

3

A Ferramenta ComunIHC-ES

Este trabalho propõe o uso da ferramenta ComunIHC-ES para aumentar o entendimento e apoiar a comunicação entre os profissionais de IHC e os engenheiros de software sobre o conteúdo do projeto da interação. Em outras palavras, a ComunIHC-ES tem como objetivo explicitar o projeto da camada de interação para os engenheiros de software, e a partir disto, criar um entendimento comum da solução de IHC e possibilitar a negociação desta entre estes dois tipos de profissionais.

Espera-se que a interpretação dos engenheiros de software sobre a solução projetada seja compatível com a do designer de IHC, para que se crie um entendimento comum sobre o projeto da interação, possibilitando que cada tipo de profissional siga o seu trabalho baseando-se em um mesmo mapa global de interação (por exemplo, a definição da interface final pelos designers de IHC e a especificação e implementação das funcionalidades do software pelos engenheiros de software).

A construção da ComunIHC-ES se baseou na definição do conteúdo necessário para se comunicar o projeto de IHC para os engenheiros de software apresentado na seção 1.2.1. Ou seja, ela levou em consideração a explicitação das seguintes informações: quem são os usuários da aplicação e o seu contexto, as metas que os usuários desejam alcançar através do sistema, e o projeto da comunicação usuário-preposto do designer e os motivos de a solução ter sido proposta desta forma.

Para representar todas essas informações, a ferramenta é constituída pelos seguintes componentes (Figura 9): a apresentação do conhecimento sobre o domínio em questão, seus usuários, as tarefas envolvidas e o contexto de uso (parte 1), uma linguagem para se projetar a interação (parte 2) e elementos para apoiar a explicação deste projeto pelo designer de IHC (parte 3). Estes componentes serão descritos em detalhes nas próximas seções.

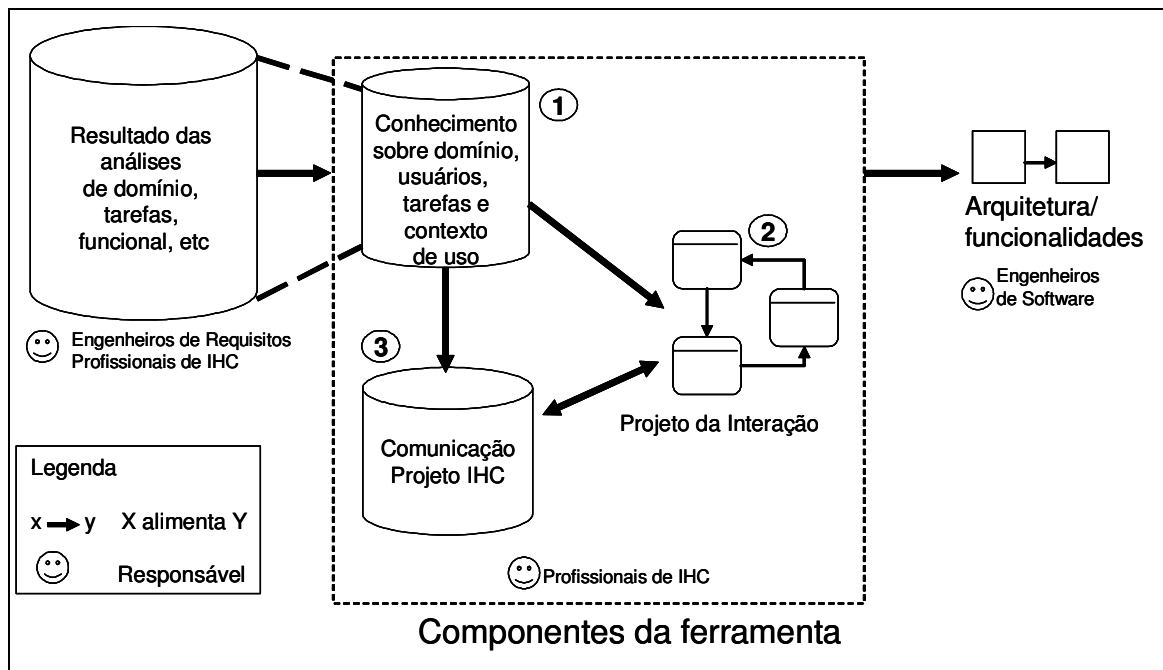


Figura 9: Componentes da Ferramenta ComunIHC-ES.

3.1 Conhecimento sobre domínio, usuários, tarefas e contexto de uso

A parte 1 da ferramenta, representada na Figura 9, tem como conteúdo:

- a descrição geral do domínio da aplicação;
- a descrição dos conceitos do domínio necessários para o entendimento do problema, através, por exemplo, de um glossário (pode-se usar um léxico, como por exemplo, a *Language Extended Lexicon* (LEL) (Leite et al. 1992 e Breitman e Leite, 2003), descrita mais adiante neste capítulo);
- a descrição dos papéis dos usuários do sistema, suas características e relações entre eles;
- as metas que cada papel de usuário quer alcançar através do sistema;
- as informações do contexto de uso, isto é, as características do ambiente onde a aplicação será utilizada.

Estas informações são provenientes do trabalho realizado previamente pelos engenheiros de requisitos e pelos profissionais de IHC. Elas são apenas transportadas para a ferramenta, para facilitar a comunicação das mesmas com os engenheiros de software. Pode-se entender que este componente da ferramenta é um filtro do trabalho já realizado nas fases anteriores do processo de desenvolvimento que expõe somente as informações descritas acima, que se pressupõe relevantes de serem comunicados aos engenheiros de software. Esta parte da ferramenta se comporta como uma espécie de base de conhecimento, mas que não necessariamente precisa estar implementada em software. Além disto, neste trabalho, propõem-se que estas informações sejam apresentadas em linguagem natural, sem a utilização de uma notação específica para a sua representação.

Não é o objetivo deste trabalho descrever como se realizam o levantamento, a análise e a modelagem destas informações. Isto pode ser encontrado amplamente na literatura como, por exemplo, em (Carroll, 1995, Hackos, 1998, Preece et al. 1994, Preece, 2002, Hix, 1993, Paternò 2000, Kotonya, 1998). O objetivo é deixar estas informações disponíveis para apoiar a comunicação entre os engenheiros de software e os profissionais de IHC. A intenção é que estas informações sejam utilizadas nos outros componentes da ferramenta que está sendo proposta.

No Apêndice A deste trabalho é apresentada a modelagem completa do domínio de “Círculo de livros” utilizando a ferramenta ComunIHC-ES (na Figura 10 é apresentada a descrição geral deste domínio). Este domínio e o seu projeto de IHC serão utilizados para exemplificar as partes da ferramenta ComunIHC-ES ao longo deste capítulo e no estudo de caso desta pesquisa (capítulo 4).

No Círculo de Livros, um professor possui um acervo grande de livros e seus alunos fazem empréstimos destes livros. Um aluno pode transferir um livro diretamente para outro aluno. O maior objetivo deste sistema é o controle pelo professor de seu acervo de livros.

Figura 10: Descrição geral do domínio “Círculo de livros”.

