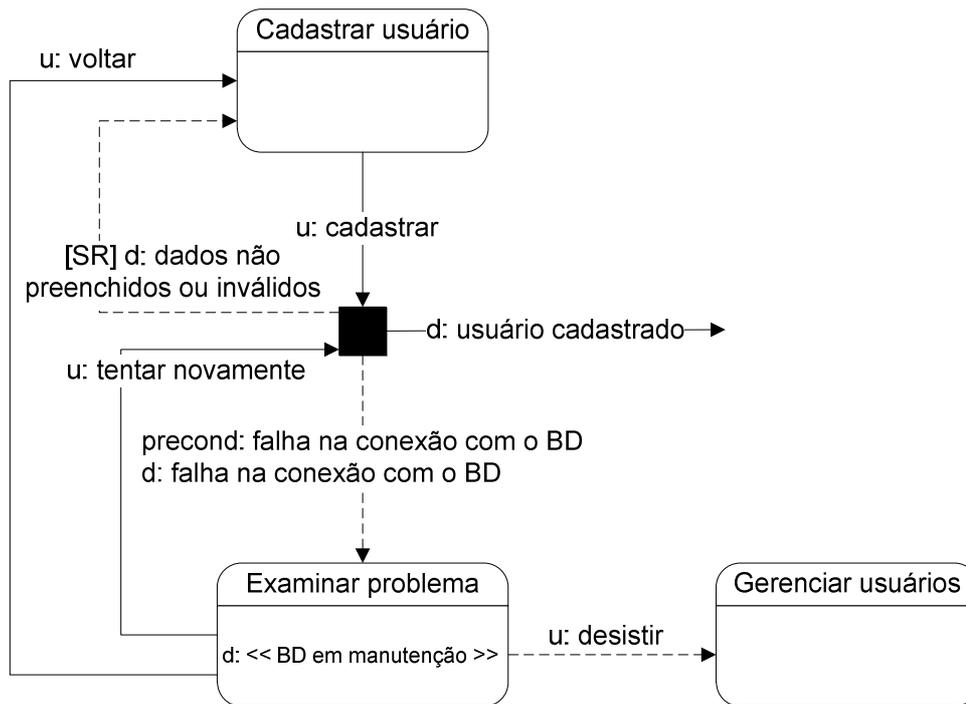


Figura 20: Resultado da aplicação do estereótipo.

Para resolver a **L7: dificuldade de mapeamento do diagrama de metas para o diagrama de interação** foi proposta a substituição do artefato diagrama de metas por uma **lista de metas**, que continua organizando as metas dos diferentes papéis de usuário, identificadas na etapa de análise, e indicando *o que* os usuários podem realizar com o sistema, mas permite uma representação menos rígida que a estrutura hierárquica do diagrama de metas.

Dessa forma, o designer pode representar as metas finais dos papéis de usuário da maneira que lhe for mais conveniente, conforme alguns critérios, tais como: domínio do sistema, quantidade de papéis de usuário, quantidade de metas de cada papel, sua experiência e facilidade de manipulação com cada tipo de representação, entre outros. Ao realizar essa escolha sobre qual é a melhor forma de representar as metas dos usuários, o designer poderá prever o conseqüente mapeamento dessas metas para o diagrama de interação. Assim, poderá, já na etapa de descrição das metas, agrupá-las de forma que possa, na próxima etapa, mapeá-las em cenas relacionadas no diagrama de interação. Além disso, poderá observar, já nessa etapa, o grau de interseção entre as metas dos diferentes papéis de usuário, a fim de tomar, tão cedo quanto possível, a decisão de modelar apenas um diagrama de interação para todos os papéis de usuário ou um diagrama para cada papel.

Para resolver a **L8: ausência de discriminação entre prevenções passivas dependentes e independentes do contexto de interação corrente** foi proposta a categorização do elemento **prevenção passiva** em **condicional** e **permanente**.

A prevenção passiva é um mecanismo de prevenção de ruptura no qual as rupturas podem ser evitadas por documentação ou instruções *online* explícitas. Por ser apenas uma instrução, e, portanto, não impedir a ruptura, geralmente é acompanhada de uma recuperação apoiada, que é um mecanismo de recuperação de ruptura, utilizado para ajudar o usuário a se recuperar, caso a ruptura ocorra.