Já podemos perceber aqui uma primeira diferença entre o conteúdo da mensagem que é transmitida para os usuários e esta que está sendo passada para os engenheiros de software. Através da ComunIHC-ES comunicamos aos engenheiros de software detalhes do

domínio da aplicação, os conceitos necessários para o entendimento do mesmo, e as características do contexto de uso. Um maior detalhamento deste tipo de informação precisa ser apresentado para os engenheiros, mas não para os usuários, pois estes últimos sabem em qual domínio e contexto o sistema está inserido e já conhecem os termos utilizados no domínio em questão.

3.2 Projeto da Interação

Nesta pesquisa, para se comunicar o projeto de IHC para os engenheiros de software, decidiu-se utilizar a linguagem MoLIC (Barbosa e Paula, 2003, Paula, 2003 e Silva, 2005), apresentada no capítulo 2 desta tese. Como já foi dito, a MoLIC é uma ferramenta epistêmica fundamentada na teoria da engenharia semiótica, concebida para apoiar os designers de IHC no projeto das possíveis conversas que podem ocorrer entre o usuário e o preposto do designer, durante a utilização do sistema interativo.

Optou-se pelo uso da MoLIC, pois, como foi visto no capítulo de trabalhos relacionados, ela é a única linguagem para se modelar a interação que permite uma visualização global de todas as conversas que podem ser travadas entre o usuário e o preposto do designer durante a utilização do sistema interativo, ou seja, que apresenta de uma forma não fragmentada todo o conteúdo do discurso interativo a ser apresentado para o usuário final. Outra questão é que, em trabalhos preliminares, a MoLIC se mostrou útil no trabalho dos profissionais de IHC (Silva e Barbosa, 2004) e também como facilitadora da comunicação entre estes profissionais e os engenheiros de software, que é o objetivo deste trabalho (Barbosa et al. 2004, Paula et al. 2004, Paula et al, 2005, Paula et al. 2005a).

Como também já foi dito, esta linguagem é composta por: um diagrama de metas que especifica as metas que os usuários terão ao utilizar o sistema e a relação entre elas, um diagrama de interação que representa uma visão global das possíveis conversas que podem ocorrer entre o usuário e o preposto do designer para alcançarem as metas representadas no diagrama de metas, uma especificação textual que detalha as conversas representadas no diagrama de interação e uma representação de signos que descreve os signos que fazem parte da conversa usuário-preposto do designer.

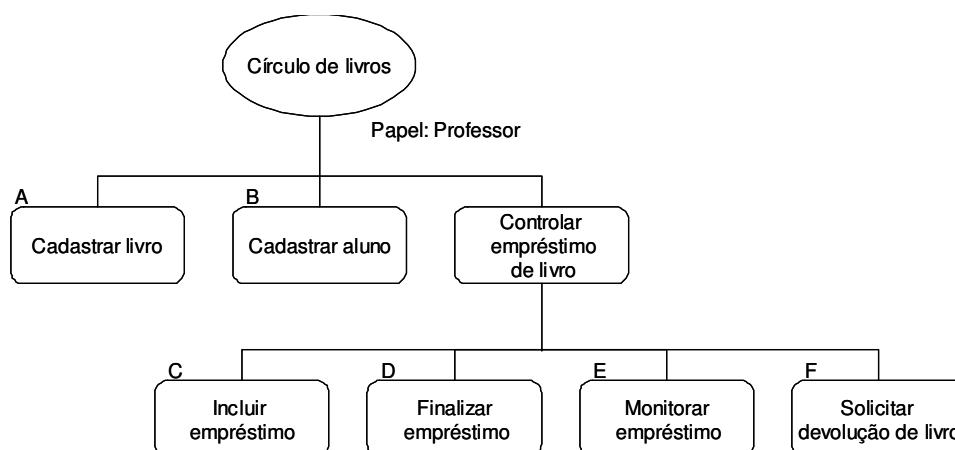
Silva (2005) apresenta no capítulo 3 de seu trabalho toda a evolução da 1ª edição da MoLIC para a 2ª edição, e no capítulo 4 toda a 2ª edição desta linguagem. Não é o objetivo deste trabalho revisar a linguagem MoLIC, ou mesmo apresentá-la em detalhes nesta tese (serão descritos apenas os elementos da linguagem que foram utilizados nos exemplos e no estudo de caso deste trabalho). A intenção é utilizá-la com o objetivo de apoiar a comunicação entre os profissionais de IHC e os engenheiros de software, e verificar se ela atende ou não a esta função. Caso não atenda, pretende-se levantar quais os motivos disto e o que deve ser modificado na linguagem futuramente.

Segue a descrição das representações contidas na linguagem MoLIC.

3.2.1 Diagrama de Metas

Através da linguagem MoLIC pode-se estruturar as metas (ou objetivos) que os usuários terão ao utilizar o sistema através de um diagrama de metas. Este diagrama provê uma visão macro das metas que cada classe de usuários pode realizar. Uma meta é representada por um retângulo com bordas arredondadas contendo o nome da meta, expresso do ponto de vista do usuário, uma letra (para facilitar a rastreabilidade com o diagrama de interação da MoLIC), e também o(s) papel(papéis) de usuários que poderá(ao) atingi-la através do sistema.

Na Figura 11 é apresentado o diagrama de metas do papel “professor” no domínio “Círculo de Livros”, cuja modelagem completa se encontra no Apêndice A.



Obs.: As metas de alteração e exclusão de livros e alunos não foram incluídas para simplificar a modelagem.

Figura 11: Diagrama de Metas do papel “professor” no “Círculo de livros”.

3.2.2 Representação dos Signos

Através da MoLIC é possível representar os signos de interface¹⁰ que serão manipulados ou apresentados ao usuário durante a conversa usuário-preposto do designer, a relação entre eles e as características inerentes de cada signo, isto é, que não são dependentes do contexto de interação. Características como, por exemplo, sua descrição e sua origem (se é um signo de domínio, transformado devido à inserção de tecnologia ou somente da aplicação). Na 1ª e 2ª edição da MoLIC foi proposta a representação destes signos em forma de tabela, mas aqui propomos que as relações entre os signos sejam representadas em forma de diagrama para facilitar a sua visualização. A descrição dos mesmos e suas características continuam sendo representadas em forma de tabela.

A Figura 12 apresenta os signos do “Círculo de Livros” e suas relações. A tabela associada a esta representação diagramática pode ser vista no Apêndice A.

¹⁰ Em trabalhos anteriores a este, os autores Paula e Silva chamaram esta parte da linguagem de uma “Ontologia de Signos”. No entanto, até o momento esta não é uma realidade, pois não existem regras e nem inferências sobre os elementos desta representação. Silva e Barbosa (2007) propuseram chamar de “esquema conceitual de signos”, mas neste trabalho estamos utilizando uma outra representação menos detalhada.

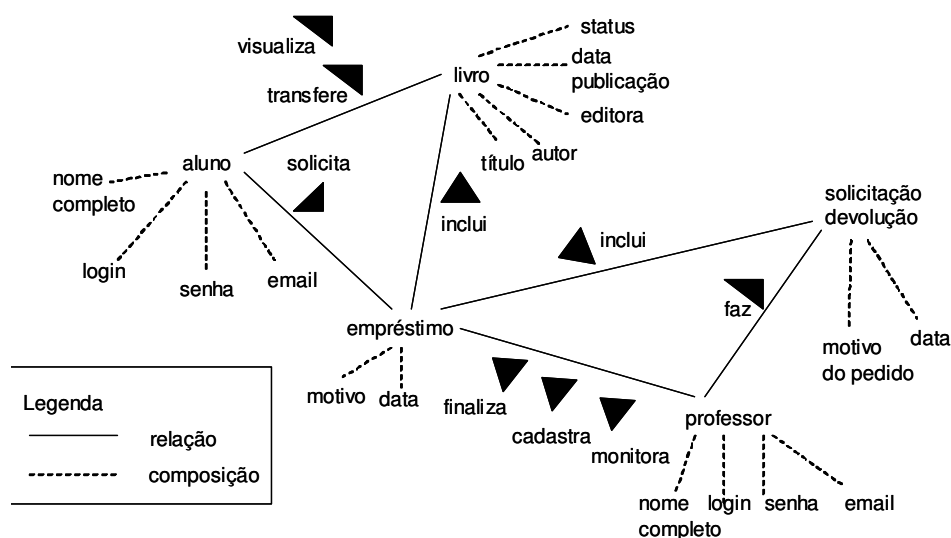


Figura 12: Representação gráfica dos signos do "Círculo de livros".

3.2.3 Diagrama de Interação

Através do diagrama de interação da MoLIC, é possível representar todas as conversas que o usuário pode travar com o preposto do designer durante a utilização da aplicação para atingir as suas metas (representadas no diagrama de metas da MoLIC). Neste diagrama, o designer de IHC define as conversas que podem ocorrer, os diálogos que compõem cada uma destas conversas, as falas do usuário que afetam o rumo das conversas e os momentos onde acontecem processamentos do sistema e as falas do preposto do designer que comunicam o resultado destes processamentos ao usuário. A seguir, cada um destes elementos serão explicados.

Para se representar a conversa e os seus diálogos, utiliza-se a noção de cena¹¹. Cena é o "local/momento" onde determinado tópico de conversa pode acontecer (Figura 13).

¹¹ A conversa é formada pelo o conjunto de diálogos necessários para se atingir uma meta.

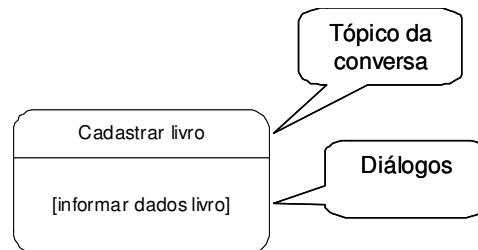


Figura 13: Cena relativa ao tópico de conversa “Cadastrar livro”.

A partir de uma cena, o usuário pode emitir falas que mudam o rumo da conversa. Estas falas na MoLIC são chamadas de falas de transição (Figura 14).

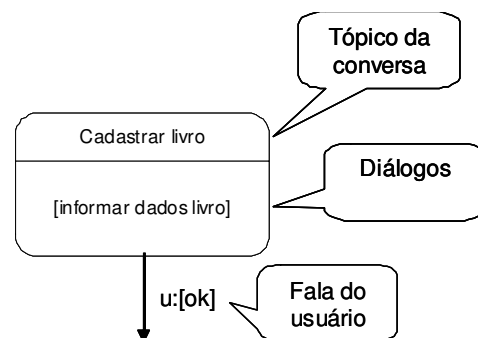


Figura 14: Representação de uma fala de transição emitida pelo usuário.

Outra representação do diagrama de interação são os processamentos do sistema. Estes processamentos geram as falas do preposto do designer como resposta a uma ou mais falas do usuário. Um processamento do sistema é representado através de uma caixa preta, para ressaltar que o usuário não sabe o que ocorre “dentro” do sistema (para o usuário, o sistema é a interface). O usuário só tomará conhecimento do que aconteceu através das falas do preposto, que são emitidas a partir do processamento (Figura 15).

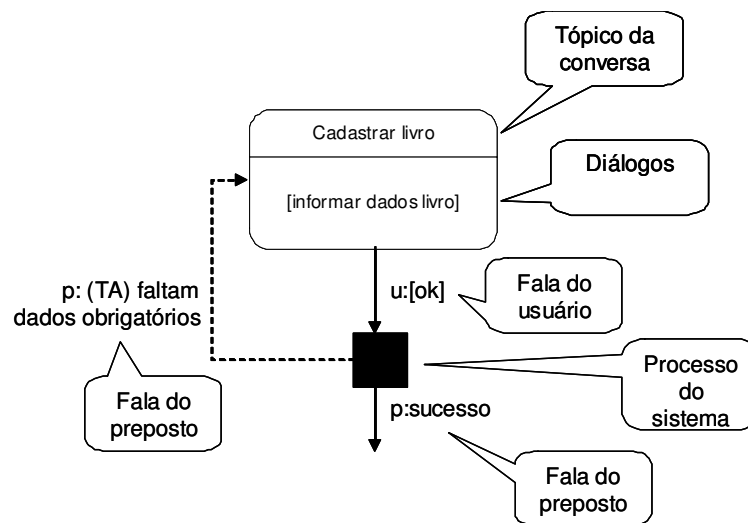


Figura 15: Representação do processamento do sistema e das falas do preposto do designer.

Como a metáfora utilizada é a da conversa, rupturas na conversa usuário-preposto do designer podem ocorrer. Estas rupturas também são representadas no diagrama através de falas do preposto que são emitidas através de um processamento do sistema. Estas falas são representadas através de transições tracejadas, como pode ser visto na Figura 15. As rupturas na conversa podem ser prevenidas pelo preposto, tratadas pelo usuário com a ajuda do preposto ou apenas informadas ao usuário. Então, para cada ruptura, uma das seguintes categorias é associada:

- prevenção passiva (PP): rupturas que devem ser prevenidas por documentação ou instruções *online* (por exemplo, o formato esperado para um campo data).
- prevenção ativa (PA): rupturas que devem ser prevenidas ativamente pelo sistema (por exemplo, impedir que o usuário digite letras ou símbolos em um campo numérico).
- prevenção apoiada (PAp): situações que o sistema consegue detectar como sendo erros em potencial, mas cuja decisão recai sobre o usuário (por exemplo, mensagens de confirmação).
- tratamento apoiado (TA): rupturas que devem ser tratadas pelo usuário com apoio do preposto do designer (por exemplo, apresentar uma mensagem de erro e uma oportunidade para o usuário corrigi-lo).

- captura de erro (CE): rupturas que são identificadas pelo sistema e devem ser notificadas ao usuário, sem que haja qualquer passo corretivo possível dentro do sistema (por exemplo, “O arquivo está corrompido.” ou “Espaço em disco insuficiente.”).

Um outro elemento do diagrama de interação é a representação do acesso ubíquo. Este elemento é utilizado para indicar quais os tópicos de conversa podem ser travados entre o usuário e o preposto em qualquer ponto da interação. Na Figura 16, por exemplo, o acesso ubíquo U2 indica que a qualquer momento durante a interação com a aplicação, o professor poderá pedir para cadastrar um livro.