A prevenção passiva pode ser condicional ou permanente. Para evitar poluir o diagrama de interação, ambos os tipos de prevenção passiva são representados apenas no esquema conceitual de signos.

Representação no esquema conceitual de signos: Com a expressão **PPc:** (*passive prevention - conditional*) ou **PPp:** (*passive prevention - permanent*) precedendo a prevenção na tabela de conteúdo de signo-tipo, valor de signo-token e mecanismos de prevenção e recuperação de rupturas.

A **prevenção passiva condicional** é apresentada apenas quando necessário, dependendo do contexto de interação corrente, para evitar fornecer informações desnecessárias ao usuário. É a instrução que o designer pressupõe que o usuário não vá precisar, ou só vá precisar na primeira vez em que estiver travando aquela conversa.

Há duas situações típicas em que uma prevenção passiva condicional é utilizada: (a) quando existe uma prevenção ativa para um signo e se o usuário tenta violá-la, é exibida uma prevenção passiva condicional instruindo o usuário sobre o signo ou seu conteúdo esperado; ou (b) quando o usuário pede uma prevenção passiva condicional sob demanda, tais como *tooltips* e ícones de ajuda contextualizada.

No esquema conceitual de signos, deve-se indicar, além do conteúdo da prevenção, a situação em que é exibida.

Exemplo na Tabela 1: **PPc:** [se PA foi ativada] máximo 20 caracteres é a prevenção passiva condicional para o signo **usuário.login** que só é exibida

quando ocorrer a prevenção ativa, isto é, quando o usuário fornecer o 20º caracter e o sistema impedi-lo ativamente de fornecer o próximo. Já a prevenção passiva condicional para o signo **usuário.apelido** é **PPc: [ajuda contextualizada] forma como você será reconhecido pelos outros usuários**, exibida através de um ícone de ajuda contextualizada.

Tabela 1: Exemplo de prevenção passiva condicional no esquema conceitual de signos.

id	conteúdo do signo-tipo			valor do signo-token			ruptura	
	descrição	origem	tipo	card	rest.	default	prevenção	recuperação
usuário .login	login do usuário	padrão	texto	1	máx 20 chars; deve ser único	-	PA PPc: [se PA foi ativada] máximo 20	SR: login inválido
usuário .apelido	apelido do usuário	domínio	texto	1	máx 20 chars	-	PPc: [ajuda contextualizada] forma como você será reconhecido pelos outros	

Já a **prevenção passiva permanente** é apresentada sempre, independente do contexto de interação corrente.

Exemplo na Tabela 2: A prevenção passiva permanente para o signo **usuário.senha** está diretamente relacionada à sua restrição, e, portanto, está implícita na coluna de prevenção (**PPp**), indicando que a senha do usuário tem que ter pelo menos um número ou símbolo. Já a prevenção passiva permanente para o signo **usuário.e-mail** não está diretamente relacionada à sua restrição, e, portanto, é explicitada na coluna prevenção como **PP: exemplo@servidor.com.br**, indicando o formato esperado para o e-mail do usuário. Ambas as prevenções são sempre exibidas pelo preposto, independente do contexto de interação corrente.

Tabela 2: Exemplo de prevenção passiva permanente no esquema conceitual de signos.

id	conteúdo do signo-tipo			valor do signo-token			ruptura	
	descrição	origem	tipo	card	rest.	default	prevenção	recuperação
usuário .senha	senha do usuário	padrão	texto	1	pelo menos um número ou símbolo	-	PPp	SR
usuário .e-mail	e-mail do usuário	domíni o	texto	1	pelo menos um @ e um ponto	-	PPp: exemplo @servido r.com.br	SR

Para resolver a **L9: ausência de indicação de pressuposições realizadas pelo designer** foi proposta a inclusão do elemento **pressuposição**, utilizada em falas de transição (para recuperação de rupturas ou não), para indicar uma pressuposição do designer sobre por que a fala seria enunciada. A pressuposição não impede que a fala seja emitida, ou seja, pode ou não ser satisfeita, ao contrário da pré-condição que restringe a emissão da fala, caso não seja satisfeita. A pressuposição é precedida do prefixo **pressup:**

Exemplo na Figura 21: Após finalizar a conversa sobre a meta **Ver dados do usuário X**, o preposto pressupõe que o usuário irá querer voltar à gerência de usuários, permitindo que ele o faça através da fala **u: gerenciar usuários**.