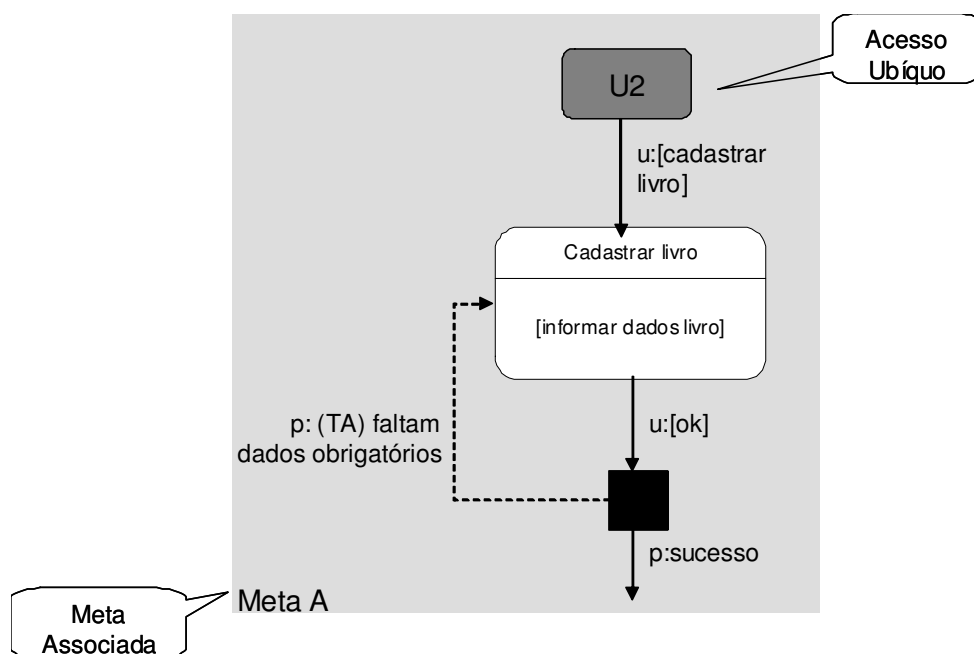


Figura 16: Representação de acesso ubíquo e da rastreabilidade com o diagrama de metas.

Na Figura 16 pode-se ver a rastreabilidade entre o diagrama de metas e o de interação (indicação de Meta A). Ou seja, no diagrama de interação o designer deve indicar quais são as conversas necessárias para se atingir as metas representadas no diagrama de metas.

A Figura 17 apresenta o mapa de todas as conversas que o professor pode travar com o preposto do designer na aplicação de “Círculo de livros”.

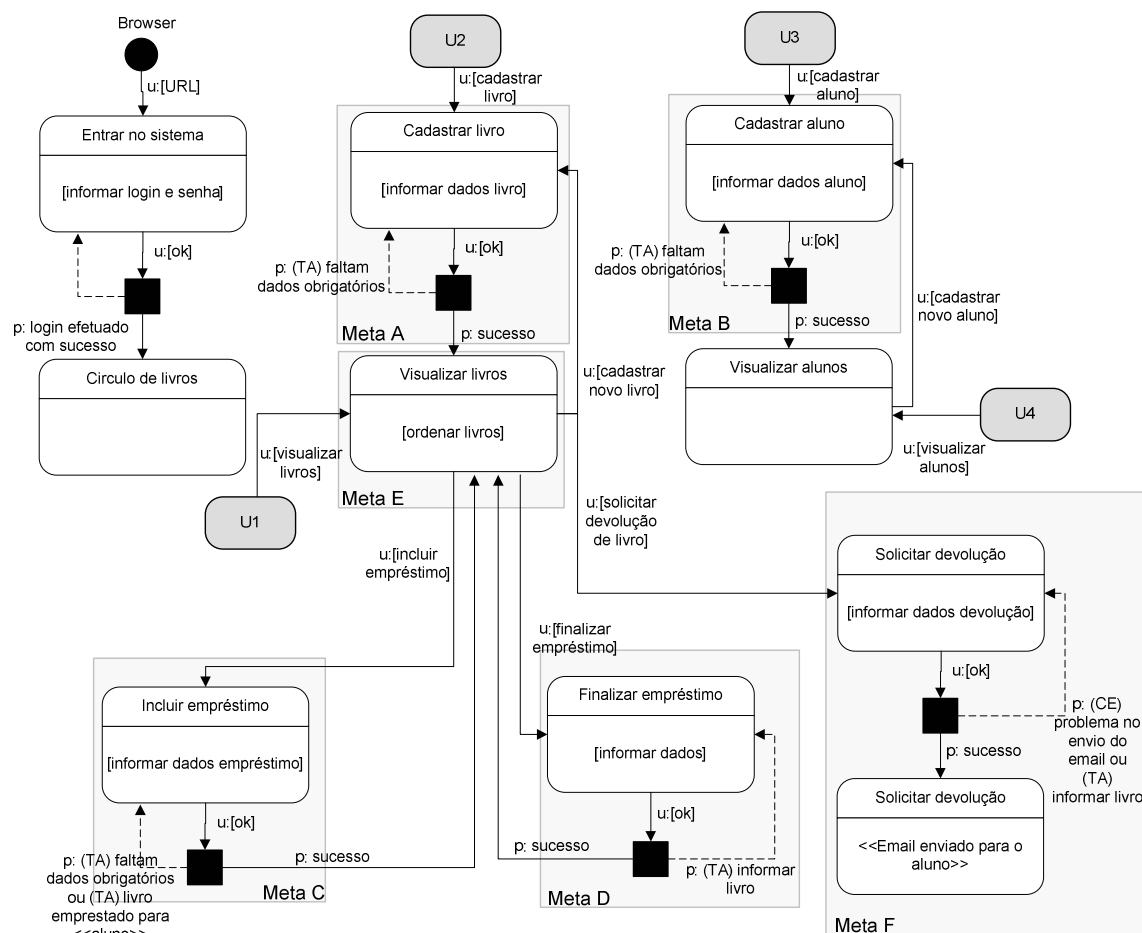


Figura 17: Diagrama de Interação do papel "professor".

Existem vários outros elementos na notação do diagrama de interação que podem ser vistos em (Silva, 2005). Os apresentados nesta seção são suficientes para entender a ferramenta ComunIHC-ES.

3.2.3.1 Especificação Textual do Diagrama de Interação

Associada ao diagrama de interação, existe uma especificação textual que representa os signos que compõem cada fala de cada diálogo¹². Nesta especificação textual, ao contrário da representação dos signos apresentada na Figura 12, são determinadas as

¹² No diagrama de interação só são representadas as falas de transição do usuário e do preposto do designer.

características dos signos dependentes do contexto de interação, isto é, da conversa onde estão inseridos. Para cada signo é informado se ele será apresentado ao usuário pelo preposto ou manipulado pelo usuário, isto é, quem é o enunciador da fala associada ao signo. Além disto, são descritos, por exemplo, o seu tipo de informação (ex.: numérico, textual etc), a sua obrigatoriedade naquele diálogo, seu valor *default*, seus possíveis valores (restrições sobre o conteúdo do signo), seu formato (restrições sobre a expressão do usuário), e o tipo de elemento de interface (ex.: texto livre, escolha simples - ordem alfabética, escolha múltipla etc) que deve ser utilizado. Os possíveis valores e formato do signo são importantes para se prevenir possíveis rupturas na comunicação. Vale observar que a definição do tipo de elemento de interface vai além da interação propriamente dita e faz uma ponte com o design da interface do sistema, sem no entanto determinar qual será o *widget* concreto utilizado, mas sim características que este deve possuir (restrições sobre a expressão do designer). Em geral, essa especificação textual é elaborada em fases posteriores do processo de design, quando o design da interação já é considerado relativamente estável. A Tabela 5 apresenta a especificação textual dos signos que compõem o diálogo [informar dados livro] da cena “Cadastrar livro” da Figura 13. Ela é apresentada aqui em forma de tabela, para organizar melhor a sua informação e facilitar o seu entendimento.

Cadastrar livro – [informar dados livro]							
Signo	Enunciador (Preposto ou usuário)	Tipo de informação	Obrigatoriedade de (S/N)	Valor <i>default</i>	Possíveis valores	Formato	Tipo de elemento de interface
título	usuário	textual	S	n/a	n/a	n/a	Texto livre
autor(s)	usuário	textual	S	n/a	n/a	n/a	Texto livre
editora	usuário	textual	S	n/a	n/a	n/a	Texto livre
data publicação	usuário	numérico	S	ano corrente	1900 até ano corrente	aaaa	Escolha simples – ordem crescente
status	usuário	textual	S	livre	Livre/empr estado para <<aluno>>	n/a	Escolha simples

Tabela 5: Especificação textual associada ao diálogo [informar dados do livro].

Então, a especificação textual contém os detalhes dos signos envolvidos nos diálogos representados no diagrama de interação. No Apêndice A deste trabalho está representada toda a especificação textual dos signos encontrados no diagrama de interação do papel “professor” apresentado na Figura 17.

É importante ressaltar neste trabalho, que através da MoLIC o designer de IHC consegue definir e representar os caminhos preferências e alternativos que o usuário pode seguir para atingir os seus objetivos. Entretanto, ainda não existe um elemento nesta linguagem que possibilite a explicitação de qual é o caminho preferencial, de acordo com o designer, para se alcançar uma meta.

Como foi visto na descrição do conteúdo da ferramenta (seção 1.2.1), a indicação dos “modos preferenciais” deve ser comunicada aos engenheiros de software. Então, fica como

trabalho futuro a definição de como se explicitar esse tipo de informação através da MoLIC.

3.3 Comunicação do Modelo de Interação

Como o objetivo deste trabalho é a comunicação e a negociação das decisões de IHC, é preciso facilitar o entendimento do projeto de IHC pelos engenheiros de software. Para isto, é fundamental explicitar a lógica de design que está por trás das decisões representadas no modelo de interação. Então, decidiu-se criar uma camada de comunicação sobre este modelo.

Os trabalhos de (Prates et al., 2000) e (Silveira, 2002) serviram de inspiração para a construção desta parte da ferramenta ComunIHC-ES. Em Prates et al. foi definido o método de avaliação de comunicabilidade, onde o avaliador utiliza expressões de comunicabilidade que representam dúvidas dos usuários ao interagirem com a aplicação, para assinalar rupturas na comunicação usuário-preposto do designer durante a interação com um sistema interativo. As expressões estão especificadas na primeira coluna da Tabela 6. Silveira propôs em seu trabalho uma ferramenta epistêmica para a construção de sistemas de ajuda *online*. A autora acrescentou às expressões de comunicabilidade outras expressões para cobrir o conjunto das dúvidas mais frequentes dos usuários ao interagirem com uma aplicação tal como apresentado na literatura sobre sistemas de ajuda (segunda coluna da Tabela 6). Estas expressões são utilizadas pelos designers na construção do sistema de ajuda com o objetivo de os usuários, através da enunciação delas em uma aplicação, possam obter informações minimalistas sobre partes do sistema, informações que incluem a lógica de design. Neste trabalho pretende-se também utilizar expressões para explicitar a lógica do designer de IHC sobre partes da modelagem de interação, mas o público-alvo são os engenheiros de software. Isto é, eles poderão utilizar as expressões para aprofundarem o seu conhecimento sobre o projeto da interação.

Expressões de Comunicabilidade	Dúvidas freqüentes dos usuários
Cadê?	Como eu faço isto?
E agora?	Por que eu devo fazer isto?
O que é isto?	Para que serve isto?
Epa!	Quem/O que é afetado por isto?
Onde estou?	Isto depende de quem/o que?
Assim não dá.	Quem pode fazer isto?
Por que não funciona?	Existe outra maneira de fazer isto?
Ué, o que houve?	Onde eu estava?
Para mim está bom...	
Desisto.	
Vai de outro jeito	
Não, obrigado.	
Socorro!	

Tabela 6: Expressões de comunicabilidade e dúvidas freqüentes dos usuários

Baseando-se nas expressões da Tabela 6 e nas dúvidas que os engenheiros de software podem ter sobre cada elemento do diagrama de interação, foram definidas perguntas que os profissionais de IHC podem responder sobre a modelagem da interação, ou seja, sobre o que está representado via MoLIC. A Tabela 7 apresenta a origem (a fonte de inspiração) das perguntas e as perguntas reescritas de forma que os engenheiros de software possam enunciá-las. Algumas expressões da Tabela 6 não foram utilizadas, pois elas se referem a questões que podem ser enunciadas pelos usuários durante rupturas em uma comunicação com uma interface (por exemplo, Onde eu estava?, Ué, o que houve?, Por que não funciona? e Desisto!) ou utilizadas em tempo de design (por exemplo, o designer de IHC se questionar “Como eu faço para evitar que o usuário se pergunte ‘Onde eu estava?’ neste pedaço da interação?”), mas não fazem sentido na comunicação designer de IHC-engenheiro de software.

Origem	Pergunta
[Comunicabilidade] O que é isto?	O que é isto?
[Dúvidas usuários] Para que serve isto?	Para que serve isto?
[Dúvidas usuários] Por que eu devo fazer isto?	Por que isto pode/deve ser realizado?
[Dúvidas usuários] Como eu faço isto?	Como o usuário pode/deve realizar isto?
[Comunicabilidade] Epa!	É possível desfazer isto? Como?
[Dúvidas usuários] Quem pode fazer isto?	Quem pode fazer isto?
[Dúvidas usuários] Isto depende de quem/o quê?	Isto depende de quem/o quê?
[Dúvidas usuários] Existe outra maneira de fazer isto?	Existe outra maneira de fazer isto?
[Dúvidas usuários] Quem/O que é afetado por isto?	Quem/O que é afetado por isto?

Tabela 7: Perguntas a serem respondidas pelo profissional de IHC durante a modelagem da interação.

Estas perguntas podem ser aplicadas sobre os elementos da MoLIC, com o intuito de explicitar a metacomunicação do designer de IHC representada ali. No entanto, fica a cargo do designer de IHC responder em linguagem natural as perguntas sobre determinados elementos que ele achar importantes, ou seja, explicitar parte da comunicação representada via MoLIC que ele julgar necessária para o maior entendimento do projeto de IHC pelos engenheiros de software.