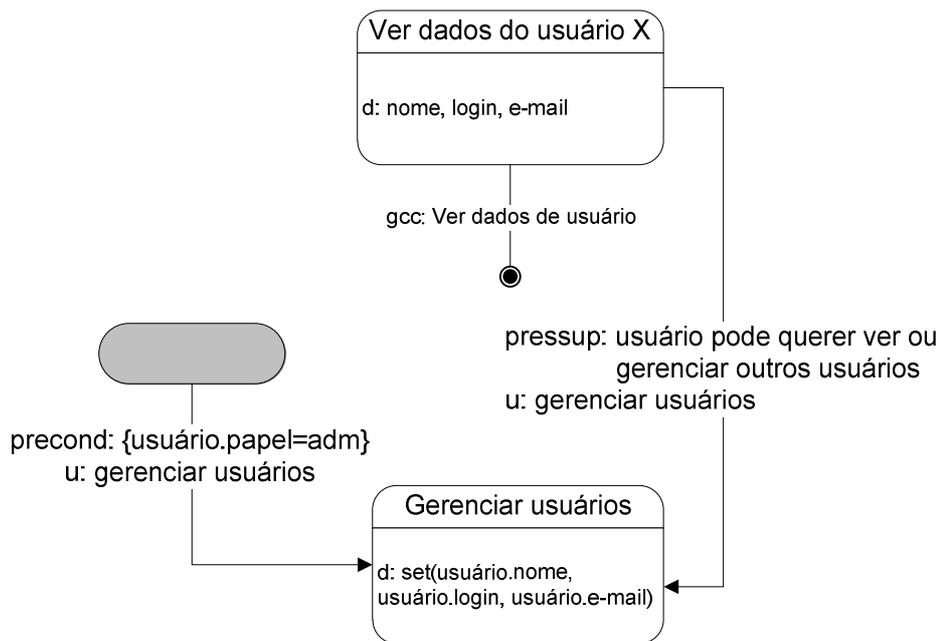


Figura 21: Exemplo de pressuposição do designer.

Para resolver a **L10: ausência da definição de um tipo de signo que apóie a representação de signos de contexto** foi proposta a inclusão do elemento **signo articulado** e sua associação com o **signo de contexto**.

O **signo articulado** é uma partição do contexto definida pelo designer. Signos articulados comuns são: sessão, usuário, máquina, documento ou objeto sendo manipulado e localização do usuário.

O **signo de contexto** é utilizado para representar parte do contexto da interação, qualificando falas do usuário ou do preposto com pré-condições, pressuposições ou efeitos perlocutórios. Um signo de contexto pode estar vinculado a um ou mais signos articulados, correspondentes a partições do contexto tal como definidas pelo designer.

Representação: Quando está vinculado a apenas um signo articulado: {signo_articulado.signo}; quando está vinculado a mais de um signo articulado: {(signo_articulado+ signo_articulado2).signo}.

Exemplos:

- Sessão: se há resultados de busca; se o *login* foi efetuado
Representação: sessão.busca_realizada; sessão.login (ou simplesmente login)
- Usuário: preferências e signos culturais do usuário
Representação: usuário.preferência.cor; usuário.cultura.idioma
- Máquina: configurações de impressora em uma máquina
Representação: máquina.impressora.Epson950c.tamanho_papel
- Documento ou objeto sendo manipulado: estilos no Word
Representação: documento.estilos
- Localização do usuário: previsão do tempo da cidade em que o usuário se encontra; impressora *default* utilizada no ambiente de rede
Representação: localização.previsão_tempo;
localização.impressora_default

- Vinculado a mais de um elemento: a cor das marcas de revisão do usuário em um documento no Word (signo vinculado ao usuário e ao documento).

Representação: (usuário+documento).cor_marcas_revisão

Os problemas de entendimento ou uso errado da linguagem serão endereçados pelas perguntas apresentadas no próximo capítulo, que constitui a principal contribuição deste trabalho.