A seguir são apresentadas as respostas típicas às perguntas da Tabela 7:

- O que é isto? - explicação do conteúdo de cada elemento da modelagem;
- Para que serve isto? - relação entre o elemento e as metas que os usuários pretendem alcançar através da aplicação;

- Por que isto pode/deve ser realizado? - motivo pelo qual algo deve/pode ser realizado pelo usuário;
- Como o usuário pode/deve realizar isto? - descrição dos passos para se realizar determinado objetivo;
- É possível desfazer isto? Como? - descrição dos passos para se desfazer determinada ação ou conjunto de ações;
- Quem pode fazer isto? - especificação de qual(is) papel(is) de usuário(s) pode(m) realizar algo;
- Isto depende de quem/o quê? - determinação das pré-condições para se realizar algo;
- Existe outra maneira de fazer isto? - descrição de formas alternativas para se realizar algo;
- Quem/O que é afetado por isto? - definição das conseqüências provocadas pela realização de algo.

Como pode ser visto acima, aparentemente as respostas são focadas apenas na experiência do usuário, ou seja, têm o mesmo conteúdo de, por exemplo, um sistema de ajuda (exemplos podem ser visto no Apêndice A). Entretanto, as respostas devem ser pensadas em relação ao trabalho dos engenheiros de software, e ao conhecimento que estes possuem. Como já foi dito, os usuários possuem um maior conhecimento sobre o domínio em questão e o contexto de uso da aplicação do que os engenheiros de software. Em contrapartida, estes últimos detêm o conhecimento e estão envolvidos com questões tecnológicas do sistema que está sendo projeto. Tudo isto deve ser levado em consideração ao fornecer as explicações para os engenheiros de software.

Por exemplo, quando uma decisão representada em um elemento da modelagem possuir um ou mais aspectos tecnológicos subjacentes que impactem decisões específicas de projeto ou implementação por parte do engenheiro de software, as explicações associadas a esse elemento devem incluir a descrição de tais aspectos. Desta forma, podemos observar que serão explicitadas informações sobre decisões do projeto de IHC que

não interessam diretamente para os usuários, mas que são importantes para os engenheiros de software.

Outro tipo de informação que é interessante comunicar para os engenheiros de software são as relacionadas a decisões que foram tomadas pelo designer de IHC com o intuito de aumentar a flexibilidade e a eficiência de uso da aplicação (princípios de usabilidade prioritários a um projeto, por exemplo). Elas geralmente não são comunicadas via help, por exemplo, para um usuário final, mas devem ser explicitadas para os engenheiros de software.

Para clarear esta idéia, a Figura 18 apresenta um exemplo de um pedaço de um modelo de interação onde o usuário pode cadastrar um livro, e durante este cadastro ele tem a possibilidade de cadastrar um autor. Aplicando a pergunta “Para que serve isto?” sobre a transição entre a cena “Cadastrar livro” e “Cadastrar autor” (transição u:[cadastrar autor]), a resposta típica a esta pergunta, pensando que o público são os usuários da aplicação, seria algo equivalente a: “Para o usuário cadastrar o(s) autor(es) do livro que ele está inserindo”. Esta resposta pode ser suficiente para explicar ao usuário a existência da conexão entre as cenas. Entretanto, pensando no trabalho dos engenheiros de software, é interessante deixar claro o motivo de o designer de IHC ter criado este “atalho” para se cadastrar um autor, ao invés de deixar esta cena de cadastro a parte, em um outro contexto de interação. Então, é importante que esta resposta seja acrescida de algo equivalente a: “Desta forma, o usuário não precisa sair da cena, mudar de contexto, cadastrar o(s) autor(es) e ter que retornar novamente a cena de cadastro de livro e recomeçar o processo. Esta solução aumenta a eficiência da interação”.

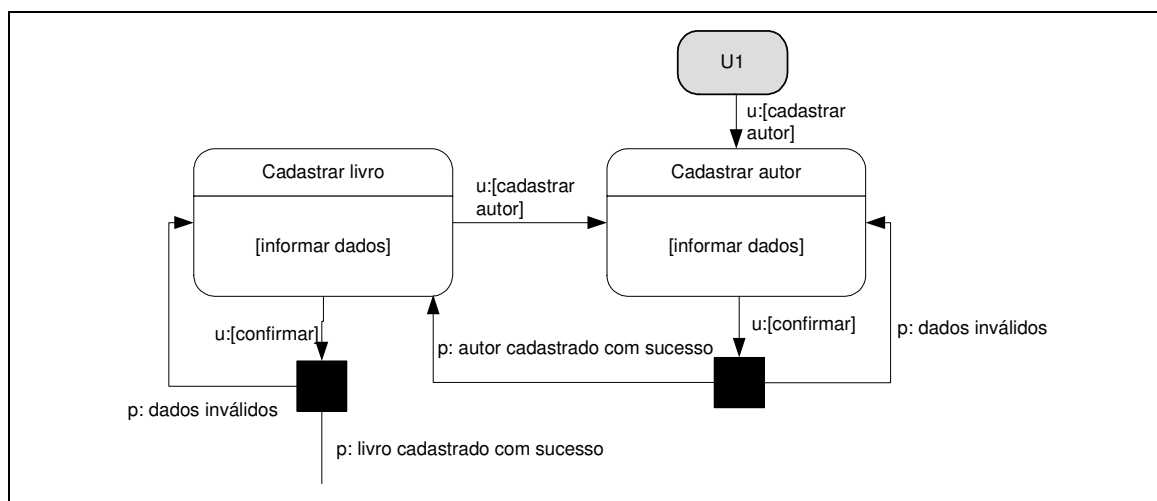


Figura 18: Parte de um modelo de interação – meta “Cadastrar livro”.

Como foi dito, é interessante também comunicar aos engenheiros de software decisões de design que foram tomadas devido a alguma característica particular do contexto de uso ou do perfil do usuário da aplicação. Retomando ao exemplo da seção 2.1.5.2.1, o sistema de biblioteca. Uma informação sobre o contexto de uso deste sistema, como foi citado, seria, por exemplo, a indicação de que só existirá um terminal na biblioteca onde todos os usuários poderão consultar o acervo da mesma. Com esta informação em mãos, o designer de IHC pode tomar a decisão de que o usuário, ao procurar por um livro e este não constar no acervo da biblioteca, ele seja levado novamente para a tela onde ele pode procurar por um outro livro Figura 19(a), ao invés de ir para uma segunda tela que apenas informa ao usuário o ocorrido Figura 19(b). Desta forma, o designer estará levando em consideração o contexto de uso, diminuindo o tempo de interação de cada usuário com o sistema interativo.

Depois de tomar a decisão acima e fazer a modelagem deste pedaço da interação, o designer de IHC pode utilizar a pergunta “Por que isto deve ser realizado?” sob a transição “p: livro não encontrado”, que volta para a tela “Buscar livro”, e explicar para o engenheiro de software esta característica do contexto de uso e, portanto, a existência desta única tela neste contexto de interação.

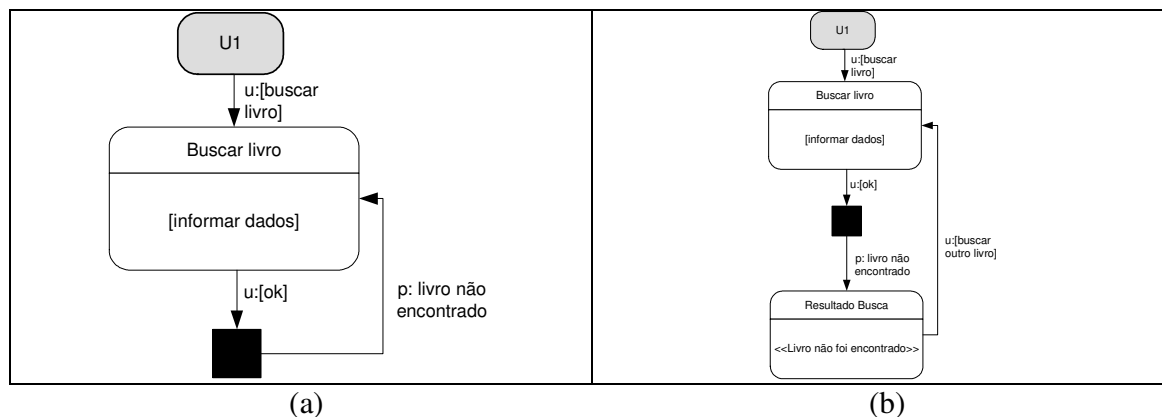


Figura 19: Parte do modelo de interação – meta “Buscar livro”.

A aplicação das perguntas sobre os elementos da MoLIC, além de apoiar a comunicação entre os profissionais, ajuda na reflexão dos designers de IHC, pois eles, no momento da elaboração das respostas, estarão “revisitando” as suas tomadas de decisão sobre a modelagem de interação. Além disto, as respostas a estas perguntas também são um insumo importantíssimo para a criação dos sistemas de ajuda online, segundo a proposta de Silveira (Silveira, 2002) descrita acima. Elas podem alimentar as respostas às expressões utilizadas na sua proposta de sistema de ajuda, ou vice-versa.

Barbosa e co-autores (Barbosa et al. 2004) também se basearam nas expressões de comunicabilidade e dúvidas mais frequentes dos usuários para enriquecerem a linguagem LEL (*Language Extended Lexicon*) (Leite et al. 1992 e Breitman e Leite, 2003), que se baseia na idéia de que primeiro se deve entender a linguagem do problema, sem se preocupar com o entendimento do problema. O objetivo do LEL é permitir a representação dos signos da linguagem do domínio da aplicação, através de dois tipos de descrições: (i) a denotação do termo ou frase e (ii) a conotação - informações extras sobre o contexto do signo. Além disto, cada signo é classificado em: objeto, sujeito, verbo ou estado. As descrições são feitas em linguagem natural e utilizando uma estrutura de hipertexto que facilita a relação e a navegação entre os termos. O LEL se baseia em dois princípios: “circularidade” e “vocabulário mínimo”. O primeiro atenta para o fato de se maximizar o uso de signos já existentes na linguagem durante a descrição de outros signos, e o segundo ressalta que a necessidade por vocabulário externo (fora do universo de discurso em questão) deve ser minimizada e reduzida ao menor conjunto possível. Kaplan e co-autores

descreveram um processo para a construção de uma representação utilizando o LEL (Kaplan et al., 2000).

O objetivo de Barbosa e co-autores em seu trabalho era aumentar a capacidade do LEL em ser um artefato de apoio na comunicação entre os membros de uma equipe de design. Para isto, os autores associaram a cada tipo de entrada no LEL (objeto, sujeito, verbo e estado), perguntas que devem ser respondidas pelo designer. Como já foi dito, estas perguntas se basearam nas expressões da Tabela 6. As propostas destes autores e a desta tese (apresentada nesta seção) são diferentes, pois atuam em representações distintas e em momento distintos do processo de design, mas possuem o mesmo objetivo – apoiar a comunicação entre os membros de uma equipe de design.

3.4 Correspondência entre a ComunIHC-ES e a UML

Como já foi dito no capítulo 2, na área de engenharia de software, o paradigma mais utilizado atualmente para design, desenvolvimento e implementação de software é o de orientação a objetos (OO) (Booch et al., 2007). Uma linguagem amplamente utilizada pelos engenheiros de software para se especificar, construir e documentar uma aplicação utilizando este paradigma é a UML (*Unified Modeling Language*) (Booch et al., 2005). Esta linguagem possui um conjunto de modelos que permite a representação do sistema em diversas perspectivas. *Use cases*, diagrama de classes e seqüências são talvez os modelos mais utilizados desta linguagem.

Os diagramas *use cases*, como foi discutido na seção 2.2.1, representam conjuntos de ações do sistema necessárias para se produzir resultados relevantes para os atores do sistema. O diagrama de classes representa a estrutura estática do sistema, isto é, as suas classes, atributos, relações e métodos. Já os diagramas de seqüências representam as possíveis interações entre objetos, isto é, as mensagens que podem ser trocadas entre eles e a seqüência destas.

Para apoiar o trabalho dos engenheiros de software após o entendimento do projeto de IHC via ferramenta ComunIHC-ES, foram definidas possíveis correspondências entre as

informações representadas nesta ferramenta e os 3 diagramas UML enumerados acima. A intenção é facilitar o “consumo” das informações representadas na ferramenta pelos engenheiros, caso estes estejam utilizando a UML. É importante lembrar que um dos objetivos da ComunIHC-ES é apoiar e facilitar o trabalho dos engenheiros de software, e não integrar aspectos de IHC com a modelagem OO, como foi apresentado na seção 2.3.3.1 .

A Figura 20 apresenta a ferramenta ComunIHC-ES com esta 4ª parte – as correspondências com a UML.

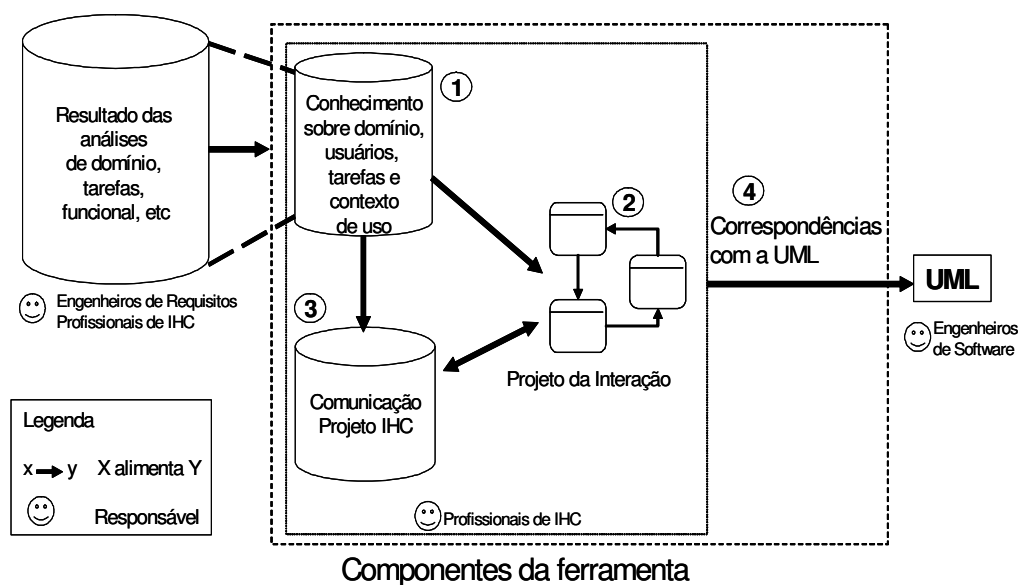


Figura 20: Componentes da Ferramenta ComunIHC-ES.

Seguem as possíveis correspondências entre a ferramenta ComunIHC-ES e os diagramas UML. Elas podem ser utilizadas pelos engenheiros de software para apoiar o seu trabalho de especificação das funcionalidades internas do software.

ComunIHC-ES → *use cases*: O conhecimento apresentado na parte 1 da ferramenta e o diagrama de metas da MoLIC (parte 2) contêm as metas que os usuários querem alcançar através do sistema, podendo apoiar a definição das ações do sistema nos *use cases*. Além disto, a parte 1 da ferramenta apresenta os papéis dos usuários, que nos *use cases* aparecerão como atores. As falas do usuário e do preposto, e os signos nelas envolvidos, representados no diagrama MoLIC, podem servir de insumo para a definição dos objetos

que são transferidos durante as ações representas nos *use cases*. Uma outra correspondência importante é a definição das possíveis rupturas na comunicação e os fluxos alternativos de eventos representados nos *use cases*.

ComunIHC-ES → diagrama de classes: A representação dos signos na MoLIC pode contribuir para a definição de classes, seus atributos e relações no diagrama de classes (estes elementos serão somente aqueles diretamente acessados pelos usuários, e não atributos auxiliares definidos para apoiar o processamento). No diagrama MoLIC, cada diálogo disponível para o usuário em uma cena pode corresponder a um método no diagrama de classes.

ComunIHC-ES → diagrama de seqüências: Quando uma das instâncias no diagrama de seqüências representa um papel de usuário, as mensagens que saem e chegam nesta instância podem ser obtidas ou verificadas através das falas do usuário e do preposto representadas no diagrama MoLIC. Além disto, uma informação muito importante contida na MoLIC é a ordem temporal dessas mensagens.

Estas possíveis correspondências podem ser vistas de forma simplificada na Figura 21. Além disto, um estudo inicial onde a MoLIC (representando a meta “Reservar quarto(s) em um hotel”) gerou insumos para a construção de diagramas UML pode ser visto em (Paula et al., 2005).

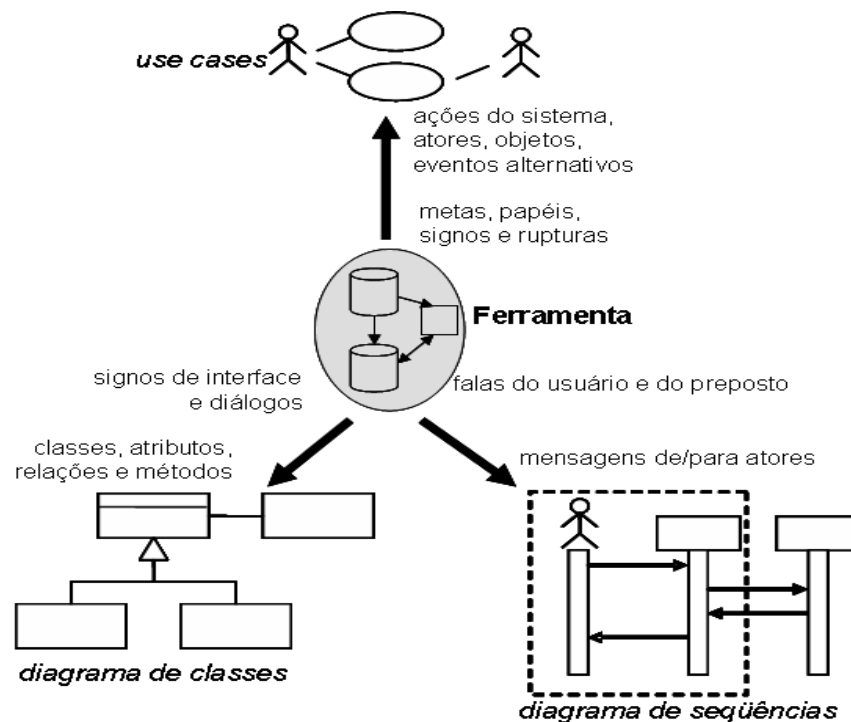


Figura 21: Possíveis correspondências entre a ferramenta ComunIHC-ES e os diagramas UML.

Um trabalho interessante seria a verificação de novas possíveis correspondências entre a ferramenta ComunIHC-ES e outros diagramas da UML, mas que está fora do escopo deste trabalho.

3.5 Processo de Leitura da Ferramenta

Como o objetivo da ferramenta ComunIHC-ES é apoiar a comunicação entre os profissionais de IHC e os engenheiros de software, para facilitar o uso da ferramenta como apoio a esta comunicação, sugerimos um caminho de leitura das partes da ferramenta pelos engenheiros de software. Vale observar que esta leitura não exclui a comunicação direta entre os profissionais.

O caminho é o seguinte:

1. O engenheiro de software lê a parte 1 da ferramenta (Figura 20), isto é, o conhecimento sobre domínio, usuários, tarefas e contexto de uso que estão por trás da solução que está sendo proposta pelos profissionais desta área;

2. Após ter entendido o conteúdo da parte 1, o engenheiro de software parte para a leitura da modelagem de interação – parte 2 da ferramenta. Ele deve conhecer a semântica e a notação da MoLIC. A modelagem deve ser lida da seguinte maneira:
 - a. O engenheiro deve ler o diagrama de metas para entender, de uma forma mais estruturada que na parte 1 da ferramenta, as relações entre as metas e os papéis de usuários que podem atingi-las através do sistema;
 - b. Entendidas as metas dos usuários, o engenheiro deve conhecer os signos que serão utilizados na aplicação, através da “representação dos signos” na MoLIC;
 - c. Com isto, o engenheiro pode varrer o diagrama de interação (primeiramente sem consultar a especificação textual) para ter o conhecimento do mapa global da interação, e à medida que for percorrendo os elementos deste diagrama, acionar as perguntas disponíveis e associadas a cada um deles – parte 3 da ferramenta;
 - d. Depois de se ter esta visão global das conversas que o usuário pode travar com o preposto do designer, ele pode partir para uma leitura mais detalhada do diagrama, isto é, recorrer à especificação textual e às perguntas associadas a seus elementos.
3. Com o conhecimento da proposta de IHC, os profissionais podem negociar a solução de IHC, fazer as modificações necessárias, e a partir daí os engenheiros de software podem utilizar as correspondências ComunIHC-ES → UML descritas na ferramenta (parte 4), caso a aplicação esteja sendo desenvolvida utilizando o paradigma OO e a linguagem UML.

O capítulo seguinte apresenta um estudo de caso realizado com engenheiros de software com o objetivo de se verificar o uso da ferramenta ComunIHC-ES no apoio à comunicação durante o processo de desenvolvimento de software.